UNED
Escuela Internacional de Doctorado
EIDUNED

TESIS DOCTORAL

2021

AUTOMATIC CATEGORIZATION OF ELECTRONIC HEALTH RECORDS

MARIO ALMAGRO CÁDIZ

DOCTORAL PROGRAMME IN INTELLIGENT SYSTEMS

Dr. RAQUEL MARTÍNEZ UNANUE
Dr. VÍCTOR FRESNO FERNÁNDEZ

UNED
Escuela Internacional de Doctorado
EIDUNED

TESIS DOCTORAL

2021

AUTOMATIC CATEGORIZATION OF ELECTRONIC HEALTH RECORDS

MARIO ALMAGRO CÁDIZ

DOCTORAL PROGRAMME IN INTELLIGENT SYSTEMS

Dr. RAQUEL MARTÍNEZ UNANUE
Dr. VÍCTOR FRESNO FERNÁNDEZ

AGRADECIMIENTOS

Este documento es la culminación de una investigación de varios años que ha supuesto una etapa de mi vida muy significativa y que recordaré con mucho cariño. Por ello, me gustaría dedicar este apartado para agradecer a todas aquellas personas que me han ayudado a avanzar durante todo este tiempo.

Como es de esperar, me gustaría dar las gracias a mis directores, Raquel Martínez y Víctor Fresno, quienes además de enseñarme constantemente los fundamentos de la actividad investigadora, me han acogido y echado una mano siempre que lo he necesitado. Estoy enormemente agradecido por sus consejos y sugerencias, y por dejarme formar parte de un equipo entrañable. A ellos les debo mi formación investigadora, pero también muy buenos momentos, recuerdos y, sobre todo, ánimos en los momentos más agotadores. En esa línea, no me olvido de Soto Montalvo, con quien he tenido la oportunidad de trabajar periódicamente durante toda esta etapa y de quien también he aprendido mucho.

También me gustaría reconocer el apoyo del resto del departamento. Gracias a ellos he tenido la oportunidad de llevar a cabo esta investigación con todas las facilidades posibles. Por esa razón admiro la labor de personas como Lourdes Araujo y Julio Gonzalo, con la que han ayudado a progresar a todo el departamento. Por otro lado, debo agradecer el esfuerzo del personal del Hospital Universitario Fundación Alcorcón (HUFA), que tan desinteresadamente nos ha cedido los datos necesarios para la experimentación.

Además me gustaría alabar la amabilidad de la gente de la Universidad del País Vasco (UPV) y University College London (UCL). En especial, agradezco la cálida y generosa acogida de Hegler Tissot, a quien aprecio mucho. Le doy las gracias por su interés y su buena disposición. Espero que volvamos a encontrarnos. Respecto mis compañeros, quisiera nombrar a todos aquellos con los que he coincidido estos años y que tanto aprecio, como Gildo, Agustín, y Alicia. Todos ellos han influido en este trabajo.

Y finalmente dedico este último párrafo a las personas que más me importan por su respaldo, mi vieja y nueva familia. Gracias a mi abuela que ya no está, a mi hermana, y en especial a mis padres, Juan Carlos y Ana. Ellos han sido siempre mi punto de apoyo en los buenos y malos momentos. Les agradezco mucho sus ánimos constantes y la ayuda que me han prestado en todo momento, como la elección de algunas palabras. Por último, quiero darle las gracias a una persona muy especial para mí que siempre ha estado implicada durante este período. Muchas gracias a Isa por todos sus ánimos durante los congresos, todas las tardes de trabajo en las que ha estado presente y todas sus opiniones sobre la narrativa y los formatos. Sin duda, un trocito de tesis es también suyo.

RESUMEN

La Clasificación Internacional de Enfermedades (*International Statistical Classification of Diseases and Related Health Problems*, *ICD*) es el estándar mundial más extendido para la recogida de información sanitaria. Este estándar organiza jerárquicamente miles de diagnósticos[1] detallando diferentes niveles de información agregada y vinculando cada uno a un identificador único, o código. Esta clasificación sirve de apoyo a estudios estadísticos, ensayos clínicos, auditorías médicas y financiación de los hospitales. Aunque el flujo de información clínica maneja conceptos estandarizados para asegurar la interoperabilidad de los datos, la mayor parte de la información relevante en los Informes Médicos Electrónicos (*Electronic Health Records*, *EHRs*) no está estructurada, ya que los médicos requieren la flexibilidad del texto en lenguaje natural para describir casos clínicos, cubriendo cada escenario de forma rápida y sencilla. Por lo tanto, la transformación del texto en datos estructurados, y en particular la tarea de asociar los hallazgos y síntomas con las etiquetas diagnósticas apropiadas, denominada codificación *ICD*, constituye un proceso importante para la categorización de los *EHRs*. Por este motivo, existen profesionales especializados en la traducción de textos a códigos *ICD*, también conocidos como codificadores.

Esta tesis doctoral explora múltiples métodos para abordar la codificación de textos desde una perspectiva informática con el objetivo de apoyar la tarea de codificación. Aunque existe una variedad de herramientas asistidas por ordenador para ayudar a los codificadores, la tarea de codificación automática se enfrenta a retos no resueltos y sigue siendo predominantemente manual. Además, hasta ahora la mayoría de las investigaciones académicas se han realizado en entornos de complejidad reducida para abordar un aspecto concreto de la codificación. Por este motivo, el objetivo

[1] Algunas modificaciones del *ICD* también incluyen procedimientos

principal de esta tesis es identificar las mejores formas de asignar diagnósticos y procedimientos adecuados a los *EHRs* sin reducir la complejidad, adaptados a todas las singularidades de la codificación. En concreto, los fundamentos de la tesis se estructuran en torno a los retos computacionales que hemos detectado en la décima revisión del *ICD* (*ICD-10*) por ser la versión adoptada en la mayoría de los países en el momento de la publicación de esta tesis. Dada la disponibilidad de datos, hemos procesado *EHRs* en lenguas europeas, con especial atención al español. El gran número de códigos potenciales (del orden de 100.000), la diferencia en la prevalencia de las enfermedades, el acceso limitado a los datos clínicos, así como la estructura jerárquica con diferentes granularidades semánticas en las categorías finales son propiedades inherentes a la codificación *ICD-10* que hemos asociado a la escasez de instancias para muchas etiquetas, a las distribuciones desequilibradas de datos, y a los problemas de generalización y concordancia semántica.

En primer lugar, nos hemos centrado en la escasez de datos globales y específicos de las etiquetas. Los registros de pacientes contienen grandes volúmenes de información sensible, por lo que el acceso a los conjuntos de datos clínicos suele estar sujeto a fuertes restricciones impuestas por las políticas de privacidad. Además de los conjuntos de datos reducidos, la exhaustividad y la especificidad de la norma implican menores probabilidades de asignación para muchos códigos, lo que significa menos ejemplos por etiqueta. Dadas las dificultades para inferir gran parte de la nomenclatura a partir de los datos, hemos explorado técnicas no supervisadas basadas en la concordancia léxica y semántica entre las representaciones de las etiquetas y los *EHRs*. Así, hemos comparado diferentes representaciones de etiquetas en un Modelo de Espacio Vectorial (*Vector Space Model*, *VSM*), utilizando un preprocesamiento exhaustivo para tratar los sinónimos. Como alternativa, hemos propuesto un método de similitud que explota la estructura de SNOMED CT para identificar conjuntos de pruebas con diferentes niveles de abstracción.

En segundo lugar, hemos abordado el enorme desequilibrio de las colecciones de datos académicas y hospitalarias. Computacionalmente, la codificación *ICD-10* implica una clasificación multietiqueta con más de 10.000 categorías, pero el número puede aumentar hasta 140.000 códigos *ICD* con modificaciones como CIE-10-ES y *ICD-10-CM*. Además del enorme espacio de etiquetas, las condiciones de salud tienen una prevalencia diferente entre las poblaciones, con pocos diagnósticos frecuentes y muchos raros, lo que lleva a distribuciones extremas de datos. Por lo tanto, hemos explorado métodos de aumento de datos para mejorar la representabilidad de los códigos minoritarios aplicando técnicas de sustitución léxica y Traducción Automática (*Machine Translation*, *MT*). También hemos propuesto el uso de métodos de Clasificación Extrema Textual Multietiqueta (*eXtreme Multi-label Text Classification*, *XMTC*) que explotan las codependencias de las etiquetas para mejorar la inferencia de los códigos menos frecuentes, al tiempo que se abordan los problemas de escalabilidad.

En tercer lugar, hemos abordado las diferencias de significado entre categorías y documentos. La codificación es una tarea de alto nivel que requiere un amplio conocimiento semántico del dominio biomédico para hacer frente a los distintos grados de abstracción. La norma está diseñada con fines estadísticos para agrupar conceptos clínicos, lo que plantea problemas de generalización durante el aprendizaje a partir de ejemplos. Por ello, hemos explorado múltiples métodos de aprendizaje por transferencia para introducir conocimiento externo en la tarea. Hemos llevado a cabo experimentos para examinar el efecto de las técnicas *MT* en la aplicación de enfoques lingüísticos cruzados. También hemos explorado la generación y aplicación de representaciones vectoriales dentro del dominio para los registros en español. Adicionalmente, hemos propuesto pre-entrenar los modelos en categorías jerárquicas superiores para incorporar características generales en el aprendizaje de las categorías finales.

Por último, hemos observado que las técnicas se han mostrado a menudo interdependientes y complementarias. Por este motivo, hemos abordado los tres retos señalados de forma simultánea con una combinación de: métodos basados en similitud semántica para predecir códigos no incluidos en los ejemplos; métodos de aumento de datos para generar nuevos ejemplos para los códigos minoritarios y reducir el desbalanceo; conjuntos de algoritmos XMTC para promover las clases minoritarias y reducir el coste computacional; y word embeddings pre-entrenados en Electronic Health Records (EHRs) y tesis doctorales de medicina para introducir conocimiento más general y mejorar la generalización durante el aprendizaje.

Como resultado, hemos implementado una aproximación compuesta por diferentes técnicas que consigue mejoras significativas respecto a otras aproximaciones convencionales exploradas. En particular, hemos observado que los algoritmos XMTC junto con el aumento de datos son los más adecuados para la tarea de codificación ICD-10. De hecho, la combinación de algoritmos XMTC produce el mayor incremento en las métricas de evaluación para los códigos frecuentes, mientras que las técnicas de sustitución léxica son la base para conseguir la mayor mejora en la predicción de códigos poco representados. Por otro lado, los métodos no supervisados son necesarios para predecir los códigos sin representación. Las hipótesis y conclusiones expuestas para dichos métodos están relacionadas con la naturaleza inherente de la clasificación y son fácilmente extrapolables al resto de las versiones ICD.

ABSTRACT

The International Statistical Classification of Diseases and Related Health Problems (ICD) is the most widespread global standard for the collection of health information. It hierarchically organises thousands of diagnoses[2] by detailing different levels of aggregated information and linking each one to a unique identifier, or code. Such a classification supports statistical studies, clinical trials, medical audits, and hospital funding. Although the clinical information flow operates standardised concepts to ensure interoperability of data, most of the relevant information in Electronic Health Records (EHRs) is not structured as physicians require the flexibility of natural language text to describe clinical cases, covering every scenario quickly and easily. Hence, the transformation of text into structured data, and in particular the task of associating findings and symptoms with appropriate diagnostic labels, called ICD coding, constitutes a major process for categorising EHRs. For this reason, there are professionals specialized in translating texts into ICD codes, also known as coders.

This PhD thesis explores multiple unsupervised and supervised methods to tackle ICD coding from a computer science perspective with the goal of supporting coders. Although there is a variety of computer-assisted tools to aid coders, the task faces unresolved challenges and remains predominantly manual. In addition, most academic research have been conducted in environments of reduced complexity to address a particular aspect of the coding. For this reason, the main objective of this thesis is to identify the best ways to assign appropriate diagnoses and procedures to EHRs without reducing complexities, tailored to all the coding singularities. Specifically, the PhD foundations are structured around the computational challenges we have detected in the tenth revision of the ICD (ICD-10) as it is the version adopted in most countries

[2] Some ICD modifications also include procedures

at the time of publication of this thesis. Given the availability of data, we have dealt with EHRs in European languages, with a special focus on Spanish. The large number of potential codes (up to 100,000 depending on version), the difference in disease prevalence, the limited access to clinical data, and the hierarchical structure with different semantic granularities in the final categories are inherent properties of ICD-10 coding that we have associated with scarcity of instances for many labels, unbalanced data distributions, and problems of generalization and semantic agreement.

First, we have focused on global and label-specific data scarcity. Patient records contain large volumes of sensitive information, so access to clinical datasets is often subject to strong restrictions imposed by privacy policies. In addition to the reduced data sets, the completeness and specificity of the standard imply lower assignment probabilities for many codes, which means fewer examples per label. Given the difficulties in inferring much of the nomenclature from the data, we have explored unsupervised techniques based on lexical and semantic matching between label representations and the EHRs. Hence, we have compared different label representations in a Vector Space Model (VSM), using an exhaustive pre-processing to deal with synonyms. Alternatively, we have proposed a similarity method that exploits the structure of SNOMED CT to identify evidence sets with different levels of abstraction.

Second, we have addressed the enormous imbalance of both academic and hospital data collections. Computationally, ICD-10 coding implies a multi-label classification with more than 10,000 categories, but the number can be increased up to 140,000 with ICD modifications such as the Spanish version CIE-10-ES and the US version ICD-10-CM. In addition to the huge label space, health conditions have different prevalence among populations, with few frequent and many rare diagnoses, leading to extreme distributions of data. Therefore, we have explored data augmentation methods to improve the representability of minority codes by applying lexical substitution and Machine Translation (MT) techniques. We have also proposed the use of Extreme Multi-label Text Classification (XMTC) methods that exploit label co-dependencies to improve the inference of the least frequent codes while tackling scalability issues.

Thirdly, we have addressed the meaning differences between categories and documents. Coding is a high-level task that requires an extensive semantic knowledge of the biomedical domain to deal with the varying degrees of abstraction. The standard is designed for statistical purposes to group clinical concepts, which poses generalisation problems during learning from examples. Thus, we have explored multiple transfer learning methods to introduce external knowledge to the task. We have conducted experiments to examine the effect of MT techniques on the implementation of cross-linguistic approaches. Moreover, we have explored the generation and application of in-domain vector representations for Spanish records. Furthermore, we have proposed to pre-train models on higher hierarchical categories to incorporate general features in the learning of the final categories.

Finally, we have noticed that the techniques are interdependent and complementary in many cases. For this reason, we have addressed all challenges simultaneously with a combination of: semantic similarity-based methods to predict codes not included in the examples; data augmentation methods to generate new examples for minority codes and reduce imbalance; ensembles of XMTC algorithms to promote minority codes while reducing computational cost; and word embeddings pre-trained on EHRs and medical PhD theses to introduce general knowledge and improve generalisation during learning.

As a result, we have implemented an approach that achieves significant improvements over other explored conventional approaches. In particular, we have noted that XMTC algorithms together with data augmentation techniques are the most suitable approaches for ICD-10 coding. In fact, bagging XMTC algorithms produce the largest increase in scores for frequent codes, while Lexical Substitution techniques are the foundations for the largest improvement in predicting underrepresented codes. Conversely, unsupervised methods are the only approaches capable of predicting unrepresented codes. The assumptions and conclusions made for these methods are related to the inherent nature of classification and are easily extrapolated to the other ICD versions.

INDEX

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACS Average Cosine Similarity 203, 205

AdaBoost Adaptive Boosting 118, 138, 161

AI Artificial Intelligence 2, 25

ANNS Approximate Nearest Neighbour Search 156

BCE Binary Cross Entropy 139, 144, 145, 149, 158–160, 192

BERT Bidirectional Encoder Representations from Transformers 24, 28, 29, 33–35, 38, 92, 117, 124, 125, 180, 184–187, 207, 209, 210, 220, 232

BNS Bi-Normal Separation xvii, 131, 132, 192

BoW Bag-of-Words 26, 90, 97, 103, 124, 125, 130, 138, 158, 161, 162, 166, 172, 186, 187, 189

BPE Byte Pair Encoding 178, 197, 200

CAGR Compound Annual Growth Rate 2

CCS Clinical Classifications Software 35

CIE Clasificación internacional de enfermedades 63, 94

CIE-10-ES Clasificación internacional de enfermedades, 10.ª revisión, edición española ii, v, xii, xv, xvi, 11, 23, 29, 36, 37, 50–61, 63, 64, 78, 83, 87, 94, 112, 207–214

CLEF Conference and Labs of the Evaluation Forum 23, 24, 27–31, 44, 48, 51, 73, 236

CNN Convolutional Neural Network xiii, 28, 29, 31, 32, 38, 138, 148, 149, 159, 161, 162

CodiEsp Clinical Case Coding in Spanish Shared Task vii, xii, xv, xvi, 24, 29, 41, 44, 50, 52, 54–56, 66, 73, 81–84, 96, 99–102, 106–109, 242

CoVe Context Vectors 181

CREATE Cohort Retrieval Enhanced by Analysis of Text from Electronic Health Records 91

CSCE Contextualised Spanish Clinical Embedding xiv, xv, 197, 200–202, 209, 210, 256

CépiDc Centre d'épidémiologie sur les causes médicales de décès 45

DCG Discounted Cumulative Gain 76

DRG Diagnosis Related Group xi, 7, 9, 10

EHR Electronic Health Record 6, 89–91, 93, 139, 149, 150, 157, 160, 212

EHRs Electronic Health Records i–vi, xiii, xvii, 5, 13, 25, 29, 37–39, 84, 87–91, 93, 94, 96–98, 112, 113, 126–129, 131, 133, 137, 139, 141, 157–159, 162, 169, 171, 186, 187, 207, 230–232

ELECTRA Efficiently Learning an Encoder that Classifies Token Replacements Accurately 181

ELMo Embeddings from Language Models 31, 35, 180, 184, 186

EMR Electronic Medical Record xii, xv, 57, 61, 62, 244, 245

EMRs Electronic Medical Records xvi, 5, 44, 51, 53, 55, 63, 65, 66

FAEs Finished Admission Episodes 2, 3

FCEs Finished Consultant Episodes 2

FCNN Fully Connected Neural Network xiii, 139, 140, 188, 190, 207

GANs Generative Adversarial Networks 32

GCNNs Graph Convolutional Neural Networks 31, 32

GDPR General Data Protection Regulation 4, 60

GEMs General Equivalence Mappings 28

GPT Generative Pre-trained Transformer 180, 184

GRNNs Graph Recurrent Neural Networks 32

GRU Gated Recurrent Units xiv, 28, 31, 32, 35, 206, 207

HAL Hyperspace Analogue to Language 177

HAN Hierarchical Attention Network xiv, xvii, 29, 32, 33, 197, 205, 206, 208, 209

HIPAA Health Insurance Portability and Accountability Act 4, 66

HUFA Hospital Universitario Fundación Alcorcón vii, xii, xv, xvi, 41, 44, 57–64, 66, 67, 81–84, 87, 88, 96, 99–102, 106–109, 115, 133, 157, 170, 177, 200, 207, 208, 210, 213, 223, 244, 245

IC Information Content 77, 78, 104, 105

ICD International Statistical Classification of Diseases and Related Health Problems i–vii, xi–xiv, xvii, 2, 6, 7, 9–11, 13, 15, 17, 19, 21–27, 29–35, 37–39, 42–46, 48–52, 63, 65, 66, 71, 72, 74, 75, 77, 83, 84, 87–94, 96, 98, 99, 103, 106, 108, 109, 112, 113, 116, 125, 126, 129, 137, 138, 157, 166, 169, 170, 173, 175, 185–187, 197, 205, 209, 211, 215–217, 220, 230, 232–234

ICD-10 Tenth revision of the International Statistical Classification of Diseases and Related Health Problems ii–vi, xi, xiv, xv, 2, 9–11, 13, 15–17, 19, 23, 24, 26–31, 33–35, 37–39, 42–48, 50, 51, 63, 72–75, 83, 84, 88, 110, 113, 165, 169, 170, 172, 173, 175, 186–188, 211, 217, 220, 222, 225, 230–232, 234, 240

ICD-10-CM International Classification of Diseases, Tenth Revision, Clinical Modification ii, v, 51

ICD-10-PCS International Classification of Diseases, Tenth Revision, Procedure Classification System 51

ICD-9 Ninth revision of the International Statistical Classification of Diseases and Related Health Problems 23, 26–31, 33–35, 187

ICD-O-3 International Classification of Diseases for Oncology, Third Edition 31, 173, 186

ICF International Classification of Functioning, Disability and Health 89

IDF Inverse Document Frequency 97

INE Instituto Nacional de Estadística 67

INSERM French institute for health and medical research 191

IoT Internet of Things 4

IR Information Retrieval viii, 10, 22–24, 27, 30, 33, 39, 57, 85, 87–93, 96, 97, 99–103, 106, 107, 109, 162, 224, 226

ISTAT Istituto nazionale di statistica 45, 191

KLD Kullback-Leibler Divergence 96, 99, 102, 103, 106

KNN K-Nearest Neighbor 23, 26, 27, 30, 98, 100–102, 109, 117, 118, 121, 122, 138, 145, 151, 158, 163

KSH Központi Statisztikai Hivatal 45, 191

LCS Least Common Subsumer 34, 78, 104, 105

LDA Latent Dirichlet Allocation xiii, 23, 26, 34, 118, 138, 145, 146, 158

LM Language Model ix, 29, 30, 39, 91, 92, 117, 118, 125, 168, 171, 178, 180, 181, 184, 198, 200, 210, 231

LOINC Logical Observation Identifiers Names and Codes 8

LR Learning-to-Rank 26

LSA Latent Semantic Analysis 27, 92, 177

LSTM Long Short-Term Memory xiii, 28, 29, 31–33, 124, 138, 141–143, 154, 157–159, 161, 162, 180, 200, 232

MCMC Monte Carlo Markov Chain 145

MeSH Medical Subject Headings 8, 30, 89, 91

ML Machine Learning 10, 13, 15, 16, 27, 30, 34, 38, 39, 70, 87, 98, 112, 114, 117, 138, 161, 169, 173, 183, 210

MLP Multi-Layer Perceptron 138, 139, 158–162

MRF Markov Random Field 91

MT Machine Translation ii, iii, v, 28, 30, 39, 112, 113, 117, 126, 128, 129, 135, 165, 170, 176, 180, 181, 183, 184, 186, 188, 189, 191, 203, 216, 230, 231

NCI National Cancer Institute 31

nDCG normalized Discounted Cumulative Gain 76, 108, 123, 232

NER Named Entity Recognition 29, 89–91, 103, 108, 109, 171, 183, 184

NHS National Health Service xi, 2, 3

NLM National Library of Medicine 8

NLP Natural Language Processing xiv, 6, 22, 27, 30, 34, 36, 38, 65, 68, 70, 83, 89, 90, 93, 97, 169, 174, 175, 177, 181, 183, 185

NLTK Natural Language Toolkit 68

NTCIR NII Testbeds and Community for Information access Research 23

OOV Out-of-Vocabulary 134, 178, 186, 197, 205, 209

OPCS Classification of Interventions and Procedures 9, 10, 23

OPCS-4 Fourth revision of the Classification of Interventions and Procedures 51

OvR One-vs-Rest 26, 28, 30, 118, 119, 124, 125, 129, 136, 138, 139, 141, 147, 161, 165, 190

PCA Principal Component Analysis 179, 200

PhD Philosophiædoctor iv, vi, 197, 233

PLT Probability Label Tree 119, 122, 123, 138, 139, 151–153, 159, 162, 166, 231

POS Taggers Part-Of-Speech Taggers 5

POS Tagging Part-Of-Speech Tagging 26, 90, 183, 184

PPP Purchasing Power Parity xi, 3, 4

RBF Radial Basis Function 100, 135

RCNN Recurrent Convolutional Neural Network xiii, xvii, 138, 144, 161, 162, 197, 205, 208, 209, 212

RegEx Regular Expression 67

RI Random Indexing 33, 92

RQ Research Question 15, 16, 39, 88, 108, 110, 113, 165, 166, 170, 217, 220, 225, 226, 232

RQs Research Questions 15, 17, 19, 40

SBWC Spanish Billion Word Corpus 31, 182, 197–199, 201, 205, 209, 258

SCE Spanish Clinical Embedding xiv, xv, 197–201, 205, 209, 212, 223, 233, 254, 258

SciELO Scientific Electronic Library Online 182, 198

SGD Stochastic Gradient Descent 124, 200

SHE Spanish Health Embedding 182, 197, 201, 205, 209, 258

SNLI Stanford Natural Language Inference 180

SNOMED CT Systematized Nomenclature of Medicine - Clinical Terms ii, v, xi, xiii, 7–10, 31, 72, 89–91, 103, 104, 126, 127, 135, 165, 186, 198, 202, 234, 235

SOTA State of the Art vii, 7, 10, 13, 15, 19, 21, 22, 26, 37–39, 90, 92, 109, 114, 116, 126, 165, 171, 177, 181, 185, 220, 221, 225, 232, 233

STS Semantic Textual Similarity xvii, 181, 203–205, 208, 258

SUC Spanish Unannotated Corpora 182, 197–199, 201, 205, 209, 212, 223, 233, 258

SVM Support Vector Machines xiii, 23, 26–28, 30, 33, 34, 98–103, 106–109, 123, 129, 133, 135, 138–140, 158, 161, 162, 223

TF Term Frequency 97

TF-BNS Term Frequency-Bi-Normal Separation 130, 131, 138, 139, 189, 190, 235

TF-IDF Term Frequency-Inverse Document Frequency 33, 90–92, 97, 98, 117, 124, 138, 150–152, 159, 160, 234, 235

ULMFiT Universal Language Model Fine-Tuning 184, 186

UMLS Unified Medical Language System 9, 30, 33, 89, 91, 186

USA United States of America 51, 66

VSM Vector Space Model ii, v, 27, 30, 90–92, 97

WHO World Health Organization 9, 10

WMD Word Mover's Distance 203–205

XMTC Extreme Multi-label Text Classification ii, iii, v, vi, viii, xvi, 13, 18, 19, 24, 34, 35, 39, 75, 84, 111–114, 116, 118, 119, 125, 126, 138, 139, 147, 148, 150, 151, 158, 159, 161, 163, 165, 166, 186, 230–234

CHAPTER

1

INTRODUCTION

This chapter provides a context for our research, highlighting the motivation for the present thesis and describing what directions it has followed. The objectives and research questions, as well as the general structure, are also presented to facilitate a reference to the research content and purpose.

Content

1.1 Context and Motivation

This section examines the context in which computer-aided coding emerged and the factors that have motivated interest from both academia and industry. The following is a brief summary.

The amount of medical data generated by patients is growing steadily, reaching volumes that are unsustainable for manual processes. Furthermore, there is an upward trend in healthcare expenditure, which can be addressed by digital systems that automatically process patient information to provide personalised services. Implementing such processing is not trivial, as regulatory policies and the diversity of sources introduce complexity. In addition, a significant part of the relevant information collected by clinicians is often in free text, resulting in extreme variability, e.g, arbitrary use of accents, acronyms, non-standard abbreviations, typographical errors, and spelling variants. Therefore, one of the main tasks performed by hospitals is the use of clinical standards to ensure data interoperability, which is currently mainly manual. Specifically, this thesis focuses on addressing computer-assisted coding for the International Statistical Classification of Diseases and Related Health Problems (ICD), which is used to normalise diagnoses and other medical conditions such as symptoms, findings, and social circumstances, for statistical purposes. The research places special emphasis on ICD-10 coding on unstructured Spanish records, dealing with data sparsity, a huge label space, unbalanced distributions, and a very heterogeneous hierarchical semantic level.

1.1.1 Context

Digitalisation is transforming information processing in all social and business sectors, leading to the emergence of new methods and architectures based on Artificial Intelligence (AI) and big data. The amount of digital data stored worldwide continues to rise every day, so automatic techniques are ever more essential. Rydning (2018) point to 33 Zettabytes (ZB) in 2018, with a projection of 175 ZB in 2025. The same report puts the volume of healthcare data at 7% of the total in 2018, as shown in Figure 1.1. Although healthcare is not currently one of the sectors with the greatest volume of data, it is the sector that expects the greatest expansion in the coming years, with an estimated Compound Annual Growth Rate (CAGR) of 36% (Rydning, 2018). This projection is correlated with the increase in hospital activity over the last decade. For example, the National Health Service (NHS) published a recent report[1] about hospital patient care activity in England showing the upward trend in Finished Consultant Episodes (FCEs) and Finished Admission Episodes (FAEs). More details are provided in Figure 1.2, which shows an increase of 21.1% in FCEs and 15.5% in

[1] `http://digital.nhs.uk/pubs/apc1920`

FAEs from ten years ago, reaching 20.9 and 17.2 million episodes respectively. Given the fact that each episode can generate up to hundreds of pages of relevant clinical information, medical expenditure is expected to skyrocket, resulting in the optimal environment for the application of automatic processing.

Figure 1.1: *Impact of industrial sectors on digital data – adapted from Rydning, 2018.*

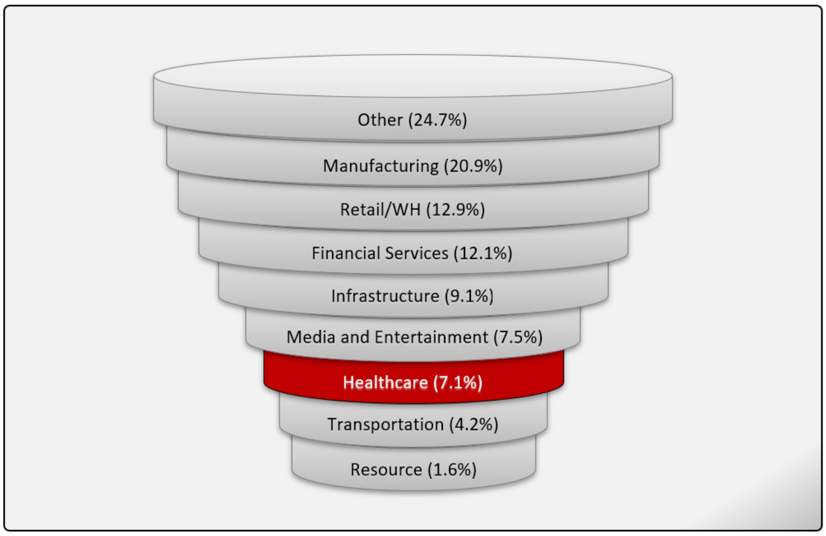
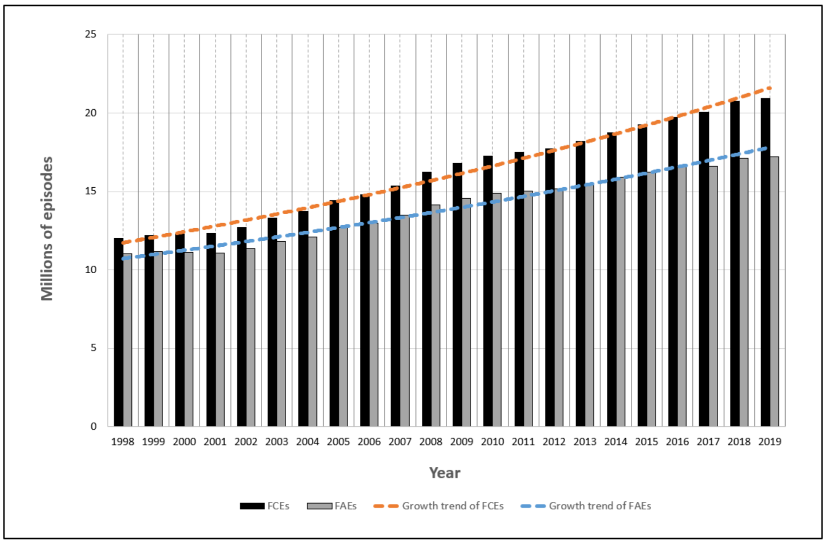
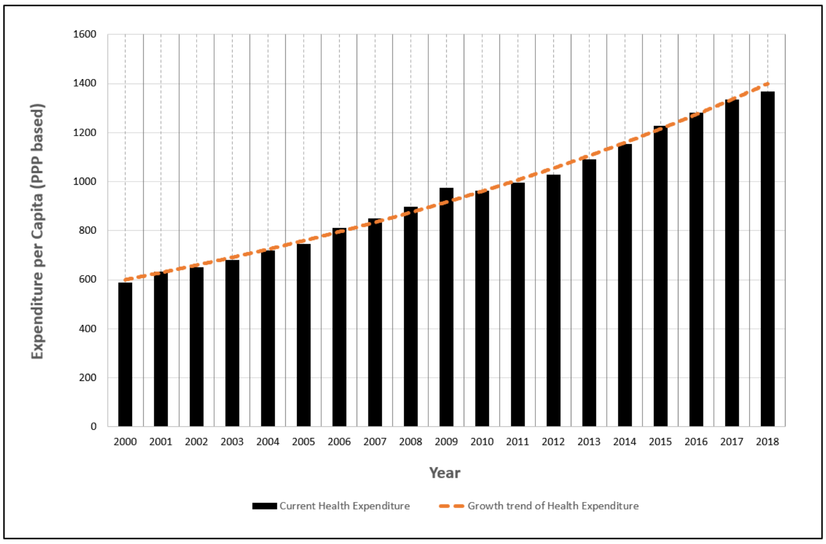


Figure 1.2: *Trends in Hospital Admitted Patient Care Activity in the NHS.*



In addition to data growth, better healthcare services are provided every year, strongly supported by medical innovations, but this is a goal that has so far been associated with the increase in health spending (Organization, 2017). As shown in Figure 1.3 in terms of Purchasing Power Parity (PPP), per capita health expenditure has increased worldwide over the last decade. An effective way to further reduce costs while improving patient outcome would be to customise patient care with the design of a digital health system (Porter and Lee, 2013). A manual examination of the increasing volume of patient-specific information is not practical as a single patient today typically generates up to 80 megabytes of graphic and textual information each year (Huesch and Mosher, 2017), so automatic analyses are needed to process all the medical and environmental patient information, applying better prevention, more effective treatments, and faster patient-doctor communication (Luo et al., 2016). In addition, digital healthcare systems would allow trend analysis on large data sets, even entire populations, to provide faster and more robust clinical studies. For this purpose, challenges such as data storage, energy costs, processing methods and data privacy must be overcome.

Figure 1.3: *Health expenditure per capita (PPP based).*



The appropriately processing of medical information is probably the most complex purpose. There is some consensus on the low digital maturity of the health sector caused by strict data protection regulations, large volumes of sources, and the great diversity of formats. Patient data is considered sensitive information and is therefore generally more protected than other types of data, as illustrated in the Health Insurance Portability and Accountability Act (HIPAA) for the United States (Ahalt et al., 2019) or in the General Data Protection Regulation (GDPR) for Europe (Mondschein and Monda, 2019). These regulations result in a range of restrictions on the storage and use of data. Furthermore, a functional health system is expected to integrate data from many different sources such as social media, web knowledge, organizations, Internet of Things (IoT) devices, research publications, laboratories, insurance providers, health centres, and government institutions. Linked to this plurality of sources, there are countless different formats including time series, human genome sequences, quantitative test results, health literature, administrative and financial information, medical images, and clinical text. Such a variety of contexts and formats hinders the design of a single, centralised strategy.

1.1.2 Motivation

Only a minor portion of health data is typically structured in such a way that it can be easily and directly interpreted and used for data mining. For this reason, one of the most demanding clinical challenges is the transformation of unstructured information into well-defined data. Although clinical reports are still collected on paper in some countries, the trend is to use increasingly advanced digital formats, from Electronic Medical Records (EMRs), which are designed like traditional documents, to Electronic Health Records (EHRs), which also contain structured fields and are enriched with different data sources. In order to facilitate information management, modern health centres try to automatically capture structured data related to the patients' care, such as gender, age, discharge date, and laboratory tests. Nevertheless, clinicians need to express observations and conclusions in a flexible manner to preserve the complexity and nuances involved in each clinical history (Ford et al., 2013; Rosenbloom et al., 2011). Therefore, a large amount of data related to diagnoses, medications, mood information, or patient history remains as images and, primarily, as text, in order to flexibly provide details. Estimates suggest that 80% of the content is unstructured (Grimes, 2008; Murdoch and Detsky, 2013). As a counterpart, the versatility of the language is linked to greater variability, so that text can include typing errors, wrong syntactical structures, synonyms, abbreviations (Barrows Jr, Busuioc, and Friedman, 2000), or ambiguity, which make an automatic processing more difficult (Edinger et al., 2012).

Specifically, clinical text involves a particular complexity in terms of spelling, lexis, and syntax (Di Renzo, 2020). Thus, clinical language tends to recurrently include elements associated with diversity and ambiguity, such as acronyms, abbreviations, symbols, and typographical errors (Ive et al., 2018). Records are characterised by freedom in the application of accent and punctuation marks and the use of capital letters for emphasis. In terms of lexis, the language relies on specialised terminology that is not accessible to all users. Such terminology is riddled with sub-technicisms, neologisms, and English words (in the case of non-English languages) (Di Renzo, 2020). As for the syntax, the domain is characterised by a variety of syntactic styles, including verbose paragraphs with numerous connectors, more typical of articles, and short sentences lacking syntax, more typical of statements in records. This heterogeneity hampers the inference of syntactic structure, which is reflected in poorer performance of Part-Of-Speech Taggers (POS Taggers) (Ferraro et al., 2013).

In addition, nominalisations, impersonal forms, and grammatical errors predominate in the texts. As regards the peculiarities of Romance languages such as Spanish (Akhtyamova et al., 2020), the predominant characteristics are the arbitrary use of accent marks, presence of Greek-Latin prefixes such as "*intravenoso*" and "*endovenoso*", free use of hyphens between words such as "*beta-caroteno*" and "*beta caroteno*", alternation between pertaynyms such as "*bacterial*" and "*bacteriano*", ambiguity of gender such as "*el tiroides*" and "*la tiroides*", and coexistence of English abbreviations such as "*PSA*" (Prostate-Specific Antigen), preferred to "*APE*" ("*Antígeno Prostático Específico*").

In the past decade, there has been increasing interest in transforming clinical text into structured data by coupling EHR systems with core coded clinical thesaurus. Transformation into semantically standardised data is needed to manage the unstructured information (Kreuzthaler et al., 2017), so ontologies and thesaurus could be a vital component to efficiently facilitate communication among healthcare professionals and support clinical practice (M. Cowie et al., 2001). Thus, different countries are developing infrastructure for national health information by implementing standards, nomenclatures, codes, and vocabularies with the aim of producing open, standard, and interoperable EHR systems (Häyrinen, Saranto, and Nykänen, 2008). In this line, clinical terminologies are key components to standardise expressions in entities, being particularly useful for supporting many processes such as the development of clinical guidelines focused on the treatment of specific conditions, retrieval of relevant data for the comparison of local and national patient care, clinical audit and outcomes studies, and decision assistance systems (Stuart-Buttle et al., 1996). Such a diversity of purposes implies the design of terminologies with different granularities, which hampers a possibly interoperability between them.

The ultimate purpose of this thesis is to contribute to the digital conversion of the health sector by automatically encoding information from medical records in natural language via computer-assisted ICD coding. In particular, research has focused on the standardisation of Spanish diagnoses and procedures, considering exclusively the unstructured data, which can be considered a summary of the content and aims to facilitate the interpretation of records and a subsequent automatic analysis. To this end, different Natural Language Processing (NLP) techniques are explored to synthesise all possible textual evidence of clinical events, requiring a decision-making process based on the most important aspects of the clinical history, physical and mental examination.

1.2 Scope

This section introduces the characteristics of ICD coding, describing the role in relation to the rest of the elements of the clinical information flow. A summary of the content is described below.

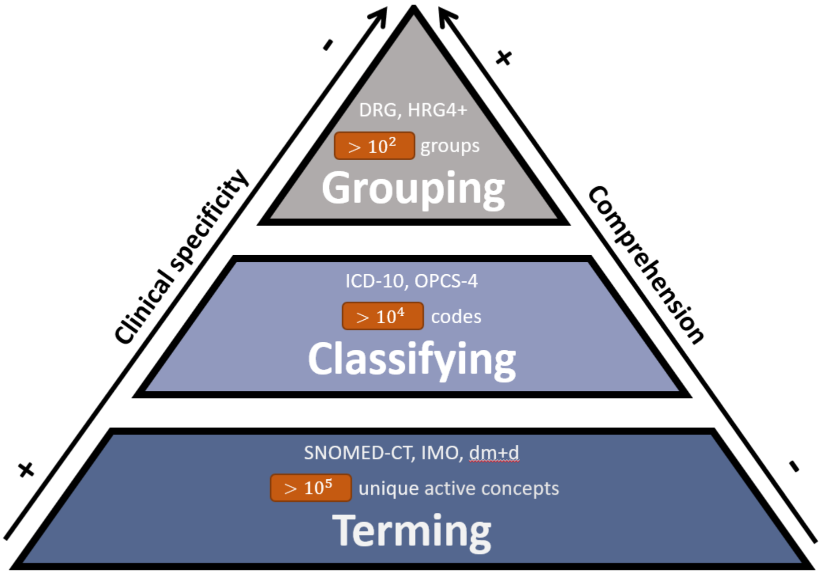
Clinical information flow is constituted on three types of standards with different clinical specificities and purposes: terming includes the most extensive terminologies such as SNOMED CT, which are used for decision support systems; classifying such as ICD coding is designed to produce statistics for comparative purposes, aggregating related concepts under the same category; and grouping comprises standards such as DRG, unifying patient events with similar costs for funding purposes. There are one-way terming-classifying and classifying-grouping maps that establish equivalences between lower- and higher-layer terms, which require additional contextual information. Among all the standards, ICD is one of the key points in the flow as it is versatile and provides the basis for grouping and estimating reimbursement.

ICD is widely used in the health systems of most countries. The hierarchical structure facilitates the definition of different levels of specificity, which are reflected in the identification tags, and establishes a versatile taxonomy for diagnoses and other clinical events. However, the coding task is hampered by the complex ICD distributions, characterised by huge, unbalanced, and sparse label space. In addition to the inherent challenges of the clinical language (discussed in Section 1.1), differences in semantic abstraction between documents and ICD categories, or between the codes themselves, result in poorer generalisations. All the mentioned constraints has contributed to the limitation of the development of computer-aided tools based on State of the Art (SOTA) methods, as the exploration of effective techniques is still at a less advanced stage than other tasks. This context has motivated the current thesis, which aims to point out the shortcomings of current techniques, generally focusing on some but not all of the above challenges, and to suggest which might be the best directions to follow.

1.2.1 Clinical information flow

The Language of Health was originally designed in the 1990s and constitutes the infrastructure that ensures shared information flow and global standards to enable inter-computer communication (Stuart-Buttle et al., 1996). Since then, it has been used to improve quality, increase service volume and boost the effectiveness of resources. The language of health describe the three main constituents comprising the data flow required for direct and indirect patient care by health care providers: *terming*, *classifying*, and *grouping*. Figure 1.4 shows a representation of the three constituents, each of which involves different granularity and specificity degrees (Cimino, 1996). Multiple standars are organised at each layer in order to provide different levels of aggregation. Upper layers include more abstract concepts and narratives, so that understanding and completeness is gained at the expense of losing clinical details. The more general and vague the concepts, the lower the granularity and the greater the number of meanings encompassed by the categories.

Figure 1.4: *Clinical information flow: Pyramidal representation for the aggregation processes through which clinical information flows – adapted from Stuart-Buttle et al. (1996).*



Terming is designed for clinicians to be used as a front-end data standard interface to accurately identify clinical concepts in terms. This has a magnitude of hundreds of thousands of terms, focused on clinical guidelines and decision support systems. Terming can be considered the most fundamental building block for any set of clinical data. There are multiple extended terminologies such as Logical Observation Identifiers Names and Codes (LOINC) (McDonald et al., 2003), Medical Subject Headings (MeSH) (Lipscomb, 2000), and Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) (Donnelly, 2006). In particular, LOINC is designed for health measurements and observations, especially useful for standardising medical laboratory results. MeSH is used to index articles for PubMed by the National Library of Medicine (NLM) (Drazen and Curfman, 2004). And lastly, SNOMED CT is a comprehensive standard reference that supports both general and highly specific concepts. It has become the most widely used terminology and is considered the richest and most complete. Each concept is defined by a set of attribute-value pairs or relationships, comprising one code per meaning, one meaning per code.

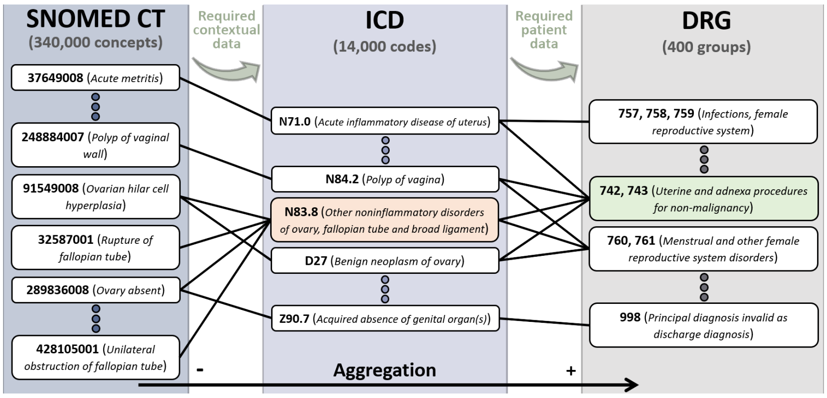
In contrast, *classifying* has focused on an easy storage, retrieval and analysis of health information for data comparisons that can provide evidenced-based decision-making. These criteria facilitate the statistical monitoring of the incidence and prevalence of factors influencing health status such as diseases, injuries, and symptoms. Classifying comprises tens of thousands of categories, offering an intermediate level of aggregation which is useful for statistical analysis of trends, epidemiology, and service management. The most widespread classification systems are ICD (WHO, 2004), and Classification of Interventions and Procedures (OPCS) (HSCIC, 2014). Both standards tend to be more general, involving multiple meanings per code. In fact, ICD includes residual categories (*other specified* or *unspecified*) to hold those diagnoses which do not fit any of the specific ones.

Ultimately, *grouping* is designed for administrative staff, managing reimbursement for provided health services. Grouping involves few thousands of high-level groups to manage a larger volume of resources by supporting service planning, contracting, and commissioning purposes. Diagnosis Related Group (DRG) is the most widely used health classification system for standardising prospective payment to hospitals and encouraging cost containment initiatives (Wiley, 2014). It covers all charges associated with inpatient stays, from admission to discharge, including services performed by outside providers. Hence, a basic DRG code is characterized by patient information and a combination of diagnoses derived from the ICD and procedures listed in the OPCS. At the most aggregated level of clinical language, the main idea is to gather patient events that have been judged to consume a similar level of resource – e.g., all neurological eye disorders (217 ICD-10 codes) are assigned to DRG code 123.

Health standards are designed with different criteria for separate purposes, with terming, classifying, and grouping being mutually complementary processes. Since terms from lower layers can be theorically related to upper-layer categories, maps exploiting the aggregation process to establish one-way equivalences have been created, as illustrated in Figure 1.5. For example, Unified Medical Language System (UMLS) provides mapping resources between multiple clinical terminologies (Bodenreider, 2004), e.g., equivalencies between a portion of SNOMED CT terms and some ICD and OPCS codes. Over 30 unique SNOMED CT terms are gathered in the ICD code N83.8 (*Other noninflammatory disorders of ovary, fallopian tube and broad ligament*). Similarly, the OPCS code W06.8 is described by *Other noninflammatory disorders of ovary, fallopian tube and broad ligament*, but there are over 200 bones in the body. Sometimes equivalences are not direct but require additional information. In particular, the move from terming to classifying need contextual and non-clinical knowledges, which are also difficult to categorize. On the contrary, the classifying-grouping step is generally used as the primary method of grouping as these maps only require patient data, which can be more readily identified. The World Health Organization (WHO) estimates that more than 3 billion dollars are annually allocated based on ICD.

Among all components of the clinical information flow, ICD and OPCS play a major role in medical services. Both standards are used for the semi-automatic generation of groups involved in funding, and for supporting the interpretability of records at

Figure 1.5: *Aggregation process using SNOMED CT to ICD-10 map and ICD-10 to DRG map.*


SNOMED CT
(340,000 concepts)
Required contextual data
ICD
(14,000 codes)
Required patient data
DRG
(400 groups)
37649008 (Acute metritis)
248884007 (Polyp of vaginal wall)
91549008 (Ovarian hilar cell hyperplasia)
32587001 (Rupture of fallopian tube)
289836008 (Ovary absent)
428105001 (Unilateral obstruction of fallopian tube)
N71.0 (Acute inflammatory disease of uterus)
N84.2 (Polyp of vagina)
N83.8 (Other noninflammatory disorders of ovary, fallopian tube and broad ligament)
D27 (Benign neoplasm of ovary)
Z90.7 (Acquired absence of genital organ(s))
757, 758, 759 (Infections, female reproductive system)
742, 743 (Uterine and adnexa procedures for non-malignancy)
760, 761 (Menstrual and other female reproductive system disorders)
998 (Principal diagnosis invalid as discharge diagnosis)
-
Aggregation
+

the diagnostic and procedural level. For this reason, improving both the speed and quality of coding is a key step in optimising the flow. Although it is a tedious process given the volume of data, qualified ICD specialists deal with the bulk of the task, involving considerable expenditures. Computer-assisted coding can provide evidence to aid classification, but is far from solving the task, as it requires complex reasoning processes to synthesize the information into the correct diagnosis or procedure. The present thesis tackles coding using Information Retrieval and Machine Learning SOTA techniques to support specialist.

1.2.2 ICD-10

ICD is a clinical classification standard supported by the WHO[2] for statistical analyses of morbidity and mortality. More than 11,000 diseases, abnormal findings, complaints, social circumstances, external causes of injury, signs, and symptoms are described in the tenth revision (ICD-10[3]). Hereafter, we will refer to all these clinical events as diagnoses. The aforementioned version is used by more than 150 countries around the world and has been translated into more than 40 languages. In turn, some countries such as United States, France, and Spain have implemented extensions by increasing the specificity of the codes and including the OPCS. According to the WHO, ICD has been cited in more than 20,000 scientific articles in 30 years. The journal index from the Web of Science[4] collects an increasing trend of publications and citations of ICD-related articles in the area of computer science, as shown in Figure 1.6.

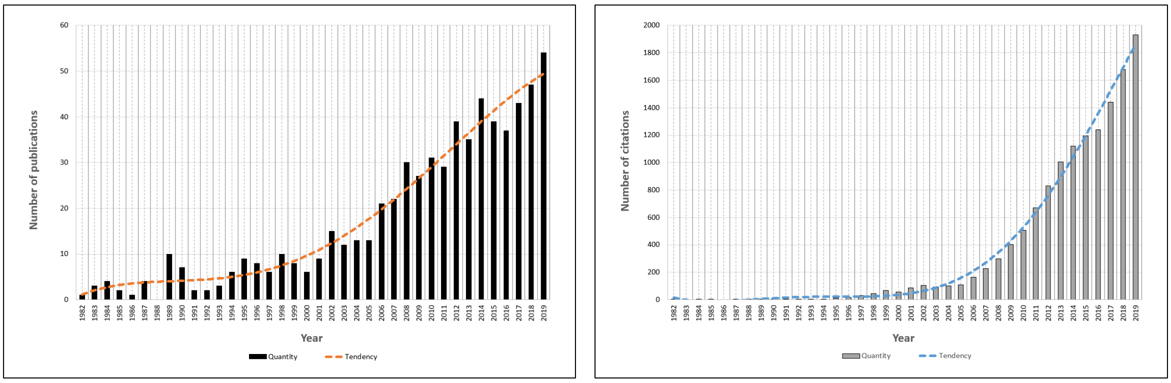
The ICD-10 is a taxonomy that categorises health concepts in a nested hierarchical

[2] `https://www.who.int`

[3] https://icd.who.int/browse10/2019/en

[4] `https://support.clarivate.com/ScientificandAcademicResearch/s/article/Web-of-Science-Access`

Figure 1.6: *Trend of publications and citations of ICD-related articles in the area of computer science – consulted on the Web of Science.*



(a) *Publications*

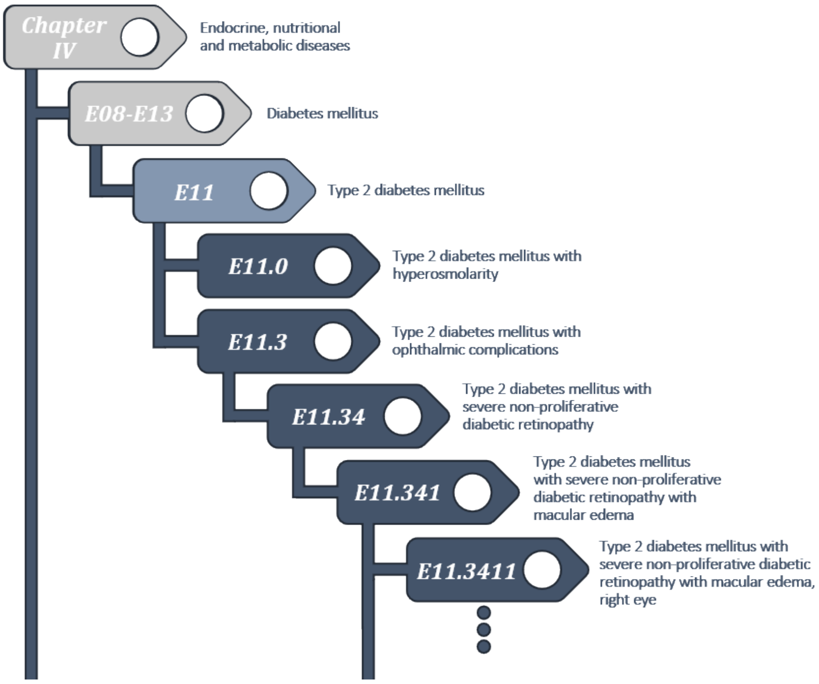
(b) *Citations*

structure, associating each one with a unique code of 3 or 4 alphanumeric characters. The codes comprise diverse information such as descriptions, related terms, and additional rules, e.g. inclusion and exclusion terms, excluded and additional codes, and priority order indicating which code comes first. The standard is based on chapters and sections, followed by 3-character codes. Beyond these, each new, more specific node in the branch increments by one character, so that the categories with fewer characters imply more generality and are placed on top of the classification. It should be noted that only the final codes are used for coding. For this reason, there are residual categories that encompass all those events corresponding to the parent category but which do not fit into the other final codes in the same branch. Moreover, not all branches have the same depth, so there are 3- and 4-character final codes. Although the reference version includes approximately 14,000 codes, other modifications such as the American version (ICD-10-CM) raise that number to more than 100,000. Similarly, the Spanish version (CIE-10-ES[5]) extends the specificity of the hierarchical structure with 5-, 6- and 7-character codes, increasing the amount to approximately 69,000 diagnoses and 72,000 procedures (notice that ICD-10 does not contain procedures). For example, Figure 1.7 shows the connection among several codes of the same family, *Type 2 diabetes mellitus*.

ICD-10 coding entails great difficulty: huge volume of potential solutions, scalability issues, high biases, data scarcity, significant imbalance, and notable differences in abstraction. Although all these problems have been described below according to the 10th ICD revision, they are inherent to ICD coding. The more recent the version, the more specificity is introduced, which accentuates each of the above-mentioned problems.

[5] `https://eciemaps.mscbs.gob.es/ecieMaps/`

Figure 1.7: *CIE-10-ES hierarchical structure: Nodes below code E11 in Chapter IV.*


Chapter IV
Endocrine, nutritional and metabolic diseases
E08-E13
Diabetes mellitus
E11
Type 2 diabetes mellitus
E11.0
Type 2 diabetes mellitus with hyperosmolarity
E11.3
Type 2 diabetes mellitus with ophthalmic complications
E11.34
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy
E11.341
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy with macular edema
E11.3411
Type 2 diabetes mellitus with severe non-proliferative diabetic retinopathy with macular edema, right eye

Firstly, the immense amount of codes involves **scalability** concerns that complicate the viability of real systems. In addition, such a volume results in a large number of syntactically valid candidates for each record, which requires accurate context-sensitive filtering for proper code assignment. Specifically, coders often need to select the most appropriate codes from among hundreds of seemingly suitable possibilities when analysing a record (Arifoğlu et al., 2014). It is a sparsity label space as only a few codes are finally assigned to a document among thousands of defined ones. Significant biases are also common in health care institutions as a consequence of the strong dependency on local factors, such as the environmental conditions and lifestyles. For example, environmental circumstances influence the incidence of certain conditions such as hypothermia and socio-economic factors predispose to disorders such as obesity. Moreover, all health facilities do not have the same resources and do not offer the same services, so certain data are not directly generated. Furthermore, diagnostic definitions change over time and countries, modifying the criteria for detection (Council, Population, et al., 2011). The large number of codes and the presence of biases are coupled with the access to data severely limited by privacy policies, restricting the number of examples per code in relation to the total volume and resulting in a large scarcity in label space (**data scarcity**).

Besides that, the very nature of the diagnoses and procedures leads to large differences in prevalence, so that certain events are manifested very frequently among patients, while others are rarely reported. Such disparities leads to pronounced unbalanced data sets (**label imbalance**), i.e., collections with a few very popular codes (orders of hundreds) and many rare ones (orders of thousands). The imbalance especially complicates Machine Learning (ML) methods, which tend to promote majority classes and underestimate under-represented classes, which are processed as noise. All the above-mentioned label space attributes (large, sparse, and unbalanced) constitute extreme data distributions that follow exponential rather than uniform histograms. This task falls within the scope of Extreme Multi-label Text Classification (XMTC), where one of the main challenges is dealing with the scalability of the solutions due to the large number of labels involved.

As for the semantic challenges, a higher level of information aggregation is typically reflected by the ICD-10, which differs from the text within records. Hence, codes tend to be more abstract, grouping together many disparate terms and demanding a deep clinical semantic knowledge. Moreover, the coexistence of final codes with varying numbers of characters, i.e., specificity, implies dealing with different semantic levels simultaneously during coding. For example, the less common diseases are clustered into general categories such as *not otherwise specified* (NOS) codes, which are used in cases with insufficient information for more specific codes, and *not elsewhere classified* (NEC) codes, comprising cases with more specific information but not covered by existing ones. Both aspects complicate the identification of common features per code and lead to generalisation issues, which demands the introduction of external knowledge (**limited generalisation**). Moreover, the task is carried out at the document level, and although records contains lexical expressions that could locally be associated with some category, disseminated information is required to propose the final codes. Thus, ICD-10 coding can be considered a multi-label classification of one-to-many, with a wide range of possibilities. There are some document types such as death certificates that are short, but most of them consist of long documents with a tendency to comprehensively collect all patient information. The great length of documents in multi-label classification severely limits the scalability of the proposals.

The rich semantic diversity and all features related to the extreme distribution of data severely complicates the production of high quality automatic results. For this reason, coding is performed with human intervention even though the automation is a key priority in most health institutions, as the individual analysis of clinical notes by highly trained health specialists (coders) involves considerable financial resources. Computer-assisted ICD coding proposals generally do not tackle the challenges outlined above. Instead, State of the Art (SOTA) methods do deal with the issues but individually, with a tendency to simplify the experiments by either reducing the code space, focusing on the most frequent ones, or reducing the size of EHRs. This thesis aims to tackle the ICD-10 coding (currently in force in most countries) using computer science methods to cope with each of the particularities.

1.3 Research questions

ICD-10 coding is a multi-label classification task whose main features are a huge label space with a shortage of examples, a very diverse range of code probabilities following a power-law distribution, and a hierarchical unbalanced tree-like architecture defining differing semantic specificity levels. In published research, it is common to simplify the problem to focus on one of the challenges given the complexity of the problem. We instead aim to address all these challenges in order to deal with the real task. To this end, this research is conducted using different data sets, but focusing on a collection of Spanish hospital discharge reports.

The aforementioned challenges motivate the Research Questions (RQs) on which this thesis has been built and which have conducted all the research, with the main RQ being posed as follows.

Research Question 1

Which are the best techniques for approaching ICD-10 coding in response to the challenges posed by the task?

One of the main challenges is to cope with data scarcity for many of the codes defined in the standard. In fact, a large percentage of the events from the nomenclature are often not reported in the collections, either because of generation biases or due to reduced probability. Although supervised approaches for zero-shot label prediction are being explored, they are currently neither efficient nor scalable. Instead, we think that scarcity demands the exploration of unsupervised techniques to exploit non-example based resources such as ICD descriptions and in-domain knowledge bases. Whereas coding is a complex activity that not only consists of gathering information but requires a synthesis process to weigh each piece of evidence, modelling such complexity in an unsupervised way is hardly effective. Nevertheless, it seems the most viable choice to address those zero-shot codes given the current SOTA. The following question arises concerning unsupervised proposals, with a particular emphasis on the use of expert knowledge from ontologies.

Research Question 2

Is it possible to approach ICD-10 coding using unsupervised techniques in a way that can be a competitive alternative to supervised methods?

Another major challenge is to overcome the issues associated with unbalanced data sets in ML methods. Imbalance prevents thorough learning for minority classes, and especially for few-shot codes, which usually represent a large percentage of the standard. The reason is that the ML models are designed to be robust to noise, so that the over-represented codes tend to monopolise the learning by skipping the sparse patterns of the lesser-repesented codes. For this purpose, class-imbalance focused techniques are required to reduce the negative effects of the extremely disparate occurrences of the codes. In turn, the extreme label space involves large computational resources, with excessive time, memory, and processing operations. Both imbalance and scalability issues could be addressed by exploiting label co-dependencies. In this context, the following question is formulated.

Research Question 3

Which techniques can increase the predictive capacity of ICD-10 codes with fewer instances while improving overall system performance? How and how much can the computational complexity of the task be reduced?

As for differences in semantic specificity, the variety of granularities lead to generalisation issues. While the more specific codes converge quickly during learning, those with a greater diversity of terms and meanings require more varied examples, often unavailable, to cover the different patterns and reach the same quality of inference. The incorporation of knowledge from external task through transfer learning methods could reinforce predictive capacity as it would be beneficial to combine information to supplement the deficiencies. This background raises the following question.

Research Question 4

Which transfer learning methods are easily applicable to ICD-10 coding and which ones are most effective in improving inference?

Finally, after answering the above questions, it is appropriate to reflect on a possible system that would simultaneously address all these challenges. Therefore, returning to the initial question, one could ask about the possible interaction between the explored methods and their collective impact on inference. Thus the last question emerges.

Research Question 5

How could alternative ICD-10 coding approaches be combined to tackle scarcity, imbalance, and generalization constraints?

1.4 Objectives

The main Research Objective of this thesis is to explore the support for ICD-10 coding, examining which artificial intelligence methods are most successful in identifying useful codes for ICD specialists on real data sets. In particular, it could be formalised as follows:

> *Research Objective*
>
> *Explore which supervised and unsupervised techniques are best suited to the particular characteristics of ICD coding, and analyse under which conditions and methods the assignment of appropriate ICD codes to free-text reports is maximised.*

The main goal will be divided into more concrete targets to organize the research, aimed at tackling the same challenges outlined in Section 1.3: the lack of data, imbalance of codes, scalability issues, and loss of generalisation.

The first step in addressing the proposed Research Questions is to establish an experimental framework including an evaluation method in accordance with the particularities of the task. To this end, we have explored experiments on several corpora, focusing on the task of coding Spanish long records. Hence, the following targets are planned:

- Define an experimental framework.
- Explore which evaluation metrics best fit the ICD-10 coding.
- Design a baseline to achieve reference results and detect the intrinsic complexities of the task.

Once the method for quantifying and comparing results has been established, the intention is to explore unsupervised algorithms to deal with data scarcity. To this end, the following guidelines are outlined:

- Propose and compare unsupervised methods based on lexical and semantic similarity to predict codes without example-based learning.
- Analyse the influence of using information from examples and impact on the poorer represented codes.
- Assess the overall and disaggregated performance of unsupervised methods.

The extreme and unbalanced label space is a barrier to learning. The following targets are therefore established to improve the predictive capacity of the minority classes and deal with scalability:

- Examine data augmentation methods to increase the instances of minority codes.
- Propose methods to promote less frequent codes by exploiting class co-dependencies.
- Explore the contribution of Extreme Multi-label Text Classification algorithms.
- Evaluate training and prediction computation times per algorithm for use in real applications.

Different semantic granularities hinder learning. The following targets aim to improve the generalisation of codes covering a greater diversity of terms:

- Examine the effects of incorporating data sets in other languages during learning on inference.
- Explore general purpose semantic representations to increase predictive capabilities and generate word embeddings and language models with the available clinical information.
- Compare the performance of retraining generic and in-domain language models, or fine-tuning, with the use of the features generated by these models in other algorithms.
- Apply hierarchical information to improve learning of common features.

Finally, all challenges should be addressed simultaneously in order to approach the task in all complexity. With regard to the combination of approaches, the following targets have been stated:

- Study the gaps and overlaps in the predictions between the approaches.
- Explore statistical methods such as voting to promote shared results.
- Perform an ablation test on the resulting combinations.

1.5 Thesis structure

This thesis is about the computer-assisted ICD-10 coding, focusing on the main attributes that characterize such classification and data sets. Central chapters (4, 5, and 6) have therefore been organised around the key challenges identified, so that the principal ideas outlined in each chapter are as follows:

Chapter 1.: A brief description of the incentives to explore automatic methods for supporting ICD coders is provided. The task is also contextualised, detailing what role the ICD plays in the clinical information flow and identifying the main attributes of the ICD coding. Based on the motivations and context, Research Questions are raised and Research Objectives are proposed.

Chapter 2.: An analysis of the most relevant work for ICD coding is provided, reviewing the State of the Art in recent decades. Some background on the evolution of the task since the beginning of computer assistance is first provided, followed by an overview of the literature that we have found to be most representative for the techniques specialised in tackling each of the challenges posed. Finally, potential gaps in the SOTA are discussed, which will lead to the proposals of the thesis.

Chapter 3.: All information concerning experimental design such as corpora, preprocessing, and evaluation metrics is collected in this chapter, e.g., statistics for the data sets used. Several evaluation metrics tailored to the task are also proposed to capture different features such as hierarchy and imbalance.

Chapter 4.: This chapter describes the use of unsupervised techniques to deal with the lack of data. The application of methods based on lexical and semantic similarities is compared. The use of structured knowledge and example-based information is also compared.

Chapter 5.: Different techniques are described to reduce the negative effect of imbalance during learning. Data augmentation techniques such as back-translation and synonym replacement are analysed. XMTC algorithms for promoting minority codes and enhance scalability are also explored.

Chapter 6.: Experiments with transfer learning techniques are described to improve the generalisation of ICD coding models. For example, the impact of cross-lingual proposals is studied. The contribution of clinical word embedding and language models is explored in comparison with general-purpose ones, generating in-domain models via text compilation. Finally, the use of hierarchical information to force the learning of common features is also explored.

Chapter 7.: The combination of techniques to tackle all the challenges at the same time is explored in this chapter. Statistical techniques and learning methods for ranking fusion are described.

Chapter 8.: The main contributions and conclusions of the research are summarized, while future lines are proposed. Some of the ideas, figures, tables, and results included in this thesis were published in scientific papers, which are listed in Section 8.4.

CHAPTER

2

RELATED WORK

Content

This chapter provides an overview of the SOTA approaches for ICD coding. Firstly, it introduces the evolution of approaches from the beginnings of computer-assisted ICD coding to the present day. Secondly, it organises some of the approaches that we have considered most representative according to the methods for addressing with the challenges of label scarcity, label imbalance, scalability, and limited generalisation.

The main objective of this chapter is to identify the research gaps in SOTA in order to propose new methods that contribute to the improvement of ICD coding.

2.1 Introduction

The standardisation of diagnoses and procedures based on patients' textual clinical evidence from hospital records (usually physician observations) is not an easy task, as it entails scenarios with strongly fluctuating free text that require dealing with high-level semantics. The most popular techniques in ICD coding have varied over time in line with the trends in the State of the Art, occasionally differing with respect to the challenges involved in the task. The earliest documented research dates back to 1968, when Howell and Loy (1968) proposed the first automated ICD coding system (8th revision[1]). This system was based on the association of codes to individual terms, selecting the final code when it exactly matches the complete text sequence. Isolated studies continued the work of Howell and Loy (1968), such as the Kodiac system (Greenwood, 1972), which increased coverage by assigning the codes that best matched despite not being exact matches. But it was not until the 1990s that computerised coding systems (9th revision[2]) became popular in scientific community (Stanfill et al., 2010).

Early coding approaches such as the "*fruit machine*" (Howell and Loy, 1968) and Kodiac (Greenwood, 1972) systems consisted of rule-based methods, generally associating expressions to codes. The manual definition of heuristics, typically based on exact matches, suffer from major limitations since they become less effective as medical coding standards have evolved by increasing specificity, e.g., moving from the 8th to the 9th ICD revision. Fortunately, computing capacity has also advanced, encouraging the exploration of more complex and elaborate methods. From purely rule-based systems, research moved on to NLP- and IR-based approaches in the 2000s, which involve advanced text processing and partial matches. These systems are more flexible by intrinsically handling the non-completeness of information in retrieving the most relevant codes. Some of the techniques applied were modelling lexical derivations and inflections with lemmatisation and stemming techniques (Erraguntla et al., 2012), morphosyntactic disambiguation using taggers and parsers (Chen,

[1] Many countries used only a short list of ICD-8 150 codes instead of the more detailed full list

[2] ICD-9-CM comprises around 14,000 codes

Barrera, and Rhodes, 2010; Erraguntla et al., 2012), and semantic analysis supported by knowledge bases (Lima, Laender, and Ribeiro-Neto, 1998; Pereira et al., 2006).

In parallel, proposals were suggested for preliminary supervised systems based on statistical information (usually word frequencies) (Gundersen et al., 1996; Lima, Laender, and Ribeiro-Neto, 1998), tipically in combination with IR methods to overcome data availability constraints (Aronson et al., 2007; Crammer et al., 2007; Lussier, Shagina, and Friedman, 2000; Pakhomov, Buntrock, and Chute, 2006; Patrick, Zhang, and Wang, 2007). But it was not until 2007, with the release of the first shared task (Pestian et al., 2007), that the supervised proposals gained greater impact on the clinical scientific community. Since then, a wide variety of supervised approaches have emerged, such as Support Vector Machines (SVM) (Dermouche et al., 2016; Kavuluru, Rios, and Lu, 2015; Perotte et al., 2014; Wang et al., 2017; Yan et al., 2010; Zhang, 2008), Naive Bayes (Dermouche et al., 2016; Kavuluru, Rios, and Lu, 2015; Medori and Fairon, 2010), K-Nearest Neighbor (KNN) (Erraguntla et al., 2012; Pereira et al., 2013; Ruch et al., 2008b; Wang et al., 2017; Yan et al., 2010), and Latent Dirichlet Allocation (LDA) (Dermouche et al., 2016; Perotte et al., 2011) classifiers. All previous approaches, with the exception of the one proposed by Dermouche et al. (2016), were designed for ICD-9 coding, involving less complexity than the 10th revision[3]. As there are hardly any freely distributable collections, researchers use different collections that are usually restricted in use, so it is not easy to compare systems. This is the reason for the emergence of evaluation campaigns, which yield publicly available data sets and facilitate direct comparisons of approaches.

Successive shared tasks such as NII Testbeds and Community for Information access Research (NTCIR) and Conference and Labs of the Evaluation Forum (CLEF) have released ICD-10 coded clinical data with which to explore computer-assisted proposals. Hence, NTCIR-11 MedNLP-2 (Aramaki et al., 2014) and NTCIR-12 MedNLPDoc tasks (Aramaki et al., 2016) have shared a collection of short Japanese medical reports, with 7 sentences on average and a total of around 550 unique codes. Alternatively, CLEF tasks have gradually evolved in complexity each year: in the first years, collections of short documents (few lines) have been published using 1, 2, and 3 languages in 2016, 2017, and 2018 respectively; medium-length documents (few paragraphs) annotated with only top-level categories (270 codes) or extended standards (CIE-10-ES) have been published in 2019 and 2020 respectively. In particular, a set of French free text death certificates comprising 3 lines on average and annotated with around 3,200 codes has been collected in CLEF eHealth 2016 Task 2 (Névéol et al., 2016). In turn, CLEF eHealth 2017 Task 1 (Névéol et al., 2017) has released a multilingual corpus with English and French death certificates. Such records are similar in size to the previous task and involve about 2,300 unique codes. In this line, another multilingual

[3] The 10th ICD revision comprises around 68,000 codes; it is sometimes combined with OPCS, which provides around 87,000 more codes

corpus is released in CLEF eHealth 2018 Multilingual Information Extraction Task (Névéol et al., 2018). The organisers describe three subsets with French, Italian, and Hungarian death certificates. Such records are also short and are associated with more than 3,000 codes in total. In contrast, the CLEF eHealth 2019 Multilingual Information Extraction Task (Neves et al., 2019) has released german non-technical summaries of animal experiments with 369 words on average per record. Such records have been coded with ICD chapters and sections. Finally, the Clinical Case Coding in Spanish Shared Task (CodiEsp) (Miranda-Escalada et al., 2020) presents a collection of clinical cases. These records have an average length of 372 words and are associated with 3,400 unique codes.

With the expansion of supervised systems and the huge increase in classes introduced by the 10th ICD revision, countless sequential deep learning models have emerged, most of them based on distributional semantic representations (Atutxa et al., 2018; Blanco, Pérez, and Casillas, 2020; Blanco et al., 2020; Miftahutdinov and Tutubalina, 2017; Ševa, Sänger, and Leser, 2018). Nervertheless, their effectiveness has been substantially restricted by the limited amount of public data. For this reason, the trend in recent years has been the exploration of general-purpose learning, typically involving language models such as Bidirectional Encoder Representations from Transformers (BERT), in order to leverage transfer learning techniques and improve model generalisation with fewer examples (Amin et al., 2019; Ji, Hölttä, and Marttinen, 2021; Manginas, Chalkidis, and Malakasiotis, 2020; Sänger et al., 2019; Silvestri et al., 2020; Velichkov et al., 2020). Even so, most of the exclusively supervised proposals for ICD-10 coding focus on a reduced set of codes (the most frequent ones) instead of the full set of codes due to challenges associated with extreme distributions. XMTC algorithms have begun to be explored with the aim of promoting underrepresented codes (Chalkidis et al., 2020; Zhang, Liu, and Razavian, 2020). To the best of our knowledge, we proposed the first study comparing multiple XMTC algorithms and proposing an ensemble (Almagro et al., 2020).

The eleventh revision (ICD-11) is currently being adopted in several countries. It has been designed with a focus on digital accessibility and automatic processing, so that the implemented knowledge base structure resembles a multilingual ontology, i.e. the described clinical entities are linked to each other (Fung, Xu, and Bodenreider, 2020). Every entity comprises attributes such as temporal, functional, treatment and causal properties, in addition to a variety of related terms, enhancing lexical diversity. In addition, the revision includes post-coordination rules to combine multiple codes into a more specific one. Thus, the use of these electronic resources in an information system would overcome some of the limitations of ICD-10, especially with regard to unsupervised methods.

In short, the scientific community initially tackled the challenge of ICD coding with unsupervised approaches due to data limitations. But such proposals fail in dealing with the intrinsic complexity that human coders do handle: unsupervised computer-assisted methods often do not apply ICD rules such as exclusions, ignore co-dependencies between codes, and are incapable of interpreting high-level semantics. As more data has become publicly accessible, more supervised approaches capable of automatically learning the corresponding heuristic have been developed. By contrast, supervised methods are typically limited to a reduced set of codes (those with sufficient examples) as biases and extreme distribution prevent the production of sufficiently representative data sets for the majority of codes. The most common practice to mitigate this problem was traditionally the integration of Information Retrieval (IR) methods to complement predictions, but current trends advocate supervised methods specialised in few-shot learning, either by exploiting transfer learning or co-dependencies using extreme algorithms.

The different computer-assited ICD coding proposals are detailed below in terms of foundations rather than chronological order, although the two are related in some sense. The number of proposals is enormous as ICD coding has been a recurring theme in the history of AI, so we have collected information on the most representative ones.

2.2 ICD challenges

Automatic ICD coding is an unresolved task with multiple challenges to be overcome. As discussed in Section 1.2.2, the assignment of the appropriate ICD codes to EHRs requires extensive comprehension of written text involving high knowledge of clinical terminology and advanced textual interpretative skills to deal with high-level semantics. Whereas some of these skills are acquired as a result of practice, the task often requires the annotation of relatively infrequent or new codes, making essential a solid capacity for generalisation. This need lies in the inherent nature of the prevalence of diagnoses and procedures, which is dominated by an extreme distribution in which a relatively small group of codes is usually associated with the majority of records. Such a distribution does not preclude the frequent association of rare codes as ICD is a classification with a high number of labels and the coding is a muti-label categorisation with a one-to-many cardinality. As for the documents, there are different types of health records and a variety of sizes: ranging from short (e.g., death certificates) to very long (e.g., hospital discharge reports). It should be noted that long texts are more complex because there are more elements interacting with each other, i.e., more information to discard.

Hence, ICD coding is characterised by data scarcity for a large percentage of codes, a great diversity of semantic granularities involving different levels of abstraction, and extreme distributions resulting in considerable imbalance of classes. Consequently, the tendency for researchers has been to address at least one of these challenges with currently known techniques. Regarding industry, the availability of records is still very limited, in contrast to other sectors where access to data is greater than in academia. As a result, commercial proposals have focused on unsupervised implementations. Below we detail the proposals we consider most representative for each of the challenges of the ICD in the academic domain. In addition, we outline the implemented software with the greatest impact on the health sector, focusing on the proposals made in Spain (where most of the data sets used in this research are located).

2.2.1 State of the Art

As discussed in Section 1.2.2, ICD coding is governed by a set of rules defined in the standard, designed to classify diagnoses and procedures. These rules are described in two sections: the tabular list, which provides code descriptions with multiple rules such as inclusions, exclusions, and additional notes, and the alphabetical index, which includes more specific terms related to the codes. However, the national institutions responsible for adapting coding generally only facilitate digital access to the code descriptions of Tabular List, with some exceptions such as the Swiss government[4]. Therefore, the resources inherent to the task in most cases are code descriptions and annotated records.

Given the loss of information involved in the individual use of descriptions and the complexity of the inherent semantics, most researchers have opted for supervised systems that infer the task using examples derived from records already coded by professionals. Some approaches have been unsatisfactory, such as the boosting ICD-9 algorithm proposed by Goldstein, Arzumtsyan, and Uzuner (2007) and based on Bag-of-Words (BoW) features, which did not outperform another implemented rule-based method. In other cases, relatively satisfactory results have been achieved, such as with the feature selection techniques. For example, Lita et al. (2008) propose two ICD-9 One-vs-Rest (OvR) models for long records: SVM and Bayesian Ridge Regression methods based on the top BoW features selected using χ^2. There are also studies that focus on feature engineering, such as the one proposed by Erraguntla et al. (2012), which uses morphological and syntactic features, such as noun phrases, lexemes, and POS Tagging for training an ICD-9 KNN model on patient and medical statement. Generative learning methods are also frequently used to increase Recall. For instance, Dermouche et al. (2016) compare LDA, Decision Tree, Naive Bayes, and SVM models in the ICD-10 coding of discharge summaries, resulting in no significant differences between the SVM and LDA classifiers. The authors approach the problem as a multi-class task, only dealing with primary diagnoses. Other proposals such as the one suggested by Kavuluru, Rios, and Lu (2015) combine generative and discriminative methods for coding short narratives with ICD-9. In this particular case, Learning-to-Rank (LR) methods are used to sort the predictions of OvR SVMs and

[4] `https://www.bfs.admin.ch`

Naive Bayes models.

The above methods suffer from a shortage of examples, which is one of the major issues for ICD coding. Besides, such approaches do not deal with the semantic differences between codes and records, a problem that is accentuated as more information is processed, as greater specificity complicates the capture of more abstract concepts. Finally, the extreme imbalance in the prevalence of diagnoses and procedures is also not tackled. The following are the most representative SOTA methods for each of the challenges mentioned above.

Data scarcity

Differences in prevalence, in addition to the difficulties implicit in the domain to access the data, cause lack and scarcity of examples for most codes. Multiple efforts have been made to increase the coverage of the proposed systems, mostly based on the development of partially or totally unsupervised approaches and introduction of transfer learning techniques. In particular, methods combining supervised algorithms and IR techniques, similarity-based methods, and representation- and relational-based transfer learning methods have been explored to infer non-represented codes in training data sets, while instance- and parameter-based transfer methods and data augmentation techniques have been applied to deal with poorly represented codes. Similarity-based approaches and representation- and relational-based transfer learning methods are detailed below as they are more closely related to the semantic processing.

As for unsupervised methods, a traditional proposal is the use of lexical matchings between manual annotations of the codes in other records and the texts from the records in the test set (Park et al., 2019; Pérez et al., 2018). Nevertheless, most authors have explored complementing ML models with unsupervised methods based on knowledge external to the task, such as entities from knowledge bases or statistical information from other corpora. Proposals such as the one by Pakhomov, Buntrock, and Chute (2006) stand out, who use a Naive Bayes model trained on the most frequent ICD-9 codes and a lookup table method based on manual annotations for recovering infrequent ones. Similarly, Patrick, Zhang, and Wang (2007) apply feature engineering in the training of SVMs on clinical notes and predict the non-suggested codes with a system based on official ICD-9 description matches. Sequential combinations have also been explored, such as the proposal by Crammer et al. (2007) for radiology reports, which explores a cascade combination by applying linear supervised classifers taking as features the predictions of the method based on matches with ICD-9 descriptions and other defined expressions. In this line, Pereira et al. (2013) implement a NLP process that deals with acronyms, stems, and grammar rules to identify clinical entities, which are used as main features for assigning epileptic ICD-9 codes preliminarily by means of rules. These authors subsequently train a KNN model on the candidate codes to provide the final codes. In turn, Zweigenbaum and Lavergne (2016) train a SVM for filtering the predicted codes by a match-based method at CLEF eHealth 2016. The authors achieve an F-Score improvement of more than 10% over the best results published so far. In contrast, Jatunarapit, Piromsopa, and Charoeanlap (2016) explore a VSM based on Latent Semantic Analysis (LSA) for estimating word similarities in Thai and English ICD-10 diagnoses, which are used for training a Bayesian model.

Another current trend is to introduce general knowledge directly on the supervised method by means of transfer learning in order to improve generalization during learning, requiring fewer examples. For example, some proposals have relied on instance-based transfer methods by mixing data of similar nature such as Subotin and Davis (2014), which use General Equivalence Mappings (GEMs) to train models for predicting ICD-10 codes on records coded with ICD-9. More specifically, the authors implement a cascade system by training a logistic regression classifier per character and position in the codes, so that the probabilities estimated by the models of a particular position are used as features for the models of the next position. Zhang et al. (2017) use ICD-9 descriptions and annotated radiology reports as queries for finding related PubMed articles, which are then used for training models on 45 codes. Alternatively, Jeblee et al. (2018) and Ševa, Sänger, and Leser (2018) use multilingual vectors to jointly train a sequence-to-sequence model on two ICD-10 corpora with different languages at CLEF eHealth 2018. Jeblee et al. (2018) explore the application of fastText embeddings trained on Common Crawl and Wikipedia to a bidirectional Long Short-Term Memory (LSTM) with attention mechanisms. The proposal achieved the third best results for the Hungarian and Italian subsets of data, and the fourth best results for the French subset. Ševa, Sänger, and Leser (2018) first train word vectors on Wikipedia and the training data sets by applying the word2vec algorithm, and then use them in a network based on Gated Recurrent Units (GRU). This proposal obtained the fourth position for the Italian set, but the last position for the French and Hungarian sets (it achieved less than half of the scores compared to the first team). We proposed the use of MT techniques to introduce instances of collections in another language as a function of the frequency ranges of the codes (Almagro et al., 2019). For this purpose, we use the data sets released in CLEF eHealth 2018 Multilingual Information Extraction Task.

Other authors have opted for parameter-based transfer methods, usually with BERT-style models, which limits the maximum size of documents. Accordingly, Ive et al. (2018) train a convolutional encoder-decoder network using character-level word embeddings on French records at CLEF eHealth 2018, which is fine-tuned on an Italian data set. The authors achieved about 25% lower F-Score than the first team. As for the BERT architecture, Sänger et al. (2019) fine-tune a multilingual model on German records, which outperforms SVMs and match-based methods at

CLEF eHealth 2019. The proposal achieved the best results for the task with a 0.8 F-Score. Amin et al. (2019) also analyze the performance of multiple models on coded German records at CLEF eHealth 2019. In addition to the multilingual model, the authors explore fine-tuning the English BERT model and BioBERT (a BERT-style model trained on PubMed) by applying Machine Translation (MT) methods for adapting German records to the English language. BioBERT surpasses other implementations, including OvR SVMs, Convolutional Neural Network (CNN), LSTM with attention mechanisms, and Hierarchical Attention Network (HAN) with English and German fastText embeddings. BioBERT reached the second position in the ranking with an F-Score value of 0.73. Instead, Silvestri et al. (2020) explores the cross-lingual use of a BERT model tuned on clinical notes in a different language to the test set. Such records comprise 93 words on average and are coded with a reduced set of 24 ICD-10 codes. Chalkidis et al. (2020) fine tune RoBERTa on MIMIC-III, which shows slight improvements in the clinical domain in comparison with BERT. MIMIC-III (Johnson et al., 2016) is composed of around 52,000 discharge summaries with 1,600 words on averaged and 8,761 associated ICD-9 codes. López-García et al. (2021) release multiple transformers pre-trained on general-purpose corpora and a corpus composed of 30,900 oncology records with around 2,000 words on average. Then, the authors fine tuned the models on the CodiEsp corpus from CLEF eHealth 2020. The results achieved are approximately 5% better than the best values reached by the shared task approaches. From a corpora perspective, Zhang, Liu, and Razavian (2020) compare the performances of BERT models pre-trained on general-purpose corpora, health articles from PubMed (BioBERT), and EHRs for the ICD-10 coding of medical notes. Specifically, only the first 1,000 chunks of the notes and a small group of 2,292 codes with more than a thousand occurrences in the corpus are used. In this experimentation, pre-training on clinical domain via records outperforms all others.

As an alternative, the application of data augmentation techniques such as noise injection and text generation has also been explored. For example, Biseda et al. (2020) use two-step cascade system, based on BERT representations for feeding a CNN to predict chapters and a LSTM that uses information from the chapter predictions to propose the specific categories on MIMIC-III. Biseda et al. claim that randomly shuffling the sentences to increase the training data produces a significant improvement in the results and generalisability of the model. Other more advanced data augmentation methods are applied by Ollagnier and Williams (2020), exploring a combination of CNNs and LSTMs fed with representations generated by BERT. Ollagnier and Williams compare the performance of this architecture on the training data set from CLEF eHealth 2020, which is augmented by applying synthetic record generation and synonym substitution. The first one is based on a Language Model pre-trained on the same data; the second one uses WordNet as the reference ontology and obtains the highest MAP performance. In this line, Garcıa-Santa and Cetina (2020) duplicate the number of training examples synthetically by building a language model from MIMIC-III, PubMed and the training data. The authors combine a knowledge graph based on CIE-10-ES and BERT implemented as a Named Entity Recognition (NER) method to predict diagnoses and procedures in the CodiEsp task from CLEF 2020. The proposal ranked third in the first two subtasks according to MAP values (CodiEsp-Diag. and CodiEsp-Proc.) and first in the third subtask according to F1 values (CodiEsp-Expl.).

Lexical and semantic diversity

Clinical language is riddled with synonyms, acronyms, and other interchangeable expressions, which implies an enormous lexical diversity that complicates example-based learning. Moreover, the length of documents is often unfavourable for learning, as they tend to introduce irrelevant information that diminishes the impact of diagnostic information. In addition, the ICD is composed of both specific codes such as A23.0 ("*Brucellosis due to Brucella melitensis*") and general codes designed to group concepts such as A23.9 ("*Brucellosis, unspecified*"), which comprises Malta, Mediterranean, and undulant fevers. These differences in abstraction imply heterogeneity in the learning of the codes, so that each one has a different associated complexity. Several approaches such as the enrichment of records with synonyms, introduction of partial match-based IR methods in supervised proposals, representation- and relational-based transfer learning methods, application of attention mechanisms for filtering information, and estimation of code similarities have been explored in order to tackle the lexical and semantic diversity of the task.

Most research approaches have focused on tackling lexical and semantic diversity by means of external knowledge bases, such as dictionaries and clinical ontologies. Hence, some authors have enriched representations with synonyms to apply Machine Learning techniques for modelling the complexity of coding. In this regard, Pereira et al. (2006) explore an NLP method for identifying Medical Subject Headings (MeSH) entities, which are subsequently mapped into the ICD-10 codes. Alternatively, Aronson et al. (2007) replace the Unified Medical Language System (UMLS) concepts detected in radiology reports by the corresponding identifiers, which are processed as words. Then, SVMs, KNNs and IR systems are assembled via stacking techniques for ICD-9 coding. Ruch et al. (2008b) train a KNN on French hospital records coded with ICD-10 while exploiting a Vector Space Model (VSM) method based on a French thesaurus for predicting unseen codes, and Boytcheva (2011) expand the sentences within Bulgarian medical texts by creating alternative text sequences with each of the identified synonyms for the words. OvR SVMs are trained on the union of these sequences for ICD-10 coding. In addition to medical ontologies, other resources such as general-purpose knowledge bases and Machine Translation (MT) tools are also used. For example, Rizzo et al. (2015) expand the ICD-9 descriptions with Wikipedia entries with the aim of applying VSM, Language Model (LM), and BM25 methods.

Such methods are evaluated on radiology reports and 45 ICD-9 codes. Van Mulligen et al. (2016) apply MT techniques via Google Translate and Microsoft Translator to identify French entities using UMLS (which provides English terminology) at CLEF eHealth 2016. The authors explore a system based on indexing concepts from other records, in which the terms from the exclusions defined in the standard are removed to increase accuracy. The proposal obtained the best results for the task with an F-Score of 0.848. Finally, Ho-Dac et al. (2017) exploit multiple representations for the codes at CLEF eHealth 2017, such as the concatenated associated records, ICD description, terminology from dictionaries, and the corresponding concepts from Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT). The authors achieved the second place in the ranking with this proposal, achieving an F-Score of 0.51 (30% lower than the best value).

Other authors have exploit the semantic information embedded in the word co-occurrences of external corpora, generally using sequence-to-sequence models. Following this idea, Miftahutdinov and Tutubalina (2017) explore vector representations pre-trained on social media texts via LSTMs, and Ševa, Sänger, and Leser (2018) propose to fed bi-LSTMs with attention mechanisms with fastText embeddings pre-trained on Common Crawl and Wikipedia. Li et al. (2018) extract sentence-level CNN features using learnt in-domain word embeddings, which are combined with also learnt document embeddings, and Blanco, Pérez, and Casillas (2020) explore the contextual representations Embeddings from Language Models (ELMo) with GRUs for analysing the differences in coding performance between medical specialties and ICD granularities. At the same time, Blanco et al. (2020) proposes a comparative study between a variety of vectorial representations: fastText vectors pre-trained on Spanish Billion Word Corpus (SBWC), the concatenation of fastText, word2vec, and GloVe pre-trained on the same corpus, and ELMo representations. While the authors obtain better Precision values with the combination of vector spaces, the Recall improvements achieved by ELMo representations yield higher F-Scores. Such experimentations has been carried out on a non-public collection of hospital records, dealing with only a limited set of fewer than 20 codes. The combination of contextual information from corpora and structured knowledge from ontologies has also been explored. Hence, Patel et al. (2017) modify the algorithm word2vec to fine tune word embeddings on the ICD-10 by adding hierarchical information. The system is designed by using logistic regression and vectors pre-trained on PMC, PubMed, and Wikipedia for code predictions on a private medical claims review data set. Furthermore, Alawad et al. (2018) apply retrofitting techniques for enriching pre-trained word embeddings with the SNOMED CT, ICD-10, and National Cancer Institute (NCI) thesaurus structure. Such retrofitted vectors are subsequently feed CNN models to predict a reduced subset of 12 codes for pathology reports using International Classification of Diseases for Oncology, Third Edition (ICD-O-3).

Relational-based transfer learning methods have also been explored to support zero and few-shot code suggestions. In this sense, Lu et al. (2020) and Rios and Kavuluru (2018) suggest very similar proposals applied to MIMIC-II and MIMIC-III. Both authors use Graph Convolutional Neural Networks (GCNNs) fed with the sum of the word embeddings pretrained on PubMed and corresponding to the ICD-9 descriptions to produce the label representation vectors, introducing hierarchical information in the label space. Lu et al. (2020) also introduce label co-ocurrence information in GCNNs. In addition, Rios and Kavuluru (2018) employ CNN layers and label-wise attention mechanisms to process records. Such an attention layer exploit the label vectors resulting from the sum of word embeddings instead of learning label specific parameters in order to learn relationships between semantic information rather than patterns for specific labels. Finally, the output layer compare matches between the latent features extracted from the records and labels. Chalkidis et al. (2020) also follow this line and compare multiple modifications on MIMIC-III. For example, the label representations based on the sum of vectors are replaced by node2vec representations (Grover and Leskovec, 2016). Alternatively, Song et al. (2020) propose a system based on CNNs that also applies a label-wise attention layer but does learn latent features per label. To this end, the use of Generative Adversarial Networks (GANs) to generate synthetic latent features for learning codes without examples are explored on MIMIC-III. In addition, GCNNs is replaced by Graph Recurrent Neural Networks (GRNNs).

Multiple authors have studied the positive effect of attention mechanisms, for promoting the sentences and expressions relevant to the final diagnoses and identifying features closer to the codes. In both cases, the aim is to deal with the record length required by real applications. Thus, Li et al. (2018) explore the extraction of sentence-level CNN features using learnt in-domain word embeddings, which are combined with also learnt document embeddings, while Li and Yu (2020) propose a CNN architecture based on multiple filter sizes, residual blocks, and label-wise attention layers. Both evaluate on MIMIC-II and MIMIC-III, with the first ones not truncating summaries and the second ones limiting the number of tokens to 2,500. In addition, Li and Yu (2020) evaluates performance on the 50 top codes. In contrast, Ji, Pan, and Marttinen (2020) explore a CNN architecture with gating mechanisms for filtering information and embedding injections in intermediate layers for avoiding forget semantic information. Besides, the model is fed with label representations by summing the word vectors of the ICD descriptions to link the notes and codes more easily. The combination of such techniques is intended to handle long texts. In order to present the results of the proposal, the authors have used the same assessment as Li and Yu (2020). Besides, Baumel et al. (2018) extend a HAN architecture by replacing LSTM units by GRU layers. The authors propose a document encoder comprising GRU and sentence attention layers, followed by a classifier consisting of label attention and fully connected layers, which are evaluated on MIMIC-III. It should be noted that the authors do not truncate the summaries but preserve the entire documents and evaluate over all codes, without reducing to a subset.

Similarly, Dong et al. (2021) also modify the HAN architecture by proposing a document encoder based on label-wise attention mechanisms for words and sentences and a document-to-label space matcher based on bi-GRU. Label representations are injected in the output layer for learning connections between the semantics of records and codes. During the experimentation, summaries are truncated to 2,500 tokens for HAN and less than 600 tokens for the rest. The authors evaluate the proposals on the whole set of ICD-9 codes and on a subset of the 50 most frequent ones. Ji, Hölttä, and Marttinen (2021) also explore a similar architecture to HAN, but proposes to split clinical notes from MIMIC-III into chunks to extract BERT representations for each fragment. These representations are truncated as in the previous proposal to feed a transformer with label-wise attention layers. The authors follow the same evaluation as in the previous proposal. Finally, Sen et al. (2021) explore a two-step system based on a multitasking approach, with two components for sentence tagging and ICD-10 coding, consisting of LSTMs with attention mechnisms and word embeddings pre-trained on PubMed. The first component classify the sentences within records as relevant or irrelevant, while the second component assigns the ICD codes to the sentences that pass the filter, yielding the final code set. The experimentation is carried out on a corpus of pathology reports and a reduced set of 410 ICD-10 codes.

Similarity techniques rather than discriminative learning models have been explored to find related codes with no training data, bridging the gap between meanings. In the line of lexical similarities, Lima, Laender, and Ribeiro-Neto (1998) propose an IR method exploiting a word graph that captures the ICD-9 hierarchy. The final score per code is calculated as a function of the corresponding traversed path for a text fragment. Author compare such a proposal with a traditional SVM method on a set of discharge summaries and a subset of almost 2,500 ICD-9 codes. Likewise, Medori and Fairon (2010) produce custom dictionaries based on UMLS and manual annotations in order to identify relevant clinical entities in the plain text. Such entities are then used as features in a Naive Bayes model and similar-based method, which compare syntactic structures via graphs. The authors use around 20,000 discharge summaries and a subset of 4,000 ICD-9 codes. Chiaravalloti et al. (2014) use Sørensen-Dice distance to find the most similar ICD-9 descriptions. A lexical normalization based on external sources is explored in order to maximize lexical overlapping, including the replacement of synonyms, acronyms, and abbreviations.

As for the semantic information, Chen, Barrera, and Rhodes (2010) propose an IR ICD-9 coding system based on syntatic parsers for radiology records. To this end, the authors match syntactic tree between sentences by using node aligment. Then, the similarity between radiology records is calculated as the sum of sentence similarities.

Henriksson, Hassel, and Kvist (2011) apply Random Indexing (RI) for representing the semantics of each patient record word in dense vectors, so that the ICD-10 words that are semantically closest to each word in the document are retrieved. In turn, Moen et al. (2015) use the cosine similarity between vectors based on TF-IDF values and weighted sum of embeddings for coding clinical notes with ICD-10. Different proposals that use semantic information embedded in the structure of ontologies (Chen, Lu, and Li, 2017; Ning, Yu, and Zhang, 2016) have been explored for coding short sentences using ICD-10. Ning, Yu, and Zhang (2016) propose to estimate the similarity between diagnoses in the form of text fragments and the ICD categories by means of individual word similarities, which are computed with HowNet (a chinese semantic knowledge base) and the Least Common Subsumer (LCS). Final codes are predicted in a cascade way, i.e., the chapters are first predicted, then the next sections, and so on. In the same way, Chen, Lu, and Li (2017) extend the Lin measure, which is based on LCS, to yield global similarities by weigthing the concepts from HowNet.

Imbalance

The population from any region tends to suffer from the same pathologies, so that the diversity of codes is conditioned. In fact, coding distributions usually follow a power law, with general codes associated with the majority of patients and more particular codes characterising clinical cases. Such differences in the code histogram lead to large decrements in performance when generalising knowledge through learning as ML algorithms tend to promote the most repetitive patterns, in this case the majority classes, and consider those less frequent, or minority classes, as noise. For this purpose, approaches based on the use of hierarchical information to improve the representation of less frequent codes, data augmentation methods to balance the number of examples among codes, and the use of XMTC algorithms focusing on the handling of co-dependencies and other balancing techniques have been explored. Proposals based on data augmentation methods have already been detailed in Section 2.2.1.

The variety of approaches that exploit the ICD hierarchy are designed to improve the generalisation of under-represented codes by exploiting the features of similar codes during learning. Examples include the proposal of Zhang (2008), which explore sequential SVMs imitating the ICD-9 structure, so that each classifier is only trained on a code branch. During inference, a classifier is only applied if the prediction at the top node exceeds the confidence threshold. Only 45 ICD-9 codes are used for a set of radiology reports. Another example is proposed by Perotte et al. (2011), training an LDA model and exploting the hierarchical structure so that the distribution of the superior categories is created with the documents associated to the final codes. Another way is proposed by Arifoğlu et al. (2014), who explore the indexation of the alphabetical index and the construction of queries from records, using NLP processes to expand the vocabulary with synonyms and remove negated entities. The codes are then sorted by lexical overlapping and subjected to voting in order to generate a ranking. This process is applied in a hierarchical way on the ICD branches: first the chapters are selected, then the sections, and so on to the final codes. The evaluation is performed on 6,000 discharge summaries annotated with a total of 7,298 ICD-9 codes.

As for more recent approaches, Manginas, Chalkidis, and Malakasiotis (2020) combine BERT-style models with the exploitation of hierarchical information to reduce imbalance. The idea is to associate the BERT layers with the ICD-9 levels. To do this, the authors assume that each record is also associated with the parents of the annotated codes, in a similar way to Perotte et al. Fine tuning is performed by unfreezing the last 6 layers in a gradual way, so that only the codes of the corresponding ICD layer are considered, starting with the chapters (more abstract) and deepening the specificity. For example, the second, third, and fourth unfrozen layers are trained on the 3-, 4-, and 5-digit codes respectively. The evaluation is performed on MIMIC-III, truncating the documents to 512 tokens. Other interesting proposal is released by Velichkov et al. (2020) applying data augmentation techniques and parameter-based transfer learning via a BERT-style model for sparsity. Noise injection is used by swapping letters for increasing the number of examples and clustering is applied at various hierarchical levels for replacing the ICD-10 codes with few instances by higher-level categories. In this way, the number of records is balanced for each annotated code. The authors use the corpus proposed by Boytcheva (2011). In another line, Sun et al. (2021) propose a multi-tasking scheme for learning ICD and Clinical Classifications Software (CCS) simultaneously. CCS encodes dependency information among ICD codes, which appears to be reflected in the results. The evaluation is conducted on summaries from MIMIC-III, which are truncated to 512 tokens, and the 50 top codes.

Proposals based on XMTC algorithms that exploit co-dependencies between labels aim to improve the inference of minority codes while learning the relationships among codes and reducing the computational complexity of the process. In this regard, Chalkidis et al. (2020) compare bi-GRU and transformer models with XMTC algorithms. Parabel, Bonsai, and AttentionXML algorithms are explored for ICD-9 coding. In turn, bi-GRUs with label-wise attention mechanisms, using ELMo representations, and BERT and RoBERTa models are also applied. In experimentation, transfer learning suits for general-purpose environments, but lacks clinical domain representations, so AttentionXML fits better in the clinical domain. The other published proposal extend the BERT model by introducing mechanisms from AttentionXML (Zhang, Liu, and Razavian, 2020). Label representations based on the sum of BERT vectors are used for initialising the label-wise attention parameters in order to accelerate generalisation during learning. The introduction of the AttentionXML layer introduces improvements in each proposed setup.

2.2.2 Commercial software

The implementation of institutional and commercial software is far from the state-of-the-art. For example, a widely used software is IRIS ICD coding tool[5], supported by several countries such as France, Germany, Hungary, Italy, Netherlands, USA, England and Sweden. It is a dictionary-based approach, which also apply lexical standardisation with language-dependent NLP pipelines.

In Spain, there are several CIE-10-ES coding assistance tools that provide easy navigation in the nomenclature and suggest codes in a semi-automatic way. To the best of our knowledge, all released software relies on unsupervised similarity estimation techniques, enriched with external knowledge bases, and supervised algorithms that do not deal with data sparsity or imbalance. According to publicly available information, no released tool handles long texts automatically, but works with text fragments.

Among the available frameworks, the Spanish Ministry of Health[6] has released the eCIEMAPS tool[7], which allows easy navigation through the CIE-10-ES tables, on the corresponding statistical portal. In turn, the Andalusian Health Service[8] has developed a tool with a similar functionality called MAC, which builds a thesaurus from the nomenclature. This tool supports more lexical alternatives, such as acronyms, and can narrow the search by structured fields, such as anatomical location. The Catalan Health Service[9] also offers a similar online tool.

As far as commercially available software is concerned, the company 3M[10] has developed a search engine that uses natural language to propose the most suitable code. In the absence of information, the system explicitly asks the coder for the missing data. In a similar way, the Basque Health Service[11] and Ibermática[12] have implemented Kodifika, a search engine based on dictionaries and annotations combined with semantic analysis. The CTMAP tool[13], developed by bitac[14], builds a knowledge base from thousands of medical records in such a way that semantic relationships are searched for the expressions marked by the coders in order to offer the most appropriate CIE-10-ES codes.

TeamCoder[15] is another software to facilitate the navigation of CIE-10-ES tables,

[5] http://www.iris-institute.org

[6] https://www.mscbs.gob.es

[7] https://eciemaps.mscbs.gob.es

[8] https://www.sspa.juntadeandalucia.es/servicioandaluzdesalud

[9] https://catsalut.gencat.cat

[10] https://www.3m.com.es

[11] https://www.osakidetza.euskadi.eus

[12] https://ibermatica.com

[13] https://www.bitac.com/plataforma-ctmap

[14] https://www.bitac.com

[15] https://www.sigesa.com/case-studies/enara-teamcoder

which is implemented by Sigesa[16] and Alfatec Sistemas[17]. In addition to the extended code search, it offers a predictive and dynamic code system based on statistical data. Alternatively, CliniCoder[18] have collected a set of online tools designed by Indizen[19] to promote semantic interoperability between different clinical data sources. One of its functionalities is to use of NLP and ML techniques on predefined sections to propose CIE-10-ES codes automatically. Finally, the company ASHO[20] has developed two tools, ASHOINDEX2 and AshoCoode, based on Machine Learning techniques and semantic search engines. ASHOINDEX2 is a coding assistant software that works by suggesting codes to the pieces of text selected by the annotators, while AshoCoode uses more advanced natural language processing techniques to provide the most relevant set of codes to the incoming clinical descriptions.

To the best of our knowledge, there is no published and supported evidence on the performance of any of these approaches[21] given the proprietary status of the software.

2.3 Discussion and concluding remarks

Section 2.1 summarises the evolution of computer-assisted ICD coding in order to provide an overview of the research thrusts of the task. Meanwhile, Section 2.2 presents the SOTA organised according to the different approaches proposed, which in turn have been grouped around one of the unresolved challenges identified in Chapter 1. As for the data used by other authors, there are proposals for multiple revisions of the ICD, different types of EHRs, and several simplifications, such as the reduction of the length of the records and number of codes. Only few approaches have tried to address the problem with the whole complexity (ICD-10, long records, and no simplifications) and none of them have tried to deal with the challenges identified in Chapter 1 simultaneously (data scarcity, extreme distributions, and generalisation problems). We have used 3 data collections with different complexities to address separate experiments: two academically designed corpora with short- and medium-length records, and another corpus with long records directly collected from hospitals; however, the aim of the thesis is to address coding in the full complexity, so we have focused on the collection of hospital-collected records.

The following is our view of the effectiveness of the approaches and the gaps detected in the SOTA.

[16] https://www.sigesa.com

[17] https://alfatecsistemas.es

[18] https://clinicoder.indizenlabs.es

[19] https://indizen.com

[20] https://asho.net

[21] With the exception of IRIS, whose performance has been compared in multiple studies, such as the one proposed by Harteloh (2020)

Overview The direct use of ICD descriptions to code records requires a great understanding of the domain as a large number of codes are designed to conglomerate groups of concepts and the corresponding descriptions imply a higher level of abstraction than the expressions written by clinicians. In addition, the inclusion and exclusion rules provide essential information that is not included in the descriptions, so that missing such data leads to information loss and coding errors. Coders must filter through hundreds of possible codes when analysing a record (Arifoğlu et al., 2014), so professionals who use computer-assisted ICD systems but do not follow the steps of coders report 32% more errors according to the study published by Yamada et al. (2010).

Supervised proposals use already coded EHRs to learn how to assign the codes, so in theory the ICD rules are implicitly inferred. Nevertheless, the reliance on examples entails other shortcomings characteristic of the ICD. The standard is too exhaustive for the data generally collected by health centres, so that many codes suffer from data scarcity, being poorly represented. Besides, differences in prevalence also lead to unbalanced data sets, in which a relatively small subset of codes dominates. Finally, ICD codes are designed with different semantic granularities, so differences in specificity require high-level clinical language knowledge. This is especially necessary in long records as more reasoning and abstraction capacity is needed. All these issues are unsolved challenges in the NLP field.

In the face of restrictions on access to clinical data, most comercial softwares are NLP-based computer-assisted systems, built on a combination of rules, statistical analyses and dictionaries (Campbell and Giadresco, 2020). Such systems often suffer from a lack of robustness as misinterpreting a term can have a high impact on the result, while the domain is subject to frequent typos, acronyms and other morphosyntactic structures that introduce variability (Weinberg et al., 2015). In addition, lexical rather than semantic processing is susceptible to failures in the interpretation of meanings, surrounded by conditionals such as negations (Perera et al., 2013). Nitsuwat and Paoin (2012) also highlight the errors involved in only applying lexical matching and point out the need to check the consistency of the proposed codes with the record.

In contrast, the trend in the academic community is towards the use of Machine Learning methods with which to infer the correct association of codes. Research proposals are conducted in controlled environments, reducing the number of under-represented codes and the size of the EHRs to focus on overcoming some of the challenges individually. In fact, many of the proposals do not describe the corpora used, which are not publicly available, no specifying the number of annotated codes and length of the records, and thus impeding a possible comparison between performances for similar conditions. However, conventional ML methods are not designed to deal with large-scale datasets (Popa et al., 2007) and long documents, as the relevant information for each code is more scattered across the records as the size increases

(Rios and Kavuluru, 2018). For example, SOTA approaches such as BERT tend to reach optimal results when processing documents of hundreds of words, but CNN architectures generally perform better in the categorisation of longer texts.

Research gaps The task of coding hospital records using ICD-10 involves data scarcity, label imbalance, scalability, and generalisability issues. As mentioned above, most proposals do not tackle such a task without some simplification, so there are many works that only focus on the most frequent codes or rely on shortening texts to deal with scalability and generalisation. Nevertheless, any real application requires an appropriate solution to cope with all the complexity involved. For this reason, we explore a variety of techniques to improve the ICD-10 coding of hospital records, addressing the main Research Objective of this thesis as outlined in Section 1.4.

As there is no proposal offering a complete and definitive technique, we have explored unsupervised and supervised approaches in more detail. We assume that unsupervised approaches yield worse accuracy for prediction in the absence of learning, but they provide better coverage and can complement supervised approaches in those underrepresented codes. We have organised the core of this research according to the recognised challenges, in a similar way to the structure of the described SOTA. Chapter 4 addresses unsupervised proposals which do not need examples and can predict codes misrepresented in data sets, Chapter 5 comprises proposals for promoting under-represented codes during learning, and Chapter 6 focuses on transfer learning methods to improve generalisation. In addition, Chapter 7 provides an overview of possible ensembles that deal with all the problems simultaneously.

Although numerous systems combining IR and ML methods have been proposed, none offer further details on the individual impact on the final performance. Therefore, we have proposed multiple experiments to support the Research Question 2. Specifically, we have studied in more depth the differences in performance of similarity-based methods on various differing types of EHRs sets: from academic environments, consisting of short documents and a smaller number of codes, to real data sets with long records and large code sets. Chapter 4 shows several lexical and semantic matching proposals that exploit inclusions and exclusions and different degrees of specificity. The performance of these proposals is also analysed in comparison with conventional supervised methods.

One of the most remarkable gaps is the absence of proposals dealing with imbalance, and extreme distributions in particular as it is one of the main features of ICD. This line is closely linked to Research Question 3. For that reason, we have explored multiple XMTC methods and settings in Chapter 5 to identify which foundations are best suited to the coding task. Different data augmentation methods have also been explored as an alternative to improve the representativeness of minority codes.

As for the Research Question 4, Chapter 6 includes different proposals based on the application of transfer learning methods to improve the generalisation of codes. In this line, the use of MT methods in crosslingual training has been explored to increase the predictive capacity. Different vector spaces and Language Models have also been generated from the data sets used in this thesis to provide quality representations adapted to the domain of Spanish EHRs. Furthermore, a parameter transfer method using hierarchical information to improve the inference of few-shot codes has been explored.

Finally, we return to the main topic about the need for a complete proposal that addresses all of the icd issues in Chapter 7, putting all of these explorations together, with comparisons and combinations of some of the most effective methods. This leads to answers to Research Questions 1 and 5.

CHAPTER

3

EXPERIMENTAL FRAMEWORK

Content

This chapter presents the experimental framework that will facilitate comparative studies between the different ICD-10 coding proposals, i.e., all the elements in interaction with the proposals and involved in the experimentation, i.e., data and evaluation. For this purpose, the collections to be used are examined, identifying trends and making statistical comparisons between data. Main features such as lexical diversity, coding criteria, and ICD coverage will be described to explain the results achieved in the different experiments. Besides, all basic and common text transformations for dealing with clinical records will be presented. Finally, evaluation criteria are proposed to compare the performance of different proposals according to specific attributes.

The following objectives are planned:

- Describe the collections used in the following chapters.
- Compare collections by assigning common attributes.
- Establish all pre-processing methods to be applied.
- Describe the evaluation metrics used to assess the different ICD approaches proposed in this thesis.
- Design a baseline to achieve reference results.

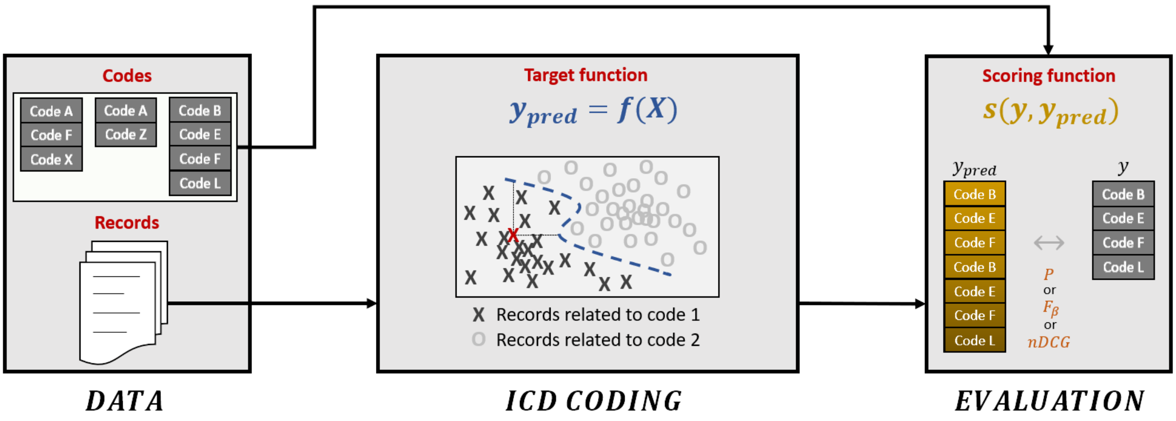
3.1 Introduction

As discussed in the introduction (see Section 1.4), a common experimental setup needs to be established to ensure the compatibility of the results before exploring any proposal.

The idea of coding a fragment or entire record is to provide a set of relevant codes, thus the ICD coding proposals are generally designed to build functions that maximizes the correlation between clinical text representations and relevant code sets. Such functions are used to score the correspondence of the codes to a new instance so that the relevant codes can be determined by setting threshold values or selecting the top K ones. This process of predicting multiple classes given a data point is a multi-label classification.

Data points, labels, target functions and score functions are the main elements involves in any data classification experiment approached from a computer science perspective. In our case, data points are the records from the data collections, labels are the codes associated by ICD specialists, target functions are the ICD coding functions to be explored and score functions are the metrics we will use. Figure 3.1 shows an overview of the experimental flow in which such elements are involved:

Figure 3.1: *Experimental outline.*



- *ICD CONDING* is the main component in the flow, which discriminates documents according to their matching with the codes. The prediction function is based on specific coding criteria which may be derived either from general knowledge and axioms or from data.

- The *EVALUATION* element measures performance by applying some metric defined by the scoring function. It is necessary to execute the proposal on a controlled collection of data to assess behaviour.

- *DATA* completes the flow by providing reports for the coding algorithm, which will result in predictions, and codes for the evaluation function.

This chapter aims to describe all the elements involved in the experimentation with the exception of the coding proposals themselves, as the ICD-10 approaches will be described in the following chapters. In addition to data and evaluation, a data pre-processing section has been included with the intention of standardising the format of the data to be fed into the approaches. The content of each section is briefly introduced below.

Data Access to data is a requisite for fully exploring approaches. As shown in Figure 3.1, the conduct of ICD coding experiments requires coded examples in order to evaluate the performance. At the same time, supervised techniques also demand large amounts of coded data to learn task patterns. During the development of this thesis, three collections of medical records have been made available, two of which were created expressly for shared tasks, while the third was generated collecting coded records from hospitals. Further details of the three collections have been described in Section 3.2.

Data pre-processing Electronic Medical Records are complex documents, full of sensitive information and a huge lack of uniformity, with many variation of formats and syntax outside the norm. For this reason, before representing the documents, certain filtering and standardisation processes are necessary to be fed into a coding algorithm. Section 3.3 aims to detail the pre-processes that have been carried out to improve the representation of documents for proposals. Although the same sub-processes have been applied to all three corpora by customising the language-dependent components, the pipeline has been designed especially for Spanish as the Hospital Universitario Fundación Alcorcón (HUFA) corpus is more relevant to this research.

Evaluation In terms of comparison of results, reference data and scoring criteria are needed. We have assumed that the ultimate codes assigned by the ICD specialists are appropriate and, in fact, the only possible solution. These codes will therefore be the reference values, constituting the gold standard of each collection. Based on these reference values, multiple metrics that highlight a variety of features of the results have been proposed to assess the proposals in different aspects. Section 3.4 details the restrictions and computations applied to the comparison between the predicted code sets and the target sets suggested by the ICD specialists.

3.2 Corpora

Data collections are essential for both the evaluation of the explored proposals and the implementation of supervised approaches. We have used 3 different collections for the elaboration of this thesis that will be referenced in the individual chapters detailing the ICD coding proposals. Two of the collections corresponds to data sets published in shared tasks, while the other collection is constituted by authentic coded medical reports proceeding from hospitals.

The data sets elaborated for ICD-10 competitions have introduced further restrictions and simplifications. First, the Conference and Labs of the Evaluation Forum eHealth 2018 has provided the multilingual *Causes of Death* corpus (Névéol et al., 2018) to the participants, comprising 273,744 death notes in French, Hungarian, and Italian. Second, the CLEF eHealth 2020 has released 3,751 Spanish EMRs to participate in the CodiEsp (Miranda-Escalada et al., 2020).

As for the corpus with real data, it is the product of a collaboration with the Hospital Universitario Fundación Alcorcón (HUFA). As a result of the agreements with HUFA, we have access to about 40,000 Spanish coded hospital discharge reports collected between 2016 and 2018, and over 120,000 Spanish uncoded reports from previous years. Anonymisation is required to process such reports, so we have developed a customised anonymisation method, which will be detailed in Section 3.3.

The three collections are described in separate sections below. For this purpose, details have been organised into certain common sub-sections in order to provide uniformity. These subsections are:

- *General description and structure*, for presenting an overview of the corpus and the reports it contains.
- *Coding criteria*, for detailing the guidelines followed by the professionals.
- *Example*, for illustrating a record from the corpus.
- *Textual descriptive statistics*, for providing details about lexical features.
- *Coding descriptive statistics*, for giving an idea of trends in annotated codes.
- *Feature summary*, for outlining the main aspects of the corpus.

3.2.1 *Causes of Death* corpus

General description and structure

The *Causes of Death* corpus (Névéol et al., 2018) comprises 273,744 death certificates from different countries and coded with the ICD-10. Death certificates are records with a significant proportion of structured data; nevertheless, the most important section*, the sequence of diseases or events leading to death,* remains as free text. This corpus collects the digitisation of the content of such section together with some relevant data such as which code is the primary cause of death, the gender or age of the deceased, and the location of death. The template used for data collection is attached in appendix A.

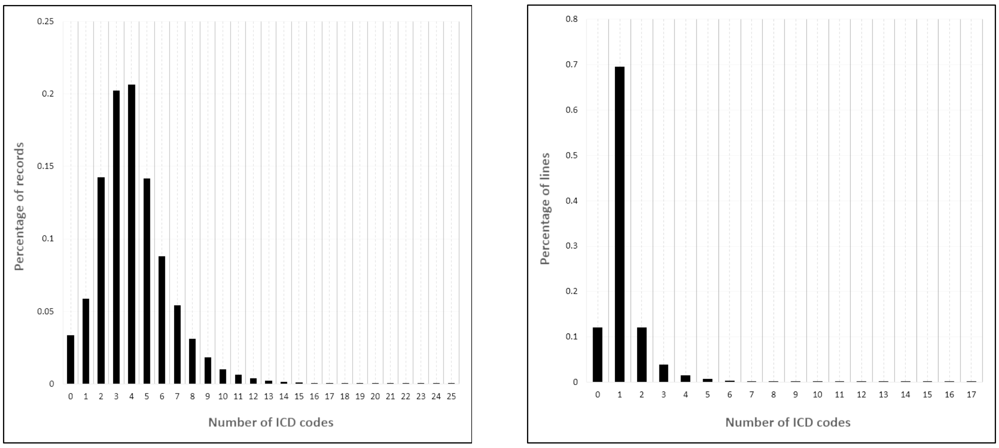
The content of the sequence of diseases consists of 1 to 6 short lines stating the causes of death. A total of 907,091 statements of causes of death are available, which implies 41 megabytes of textual data if structured data are excluded.

The corpus can be divided into three separate subsets with different languages: CépiDc-FR, KSH-HU and ISTAT-IT data sets. The French Institute for Health and Medical Research (CépiDc) collected electronic death certificates from physicians and hospitals in France over the period of 2006-2015 (Pavillon and Laurent, 2003), resulting in the CépiDc-FR data set. In turn, the KSH-HU data set was supplied by the Hungarian Central Statistical Office (KSH) by electronically transcribing a sample of the deaths reported in Hungary for the year 2016. Finally, the Italian National Institute of Statistics (ISTAT) produced a synthetic corpus based on real data from different years, including linguistics variants and spelling mistakes. In the ISTAT-IT data set, records were created using coherent lines from multiple certificates and coded by ICD specialists according to the 2016 version.

Coding criteria

Records have been coded with the international version of the ICD-10, which contains only 4-digit codes and excludes procedures. Hence, the overall number of diagnoses defined in the nomenclature is limited to about 14,000.

Figure 3.2: *Percentage of death certificates and lines per number of associated ICD codes.*



(a) *Percentage of records per ICD code number.*

(b) *Percentage of lines per ICD code number.*

Certificates consist of a pair of lines stating the diagnoses, so coding has been done at the sentence level. Since the lines are short, the association of codes has been conducted as an entity linking process based on lexical normalization, which is not at all trivial due to the presence of lexical derivations, acronyms, and abbreviations not included in the dictionaries. Official coding rules have not been used to include additional diagnoses or to avoid the concurrent association of incompatible entities.

Statements do not necessarily describe a single diagnosis, but may contain multiple elements. In turn, there are events that do not fit into the classification. Consequently, coding comprises one-to-many relationships between certificate statements and ICD-10 codes. Figure 3.2 shows the trend in the number of codes associated with records and, more specifically, lines. Although most lines in the corpus match a single ICD code, 20% have been paired with multiple codes.

Example

Three examples of sequences of diseases and events from the death certificates are shown in Figure 3.3, both translated into English. The untranslated examples are shown in Appendix B. The Italian certificate is associated with 5 ICD codes and contains 9 tokens spread over 3 lines, the French example groups 10 different ICD diagnoses and comprises 20 tokens within 5 lines, and the Hungarian certificate only describes two diagnoses producing one code each.

Figure 3.3: *Examples of Italian, French, and Hungarian death certificates translated into English. Tags shown in brackets are manually annotated ICD-10 codes, so they are not part of the text. The original, untranslated example can be found in Appendix B.*

Italian certificate	
lymphoblastic leukemia acute	[C91.0]
cardiac arrest	[I46.9]
cognitive decay, disfage, parkinson	[R41.8, R13, G20]

French certificate	
neoplastic cachexia	[C80.9]
atrial fibrillation with rapid ventricular response	[I48.9, I47.2]
cardio-circulatory and respiratory decompensation	[I51.6, J98.8]
pulmonary neoplasia	[C34.9]
sigmoid resection for neoplasia, COPD, hypothyroidia	[Y83.6, D48, J44.8, E03.9]

Hungarian certificate	
liver coma	[K72.9]
liver metastatic dose	[C78.7]

The *Causes of Death* corpus typically collects diagnoses with slight lexical variations with respect to the code descriptions, but this tendency is not a general rule. Sometimes the descriptions are longer and more abstract, so that coding demands semantic knowledge. This is the case of the code R41.8 (Other and unspecified symptoms and signs involving cognitive functions and awareness) associated with the last statement within the first example. R41.8 is designed to group all diagnoses that do not fit into the preceding categories, involving a higher level of abstraction.

In this way, the Italian example captures such differences in the complexity when coding diagnoses, as the classification of the first two diagnoses is almost immediate. The codes C91.0 (Acute lymphoblastic leukaemia) and I46.9 (Cardiac arrest, unspecified) provide concise information. The same applies to the last two codes, R13 (Dysphagia) and G20 (Parkinson's disease).

The French example deals with compound words such as *cardio-circulatory*, related to the code I51.6 (*Cardiovascular disease, unspecified*), and acronyms such as *COPD* (Chronic Obstructive Pulmonary Disease), which is linked to the diagnosis J44.8 (*Other specified chronic obstructive pulmonary disease*).

Textual descriptive statistics

There are almost 150,000 French records, 100,000 Hungarian records, and 20,000 Italian records. All death certificates are records of 1 to 6 lines, with an average of

	French	Hungarian	Italian	Total
Certificates	149,749	105,877	18,118	273,744
Lines	439,111	405,555	62,425	907,091
Differentiated lines	143,832	77,370	17,426	238,475
Line average per certificate	2.93±1.21	3.83±0.93	3.44±1.03	3.31±1.18
Tokens	1,517,222	896,135	175,692	2,589,049
Differentiated tokens	30,199	16,479	4,372	49,114
Token average per line	3.45±2.61	2.21±1.68	2.81±1.83	2.85±2.27

Table 3.1: *Statistical description of the certificates from the Causes of Death corpus.*

3.31 lines per record and a standard deviation of 1.18 lines. In turn, lines comprises a mean of 2.85±2.27 tokens, with a maximum of 71. This implies a tendency for brief documents with concise lines. Table 3.1[1] shows descriptive statistics of the textual content of the corpora.

Although the differences between the data sets are slight, French certificates are composed of fewer but longer lines, with an average of 2.93 lines per certificate and 3.45 tokens per line. In addition, French data set comprises fewer repeated sentences in terms of the proportion of unique sentences in relation to the total number of sentences. On the contrary, Hungarian certificates contain more and shorter lines, averaging 3.83 lines per certificate and 2.21 tokens per line. Whatever the structure, the Italian data set is lexically more diverse as the ratio of unique tokens per document is 0.24, higher than 0.2 and 0.15 of the French and Hungarian data sets respectively.

Coding descriptive statistics

CLEF organizers have divided each data set into training and test subsets, keeping approximately an 80% and 20% split respectively. We have respected this data partitioning during the experiments.

Certain diagnoses are much more present than others in populations, so diagnostic detection is usually not as diverse as the ICD itself. Table 3.2 reflects the overall amount of codes in the corpus. The three data sets together (FR, HU, and IT) cover around 5,000 different ICD-10 codes from a total of approximately 14,000, representing 35%. As the trend of the individual codes is not reflected in these overall values, the distribution of codes is shown in Figure 3.4.

The global overview (see Table 3.2) reveals that the corpus deals with a total of 5,182 different codes across 273,744 certificates, implying a total of 1,125,079 annotations. In turn, it should be noted that 4%, 10% and 11% of the test codes do

[1] There are discrepancies between the statistical values shown in Table 3.1 and those provided by the task organizers. This is because they only count those documents with at least one associated code.

not appear in the training set, which is a significant percentage of unseen codes to be considered for a possible proposal. As for the quantity of diagnoses distributed per certificate, the average number of codes per document is 4.32±2.11, as anticipated in Figure 3.2, and decreases to only 1.30 codes per line. As a result, the identification of a single entity is expected in 8 out of 10 statements.

Figure 3.4 shows the code distribution. A small percentage of ICD codes tend to appear in most records while the rest are rarely associated with patients, resulting in power-law distributions. In this case, 1, 6, and 5 ICD codes appear on more than 10% of training certificates in the French, Hungarian, and Italian data sets respectively, while over 50% of detected diagnoses, or unique ICD codes, are associated with less than 5 records. This disparity illustrates the imbalance in the codes.

The upper points on the far left represent the number of codes attached to a single document, e.g. there are 433 and 334 unique codes with a single instance in the Italian training and test data sets. Conversely, the lower points on the X-axis represent frequencies associated with a single code. For example, the black dot furthest from the y-axis in the Italian graph corresponds to the code I46.9 (*Cardiac arrest, unspecified*), being the most frequent code in the training set as it appears in 4,406 records. Overall, the Hungarian data sets seem visually more similar, even overlapping, while there are more differences in the numbers of codes with the same frequencies between the French splits.

Feature summary

The main features of the corpus are summarised below:

- **Multilingualism** facilitates a comparison of different contexts and linguistic features. The three non-English languages suffer from a shortage of clinical tools.
- The collection comprises **short** and concise medical records.
- Low verbosity leads to more word repetitions and **less lexical variety**. There is almost 1 new word for every 5 certificates.

	French		**Hungarian**		**Italian**		**Total**
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	
Certificates	125,375	24,374	84,702	21,175	14,501	3,617	273,744
Total ICD codes	509,103	48,948	392,020	98,264	60,955	15,789	1,125,079
Unique ICD codes	3,723	1,806	3,124	2,011	1,443	903	5,182
Unique unseen ICD codes	-	70	-	202	-	100	372
Code average per record	4.09±2.28	4.15±2.33	4.63±1.84	4.64±1.85	4.20±1.81	4.36±1.90	4.32±2.11
Code average per line	1.41±0.91	1.42±0.93	1.20±0.67	1.20±0.67	1.22±0.62	1.25±0.65	1.30±0.80

Table 3.2: *Statistical description of the codes from the Causes of Death corpus.*

- The coding criteria differ from the official specifications. Annotations are designed as an association of entities governed by **lexical standardisation**.

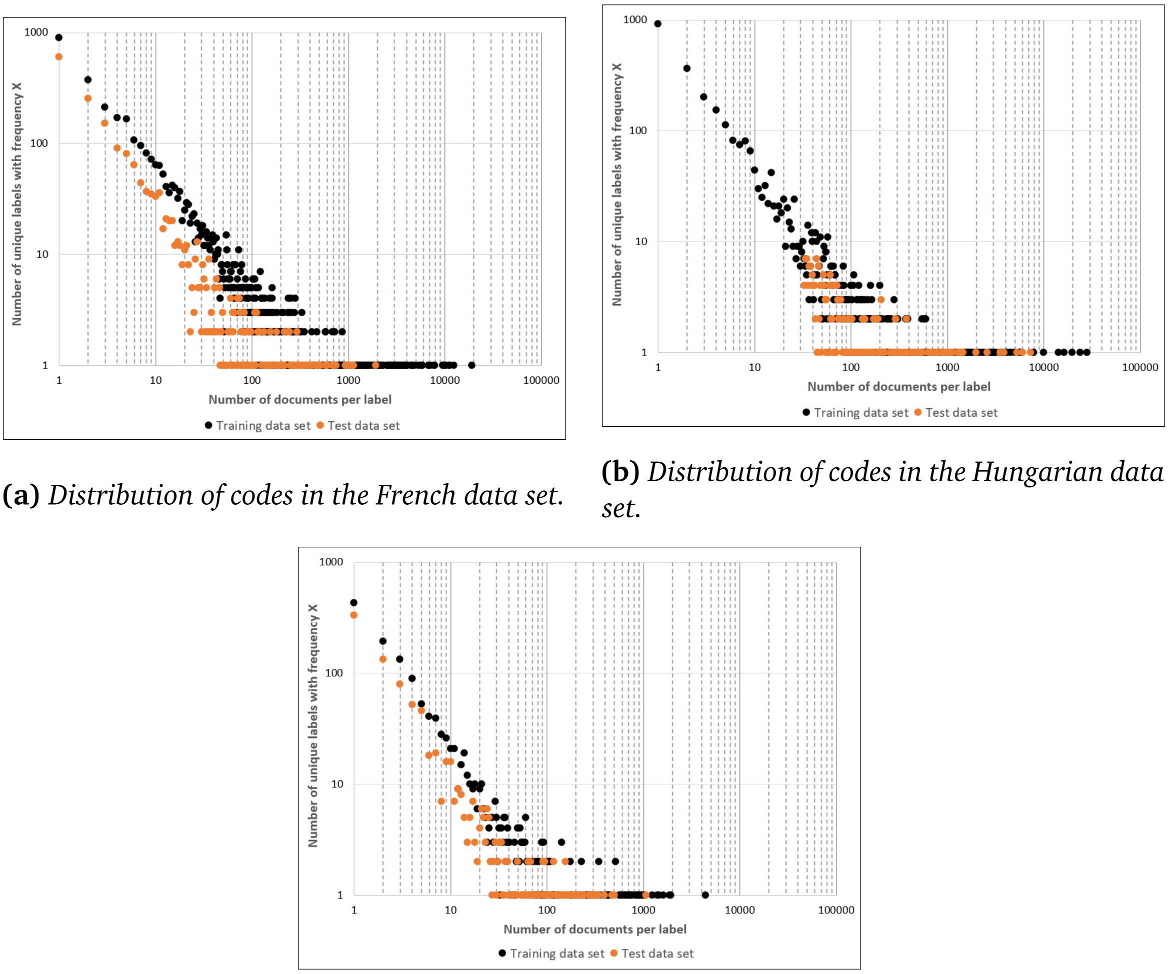
- The distribution of codes is unbalanced, but with a **coverage** of about **35%**.

3.2.2 *CodiEsp* corpus

General description and structure

A total of 3,751 Spanish clinical case reports are collected in the *CodiEsp* corpus (Miranda-Escalada et al., 2020). Of these, only 1,000 reports have been coded by ICD professionals, applying the CIE-10-ES standard. The remaining more than 2,000

Figure 3.4: *Distribution of the ICD-10 codes in the Causes of Death corpus. Each dot symbolizes the number of codes Y with the same frequency X.*



(a) *Distribution of codes in the French data set.*

(b) *Distribution of codes in the Hungarian data set.*

(c) *Distribution of codes in the Italian data set.*

reports are part of the background data set, which has been designed to build a silver standard according to the CLEF organisers.

Clinical case reports consist of a detailed explanation of the symptoms, medical signs, diagnoses, treatments, and monitoring of an individual patient. These reports usually describe the demographic and socio-cultural situation of the patient, with information collected through anamnesis and physical examinations, combined with those evidences obtained from complementary tests such as imaging and analysis.

The records have been collected from different medical services such as oncology, urology, cardiology, pneumology, and infectious diseases, although it is not specified which in each case. Such EMRs are completely unstructured documents, so they have been provided in plain text format. These are generally long documents consisting of several paragraphs and constitutes a total of 9 megabytes of textual data.

Coding criteria

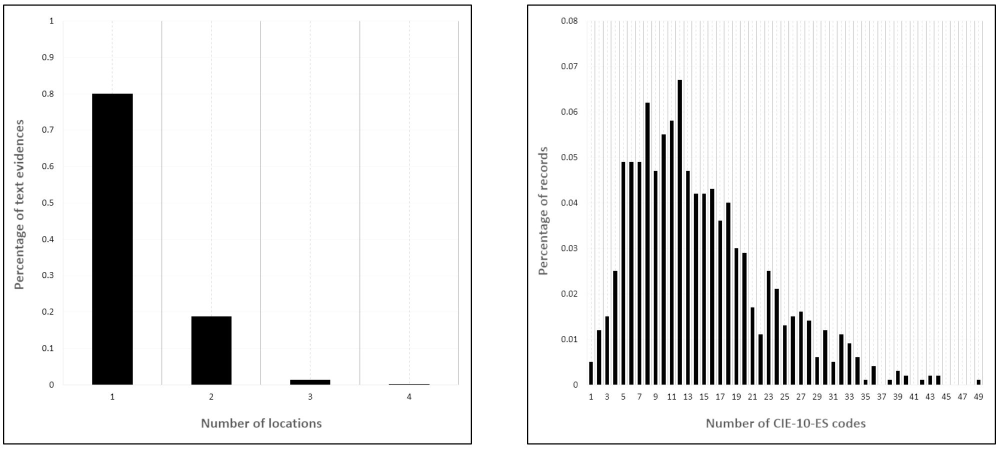
ICD is regularly updated to reflect new definitions and procedural changes. The 2018 version of the CIE-10-ES has been used to code the clinical case reports. CIE-10-ES is the Spanish modification of the ICD-10-CM and ICD-10-PCS (USA extension of ICD-10 and OPCS-4), which extend the specificity and length of codes to 7 characters. The modification of the global classification is a standard that collects a wide range of medical entities such as procedures, diseases, disorders, injuries, and other related health conditions. In particular, the 2018 version comprises 185,754 final and non-final CIE-10-ES codes, of which 98,584 are diagnoses and 87,170 procedures.

The annotation of this corpus has involved a manual coding process employing 3 clinical experts. Textual evidences have been linked to codes in order to support the assignments. After some iterations, a pairwise percentage agreement of 88.6%, 88.9%, and 80.5% have been reached for diagnoses, procedures, and textual evidences respectively. According to the annotation guide, any pathological process, symptom, sign or circumstance deriving in a disease has been coded as a diagnosis. At the same time, all the implementation of medical interventions aimed at the provision of health care has been recorded as procedures whenever at least one of a series of criteria is satisfied:

- It may entail non-topical anaesthesia.
- It is surgical in nature.
- It involves risks for the patient.
- Its implementation requires additional clinical training.

This manual annotation process has followed specific guidelines that differ from the official nomenclature. On the one hand, all entities linked to diagnoses and

Figure 3.5: *Distributions of textual evidences in terms of segments and records according to the number of codes.*



(a) *Percentage of text evidences distributed in X locations.*

(b) *Percentage of CodiEsp records per number of associated CIE-10-ES codes.*

procedures were annotated even though they are not related to the patient or do not provide the complete relevant information. For example, demographic and physical characteristics were not considered during coding. In turn, denied or medically suspected entities have been coded.

The deficiency of information implies the identification of precise and ambiguous diagnoses and procedures, either covered in the standard by final codes or in superior groups (non-final codes). Although all diagnostic groups have been used for coding, only procedures from 4 characters have been annotated in order to ensure a minimum ICD unit of meaning.

On the other hand, exhaustiveness has been prioritised. Multiple mentions have been considered by annotating each time the same identified entity in the record. Also, the possibility of overlaps between annotations has been envisaged; the same entity may correspond to one diagnosis and procedure simultaneously, or two entities may share a fragment of text. In addition, official coding rules such as exclusions between codes or inclusions of complementary codes have not been applied, which ensures that all codes have textual evidence.

Regarding the textual evidences, the expressions and their positions have been recorded together with the CIE-10-ES codes in the gold standard. The clinical-coding evidence text is not necessarily continuous, as the codes are the result of the synthesis of possibly dispersed clinical information. Figure 3.5a indicates the distribution of textual evidence according to the number of discontinuous fragments. As shown, 80% of entries are continuous textual evidences, while 20% of evidences are located in more than one text fragment.

Coding annotations are not at sentence level but at document level due to the multi-location of textual evidences. Even so, the knowledge of these locations facilitates the use of techniques more focused on the recognition of entities. Figure 3.5b shows the percentage of records per number of associated CIE-10-ES codes. As illustrated, the mentioned flexibility in CIE-10-ES annotation, in particular the criterion of coding any non-patient entity, leads to a very diverse range of codes linked to the documents. Thus, the average number of codes per document is 14.41 with a high standard deviation of 8.12.

Example

An example of a clinical case report translated into English is shown in Figure 3.6. The untranslated example is shown in Appendix C. Clinical-coding textual evidences are highlighted in bold in the text.

The illustrated record is composed of 13 sentences grouped in 4 paragraphs, so the length is slightly below average. Of the 6 codes linked to the document, 2 correspond to diagnoses and 4 to procedures. Besides, the example contains discontinuous and redundant evidences, such as the expression *cystoscopy* found in two different text locations, or the fragment *bladder resection*, which is composed of two distant words.

The assignment of CIE-10-ES codes is a process that relies more on the use of synonyms and related meanings rather than on lexical structure. The associations of the expression *bladder resection* with the code 0TTB (*Resection of Bladder*) and the entity *pelvic ultrasound* with BW4GZZZ (*Ultrasonography of Pelvic Region*) are almost direct matches; nevertheless, the other connections require deeper domain knowledge. For example, the identification of the tumour types is essential for pairing the fragment *bladder leiomyoma* and the code D30.3 (*Benign neoplasm of bladder*). In turn, the code R58 (*Haemorrhage, not elsewhere classified*) and the expression *bleeding* are connected by synonyms. Another more complex association is the relationship between the code 0TJB8ZZ (*Inspection of Bladder, Via Natural or Artificial Opening Endoscopic*) and the evidence *cytoscopy* as the official description is closer to the definition.

Textual descriptive statistics

Table 3.3 presents an overview of the lexical composition of the corpus. There are 3,751 clinical case reports with of 372 tokens on average. These records consist of multi-paragraph EMRs with an average of 18 sentences and a standard deviation of 14. Such sentences are verbose, comprising an average of 21 tokens. In turn, the widespread presence of typos, acronyms, abbreviations, and synonyms increases lexical diversity, achieving a ratio of almost 3 new tokens for every 4 sentences.

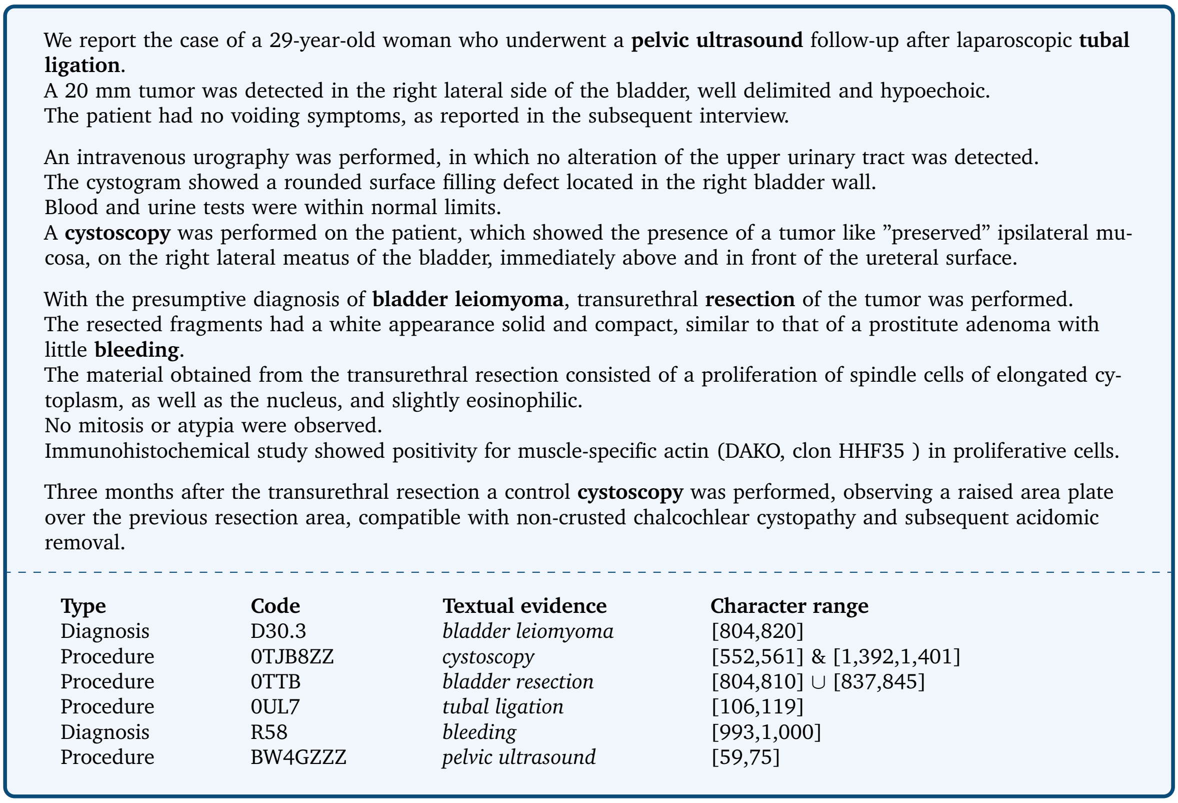
There is hardly any difference between the coded records and those constituting the background data set. Although the latter are almost 2 sentences, or 26 words, longer on average, the length of the sentences practically remains the same. In terms of diversity, the coded records do exhibit a higher ratio of new tokens per record, around 22, than the documents in the background data set, which reach a ratio of almost 16.

Coding descriptive statistics

First of all, the 1,000 coded records have been grouped into training, development, and test subsets in percentages of 50%, 25%, and 25% respectively. This partition is the division proposed by the organisers, so we have respected such data separation during the experimentation in order to facilitate the comparison of results with those reached by the rest of the participants in the shared task.

Table 3.4 summarises the global indicators referring to the CIE-10-ES codes of each subset of data. Overall, 78% of the 18,435 annotations correspond to diagnoses, while the remaining 22% are procedures. In turn, there are 7,209 examples for 1,767 diagnoses and 3,431 instances for 563 procedures in the training data set. This is

Figure 3.6: *Example of a report translated into English from the CodiEsp corpus. The footer contains the annotations along with their evidence and positions in the text. The original, untranslated example can be found in Appendix C.*



We report the case of a 29-year-old woman who underwent a **pelvic ultrasound** follow-up after laparoscopic **tubal ligation**.
A 20 mm tumor was detected in the right lateral side of the bladder, well delimited and hypoechoic.
The patient had no voiding symptoms, as reported in the subsequent interview.

An intravenous urography was performed, in which no alteration of the upper urinary tract was detected.
The cystogram showed a rounded surface filling defect located in the right bladder wall.
Blood and urine tests were within normal limits.
A **cystoscopy** was performed on the patient, which showed the presence of a tumor like ”preserved” ipsilateral mucosa, on the right lateral meatus of the bladder, immediately above and in front of the ureteral surface.

With the presumptive diagnosis of **bladder leiomyoma**, transurethral **resection** of the tumor was performed.
The resected fragments had a white appearance solid and compact, similar to that of a prostitute adenoma with little **bleeding**.
The material obtained from the transurethral resection consisted of a proliferation of spindle cells of elongated cytoplasm, as well as the nucleus, and slightly eosinophilic.
No mitosis or atypia were observed.
Immunohistochemical study showed positivity for muscle-specific actin (DAKO, clon HHF35) in proliferative cells.

Three months after the transurethral resection a control **cystoscopy** was performed, observing a raised area plate over the previous resection area, compatible with non-crusted chalcochlear cystopathy and subsequent acidomic removal.

Type	Code	Textual evidence	Character range
Diagnosis	D30.3	*bladder leiomyoma*	[804,820]
Procedure	0TJB8ZZ	*cystoscopy*	[552,561] & [1,392,1,401]
Procedure	0TTB	*bladder resection*	[804,810] ∪ [837,845]
Procedure	0UL7	*tubal ligation*	[106,119]
Diagnosis	R58	*bleeding*	[993,1,000]
Procedure	BW4GZZZ	*pelvic ultrasound*	[59,75]

	Background	Coded EMRs	Total
Records	2,751	1,000	3,751
Sentences	50,187	16,655	66,842
Unique sentences	49,268	16,472	65,497
Sentence average per record	18.24±15.75	16.65±8.84	17.82±14.26
Tokens	1,044,169	353,841	1,398,010
Unique tokens	43,165	22,399	48,702
Token average per sentence	20.80±13.11	21.48±13.19	20.91±13.13
Token average per record	379.56±320.61	353.84±166.81	372.70±287.98

Table 3.3: *Statistical description of the EMRs from CodiEsp corpus.*

	Train	Dev.	Test	Total
Records	500	250	250	1,000
Diagnosis codes	7,209	3,431	3,665	14,305
Procedure codes	1,972	1,046	1,112	4,130
Total CIE-10-ES codes	9,181	4,477	4,777	18,435
Unique diagnosis codes	1,767	1,158	1,143	2,557
Unique procedure codes	563	375	371	870
Total unique CIE-10-ES codes	2,330	1,533	1,514	3,427
Unique unseen diagnosis codes	-	427	439	790
Unique unseen procedure codes	-	164	178	307
Total unique unseen CIE-10-ES codes	-	591	617	617
CIE-10-ES code average per record	14.38±8.20	13.98±8.03	14.88±8.01	14.40±8.12

Table 3.4: *Statistical description of the CIE-10-ES codes from CodiEsp corpus.*

only 1.25% of the total number of codes defined in the standard because non-final codes are also candidates for standardisation in this collection, as mentioned in the coding criteria. Besides, development and test data sets contain about 600 unseen codes, which implies around 18 percent of the recorded codes. It involves a significant non-overlap between codes from different data sets.

Regarding the distribution, the trend of codes is again shaped by power laws. About 1,000 codes, 60% of the CIE-10-ES annotations, are coded only once in each subset of data, as shown in the upper point of Figure 3.7. In fact, only 112, 43, and 42 CIE-10-ES codes exceed 10 occurrences in the subsets of training, development, and test data respectively. All codes to the left of these 10 instances on the X-axis constitute the tail, which is often referenced in other research because of the inherent complexity for prediction.

Feature summary

The main properties of the corpus are:

- Clinical cases written in **Spanish**. In addition to the inherent complexity of the language given its syntactic and grammatical flexibility, the availability of tools in the clinical domain for the Spanish language is limited.

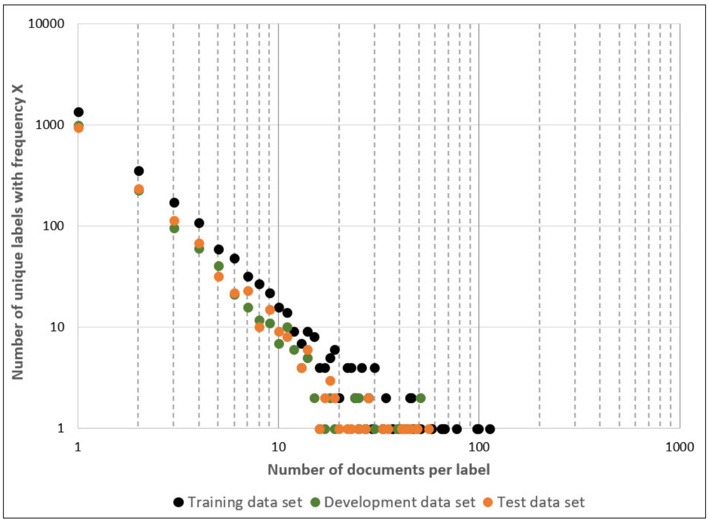
- The provided medical records are of **moderate length**, with more than a dozen sentences per document.

- Sentences tend to comprise 20 tokens, so they are **verbose** and **lexically rich**. In general, 13 new tokens appear for each document.

- The annotation of incomplete and non-patient-related diagnoses and procedures diverges from the official ICD coding criteria. Coding is supported by the detection and **identification of medical entities** present in the ICD codes.

- The frequency of codes follows a power-law distribution. As a result of the severe imbalance, only examples of 3,427 codes are collected, implying a **coverage** of almost **2%** of all possible diagnoses and procedures.

Figure 3.7: *Distribution of the CIE-10-ES codes in the CodiEsp corpus. Each dot symbolizes the number of codes Y with the same frequency X.*



3.2.3 *HUFA* corpus

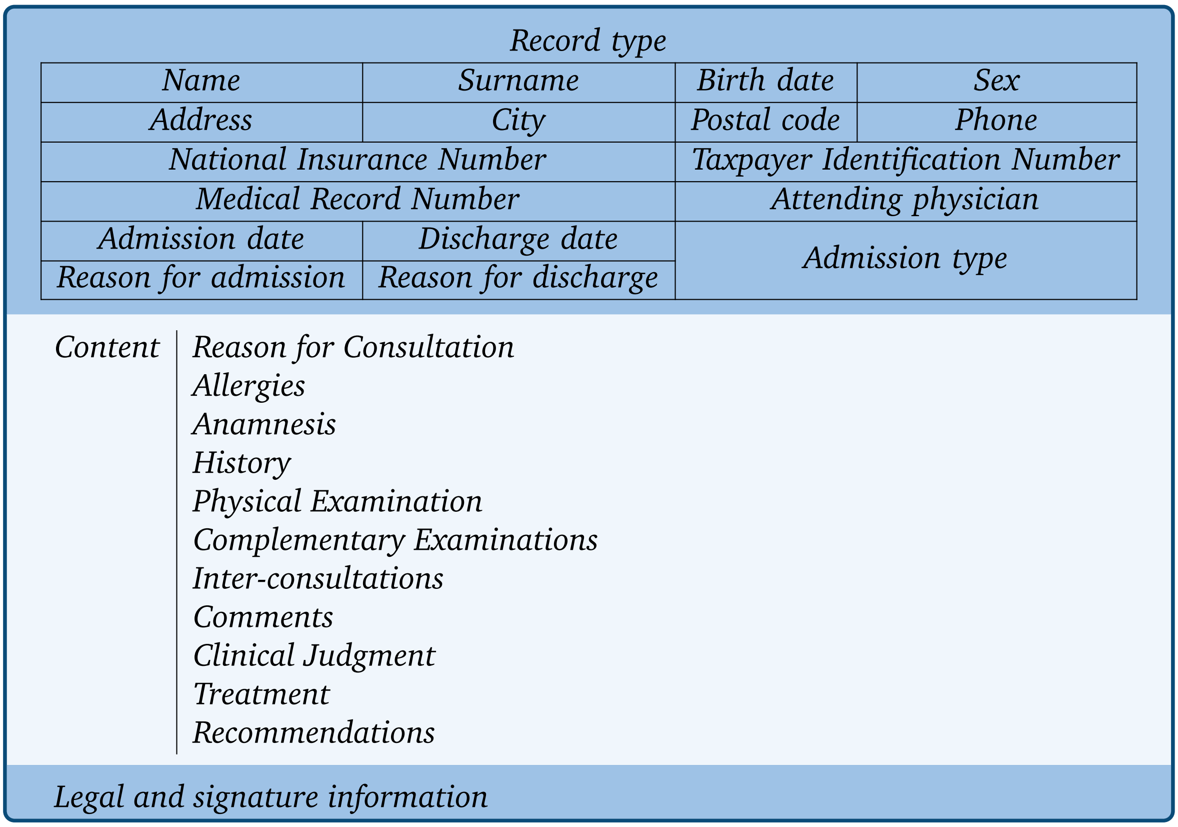
General description and structure

The main data set used in this thesis is the *HUFA* corpus, which collects 169,408 Spanish hospital discharge reports in EMR format. Of these, 36,312 records have been coded with the CIE-10-ES, only using final codes. The collected records are long documents in free-text format, with numerous paragraphs and different types of information from many sources. Overall, the corpus comprises about 1,243 megabytes of textual information altogether.

Records are generated using templates consisting of a header with the patient's data, different sections comprising the main content and a footer with legal and signature information. The type of the report is available in the headers. Figure 3.8 shows such structure defined in the hospital template. Although the content is usually separated into known sections such as *History*, *Treatment*, and *Clinical Judgment*, it is free text and often does not follow any guidelines.

We have examined the presence of each of these sections in the collection using IR techniques based on keywords. The percentage of records including the sections

Figure 3.8: *Structure of a HUFA record.*



Record type

Name	*Surname*	*Birth date*	*Sex*
Address	*City*	*Postal code*	*Phone*
National Insurance Number		*Taxpayer Identification Number*	
Medical Record Number		*Attending physician*	
Admission date	*Discharge date*	*Admission type*	
Reason for admission	*Reason for discharge*		

Content

- *Reason for Consultation*
- *Allergies*
- *Anamnesis*
- *History*
- *Physical Examination*
- *Complementary Examinations*
- *Inter-consultations*
- *Comments*
- *Clinical Judgment*
- *Treatment*
- *Recommendations*

Legal and signature information

is shown in Figure 3.9. *Clinical judgement* appears in most records (87%) and is one of the most informative sections for coding purposes. However, the entire record is required to be processed because relevant information is also found in other parts not always available or detected, such as *Procedures* and *History*.

Regardless of the high-level structure, the syntax of the records is framed in the clinical domain, which implies many particularities. For example, abrupt formatting changes and lack of uniformity predominate in records. Long descriptive paragraphs, lists of evidence and numerical analyses coexist in the same document. Furthermore, typographical errors, or typos, are frequent due to the urgency of services. In turn, physicians tend to use numerous official and even custom acronyms for saving time. Abbreviations are also common and significantly hamper lexical standardisation.

Coding criteria

Both the 2016 and 2018 version of the CIE-10-ES have been applied when coded the hospital discharged reports. The 2016 version includes 69,823 final diagnoses and 71,974 final procedures, resulting in a total of 141,797 final codes. The 2018 version introduces slight variations in some definitions and additional codes. In particular, 1,974 new diagnoses have been added to the nomenclature, while 311 of the old ones have been removed. In turn, 774 diagnostic descriptions have been modified. As for procedures, 3,827 new ones have been defined, 491 modified and 12 deleted. As a result, the 2018 version comprises 71,486 diagnoses and 75,789 procedures, totalling 147,275 final codes.

This hospital does not include all medical specialties, so it does not produce examples of determined code groups, especially procedures. Figure 3.10 shows a statistical description of *HUFA* reports types by specialty and types of procedures by

Figure 3.9: *Statistical study of the presence of the report sections.*

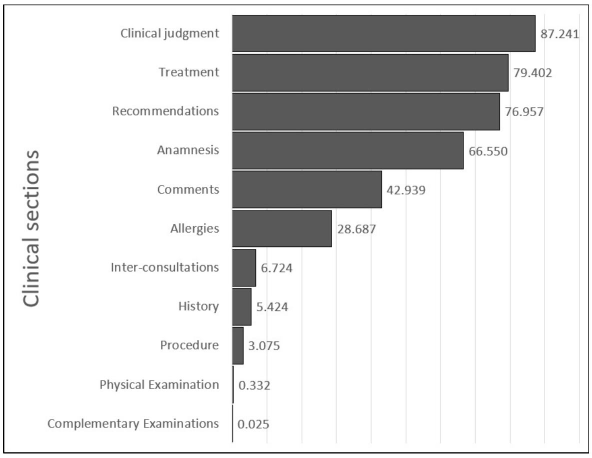
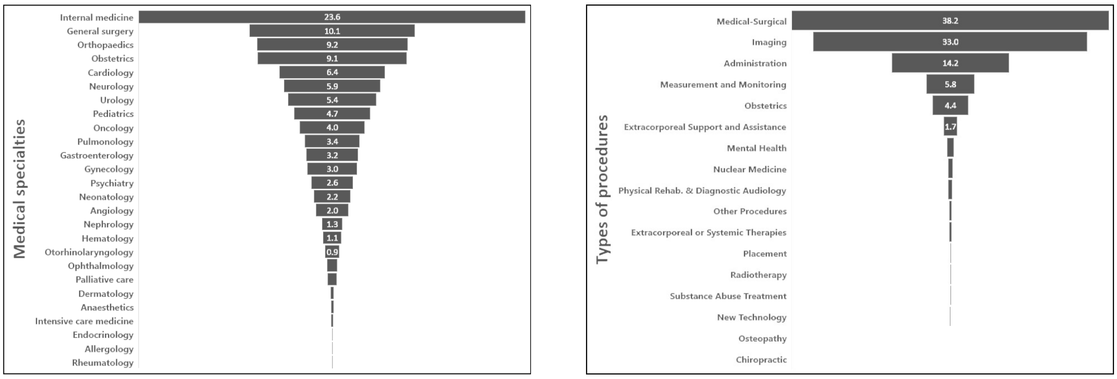


Figure 3.10: *Statistical analysis of the percentage of HUFA coded records per medical specialty and procedures per type.*



(a) *Distribution of HUFA records per medical specialty.*

(b) *Distribution of procedures per main category.*

category. *Internal medicine*, *General surgery*, *Orthopaedics*, and *Obstetrics* cover more than 50% of the collection. In contrast, other specialties such as *Dermatology* and *Endocrinology* are barely present in the data set. Although the type of report does not limit the category of codes it contains, as there are transversal procedures such as surgery, certain groups are less likely in other reports that do not correspond to the respective medical speciality. For example, there is a correlation between the increased presence of surgery and pediatrics reports and a higher number of surgical and pediatrics procedures. Similarly, the scarcity of psychological reports relates to the limited amount of mental health interventions.

The coding of *HUFA* records has been conducted by CIE-10-ES expert following all official specifications. Only those diagnoses and procedures relevant to or derived from the cause of the patient's hospitalisation were coded. Moreover, coding rules such as inclusions, exclusions, and additional associations have been respected, so that not all codes have originated from textual evidence.

CIE-10-ES annotations have been done at document level. Each record is associated with a list of diagnoses and procedures, so no information about which part of the document contains the clinical-coding textual evidences leading to the codes is provided. The first diagnosis for each record is always filled by the main cause of hospitalisation in the gold standard, while the rest of the codes do not follow any particular order.

Figure 3.11 shows the distribution of codes per document. Despite the extreme length of the reports, the average number of associated codes is 10, with 8 for diagnoses and 2 for procedures. Although there are documents with up to 43 annotated codes, most of them comprise between 3 and 12. In turn, less than 1% of the records include more than 25 codes.

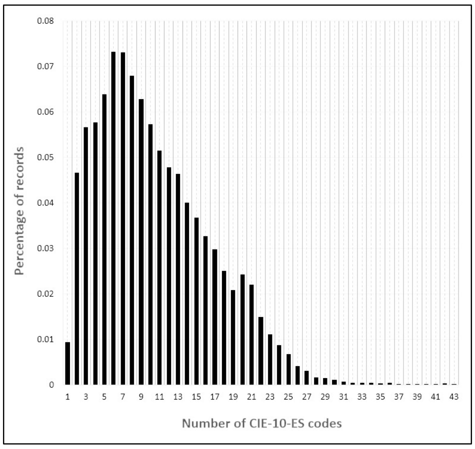
Example

Figures 3.12 and 3.13 includes a hypothetical example[2] of the content of a record translated into English. The untranslated example is shown in Appendix D.

The example includes multiple enumerations such as the list of antecedents. In this case, personal history is classified into several diagnoses such as "*PAH*" (Pulmonary Arterial Hypertension) associated with code I10 (*Essential (primary) hypertension*), "*Hansen's disease*" leading to code B92 (*Sequelae of leprosy*), and "*neoplasm in sigma*" resulting in code Z85.038 (*Personal history of other malignant neoplasm of large intestine*). More descriptive paragraphs are also present detailing for example physical explorations, in which there is information about a "*Colostomy in phase II*" that is classifiable as Z93.3 (*Colostomy status*).

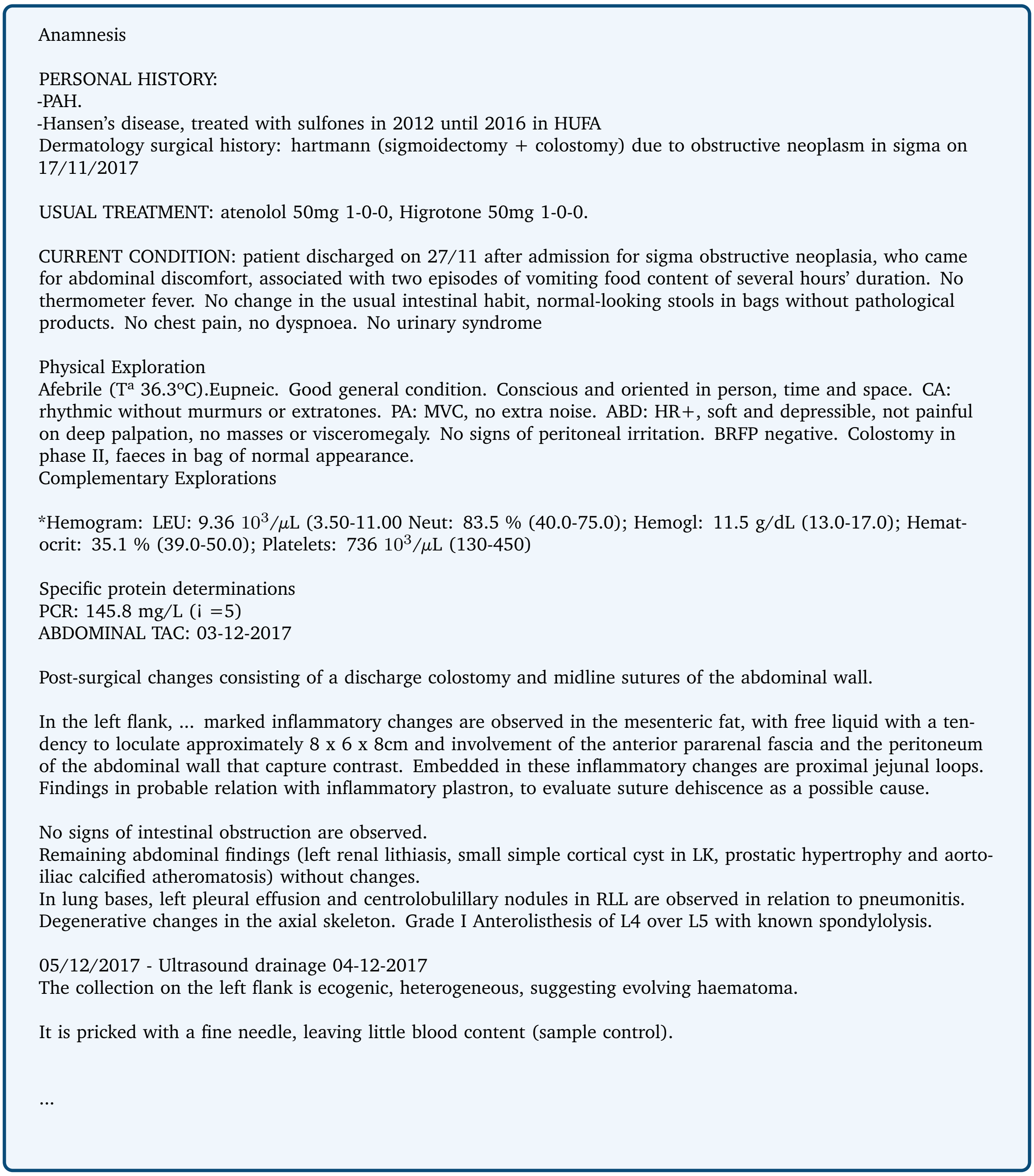
The textual elements that usually constitute the codes are neither isolated nor continuous, but are typically scattered and redundant. In the example above, the information leading to code Z93.3 can be found at different locations in the text in addition to section Physical Exploration, such as "*functioning colostomy*" in Evolution and "*left colon-colostomy*" in Treatment. Another example of redundancy occurs with the code 3E0G36Z (*Introduction of Nutritional Substance into Upper GI, Percutaneous Approach*), which can be located in sections Plan ("*initiate central TPN*") and Clinical Trial ("*TOTAL PARENTERAL NUTRITION*"). An example of discontinuous evidences is "*NG tube*" in section Plan and "*need for placement of NG tube persists...so decided to reintroduce surgery*" in Evolution, which are coded into 0DH67UZ (*Insertion of Feeding Device into Stomach, Via Natural or Artificial Opening*).

Figure 3.11: *Percentage of HUFA records per number of associated CIE-10-ES codes.*



[2] An original example cannot be given due to the European General Data Protection Regulation (GDPR).

Figure 3.12: *Example of the content of a HUFA Electronic Medical Record (part I). The original, untranslated example can be found in Appendix D.*



Anamnesis

PERSONAL HISTORY:
-PAH.
-Hansen's disease, treated with sulfones in 2012 until 2016 in HUFA
Dermatology surgical history: hartmann (sigmoidectomy + colostomy) due to obstructive neoplasm in sigma on 17/11/2017

USUAL TREATMENT: atenolol 50mg 1-0-0, Higrotone 50mg 1-0-0.

CURRENT CONDITION: patient discharged on 27/11 after admission for sigma obstructive neoplasia, who came for abdominal discomfort, associated with two episodes of vomiting food content of several hours' duration. No thermometer fever. No change in the usual intestinal habit, normal-looking stools in bags without pathological products. No chest pain, no dyspnoea. No urinary syndrome

Physical Exploration
Afebrile (T^a 36.3ºC).Eupneic. Good general condition. Conscious and oriented in person, time and space. CA: rhythmic without murmurs or extratones. PA: MVC, no extra noise. ABD: HR+, soft and depressible, not painful on deep palpation, no masses or visceromegaly. No signs of peritoneal irritation. BRFP negative. Colostomy in phase II, faeces in bag of normal appearance.
Complementary Explorations

*Hemogram: LEU: 9.36 $10^3/\mu$L (3.50-11.00 Neut: 83.5 % (40.0-75.0); Hemogl: 11.5 g/dL (13.0-17.0); Hematocrit: 35.1 % (39.0-50.0); Platelets: 736 $10^3/\mu$L (130-450)

Specific protein determinations
PCR: 145.8 mg/L (¡ =5)
ABDOMINAL TAC: 03-12-2017

Post-surgical changes consisting of a discharge colostomy and midline sutures of the abdominal wall.

In the left flank, ... marked inflammatory changes are observed in the mesenteric fat, with free liquid with a tendency to loculate approximately 8 x 6 x 8cm and involvement of the anterior pararenal fascia and the peritoneum of the abdominal wall that capture contrast. Embedded in these inflammatory changes are proximal jejunal loops. Findings in probable relation with inflammatory plastron, to evaluate suture dehiscence as a possible cause.

No signs of intestinal obstruction are observed.
Remaining abdominal findings (left renal lithiasis, small simple cortical cyst in LK, prostatic hypertrophy and aorto-iliac calcified atheromatosis) without changes.
In lung bases, left pleural effusion and centrolobulillary nodules in RLL are observed in relation to pneumonitis.
Degenerative changes in the axial skeleton. Grade I Anterolisthesis of L4 over L5 with known spondylolysis.

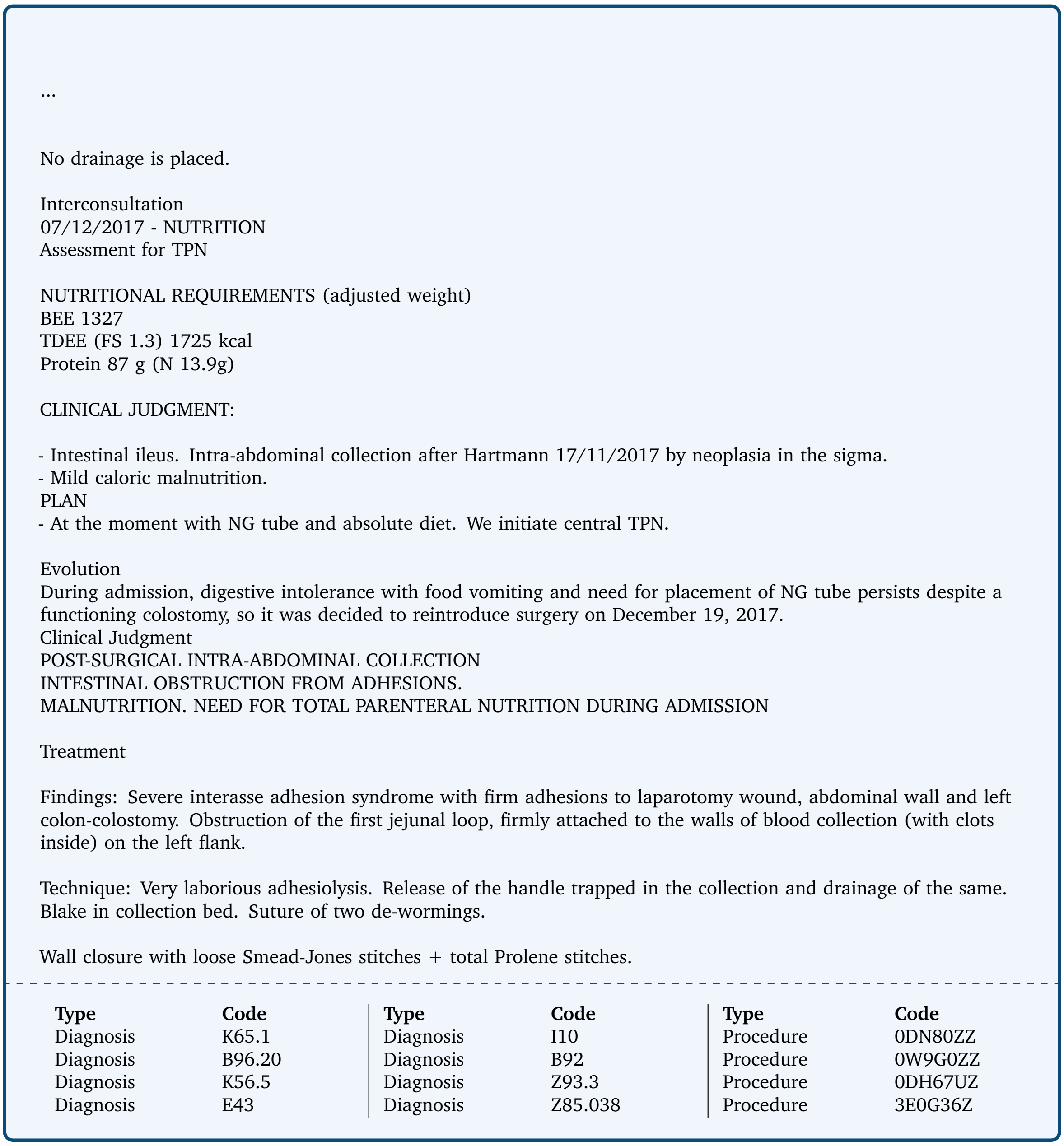
05/12/2017 - Ultrasound drainage 04-12-2017
The collection on the left flank is ecogenic, heterogeneous, suggesting evolving haematoma.

It is pricked with a fine needle, leaving little blood content (sample control).

...

CIE-10-ES coders use specific clinical expertise during annotation. For example, although the word "*malnutrition*" is mentioned several times, it is necessary to know the appropriate levels of protein and calories, as well as recognising that "*BEE*" and "*TDEE*" are the Basal Energy Expenditure and Total Daily Energy Expenditure, in order

Figure 3.13: *Example of the content of a HUFA Electronic Medical Record (part II). The original, untranslated example can be found in Appendix D.*



...

No drainage is placed.

Interconsultation
07/12/2017 - NUTRITION
Assessment for TPN

NUTRITIONAL REQUIREMENTS (adjusted weight)
BEE 1327
TDEE (FS 1.3) 1725 kcal
Protein 87 g (N 13.9g)

CLINICAL JUDGMENT:

- Intestinal ileus. Intra-abdominal collection after Hartmann 17/11/2017 by neoplasia in the sigma.
- Mild caloric malnutrition.

PLAN
- At the moment with NG tube and absolute diet. We initiate central TPN.

Evolution
During admission, digestive intolerance with food vomiting and need for placement of NG tube persists despite a functioning colostomy, so it was decided to reintroduce surgery on December 19, 2017.
Clinical Judgment
POST-SURGICAL INTRA-ABDOMINAL COLLECTION
INTESTINAL OBSTRUCTION FROM ADHESIONS.
MALNUTRITION. NEED FOR TOTAL PARENTERAL NUTRITION DURING ADMISSION

Treatment

Findings: Severe interasse adhesion syndrome with firm adhesions to laparotomy wound, abdominal wall and left colon-colostomy. Obstruction of the first jejunal loop, firmly attached to the walls of blood collection (with clots inside) on the left flank.

Technique: Very laborious adhesiolysis. Release of the handle trapped in the collection and drainage of the same. Blake in collection bed. Suture of two de-wormings.

Wall closure with loose Smead-Jones stitches + total Prolene stitches.

Type	**Code**	**Type**	**Code**	**Type**	**Code**
Diagnosis	K65.1	Diagnosis	I10	Procedure	0DN80ZZ
Diagnosis	B96.20	Diagnosis	B92	Procedure	0W9G0ZZ
Diagnosis	K56.5	Diagnosis	Z93.3	Procedure	0DH67UZ
Diagnosis	E43	Diagnosis	Z85.038	Procedure	3E0G36Z

to assign the code E43 (*Unspecified severe protein-calorie malnutrition*). Similarly, it is necessary to deal with the meaning of "*adhesiolysis*" in order to accurately identify the code 0W9G0ZZ (*Drainage of Peritoneal Cavity, Open Approach*).

Textual descriptive statistics

There are more than 182 million tokens distributed in 19 million sentences, with a vocabulary of around 400,000 different tokens. It involves more than one gigabyte of textual information. Table 3.5 summarises the statistical parameters used to describe the lexical structure of the corpus.

	Previous years (uncoded)	2016	2017	2018	Total
Records	128,339	13,177	15,404	12,488	169,408
Sentences	12,521,865	2,036,614	2,513,005	2,102,686	19,174,170
Unique sentences	4,154,643	572,713	696,730	587,903	5,658,959
Tokens	131,354,382	15,611,046	19,049,676	16,320,479	182,335,583
Unique tokens	364,794	108,231	120,399	106,173	444,087
Sentence average per record	97.57±55.75	154.56±103.58	163.14±111.46	168.38±114.60	113.18±77.82
Token average per sentence	10.49±15.13	7.67±8.78	7.58±8.59	7.76±8.94	9.51±13.34
Token average per record	1,023.50±784.00	1,184.72±864.37	1,236.67±906.97	1,306.89±954.57	1,076.31±821.59

Table 3.5: *Statistical description of the records from HUFA corpus.*

In general, the records are highly verbose, with multiple paragraphs and an average of more than 1,000 tokens in 100 sentences. As for the extremes, the longest record comprises 14,000 token spread over 1,600 sentences. In contrast, the shortest one contains only three words composing one sentence. These are rare cases as only 27 records surpass the 1,000 sentences and 5 include less than 10.

There are no significant variations in the parameters from 2016, 2017, and 2018 data sets (which comprise the coded reports), although a constant increase in the average length of the documents and number of sentences can be appreciated. However, there are more differences with reports from years prior to 2016, exhibiting fewer and longer sentences.

Coding descriptive statistics

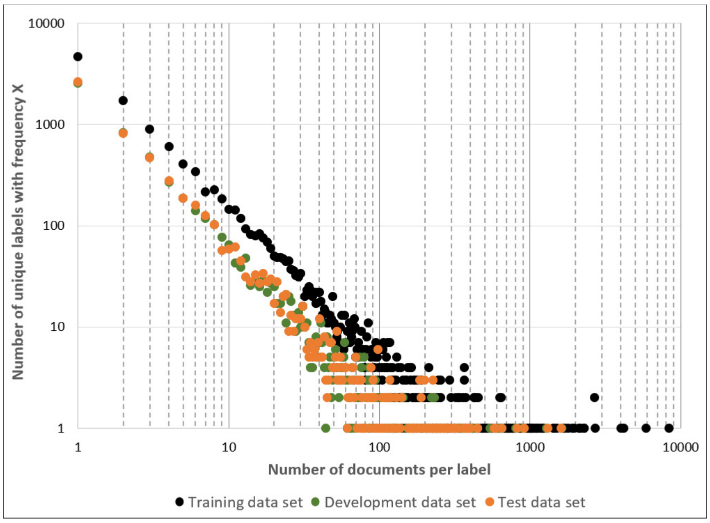
Due to the numerous changes and updates of the ICD, the *HUFA* collection gathers records from different years, so there are differences in the annotation of codes. The change from the 9th to the 10th version of the CIE happened in 2016, and an update of the latest was released in 2018. For this reason, only the records collected after 2015 have been coded with the CIE-10-ES. Although the 128,339 records with no associated ICD-10 codes cannot be used directly for the task, those EMRs can be useful in modelling the clinical language.

As regards the splitting of the codified records into subsets, the proportions of 70%, 15%, and 15% have been used to constitute the training, development, and test data sets respectively. Table 3.6 shows the number of codes in the final division.

In total there are 305,003 diagnostic and 75,779 procedure annotations, involving 13,706 unique codes. This is about 10% of all possible codes, with 12.5% for diagnoses and 7% for procedures. Despite more than 11,000 different codes in the training set, 18% of the development and test codes are unseen. This is important to be considered when designing a proposal, as a high percentage of zero-shot codes would have to be dealt with.

Focusing on the frequency of codes, we find the same distribution as in the other collections. Figure 3.14 shows the exponential trend followed by frequency, with a few very frequent codes and many infrequent ones. For instance, the most frequent code, I10 (*Essential hypertension*), appears in 30% of all records, while about 80% of codes are present in less than 10 records.

Figure 3.14: *Distribution of the CIE-10-ES codes in the HUFA corpus. Each dot symbolizes the number of codes Y with the same frequency X.*



	Train	Dev.	Test	Total
Records	25,765	5,252	5,295	36,312
Diagnosis codes	214,523	44,953	45,527	305,003
Procedure codes	52,457	11,717	11,605	75,779
Total CIE-10-ES codes	266,980	56,670	57,132	380,782
Unique diagnosis codes	7,627	3,867	3,943	8,732
Unique procedure codes	4,232	1,762	1,794	4,974
Total unique CIE-10-ES codes	11,859	5,629	5,737	13,706
Unique unseen diagnosis codes	-	622	657	1,279
Unique unseen procedure codes	-	397	418	815
Total unique unseen CIE-10-ES codes	-	1,019	1,075	2,094
CIE-10-ES code average per record	10.65±6.27	11.53±6.77	10.34±6.11	10.49±6.15

Table 3.6: *Statistical description of the CIE-10-ES codes from HUFA corpus.*

Feature summary

The corpus could be summarised in the following attributes based on the sections described above:

- **Spanish** is the language used in the records. Language flexibility and the limited availability of Spanish NLP tools in the clinical domain pose additional challenges.
- The records show very varied sizes, with a standard deviation close to the mean. At the same time, the mean is high, more than 1,000 tokens per document, implying **large lengths**.
- Length introduces **considerable lexical variability**. There is an average of 8 new tokens per coded record and almost 3 per record prior to 2016.
- Coding criteria follow **official guidelines**, identifying diagnoses and procedures from the synthesis of contextualised information and respecting coding rules.
- The codes are significantly unbalanced. The volume of the corpus is sufficient to collect examples of up to 13,706 diagnoses and procedures, which entails a **coverage** of almost **10%**.

3.3 Pre-processes for data standardisation

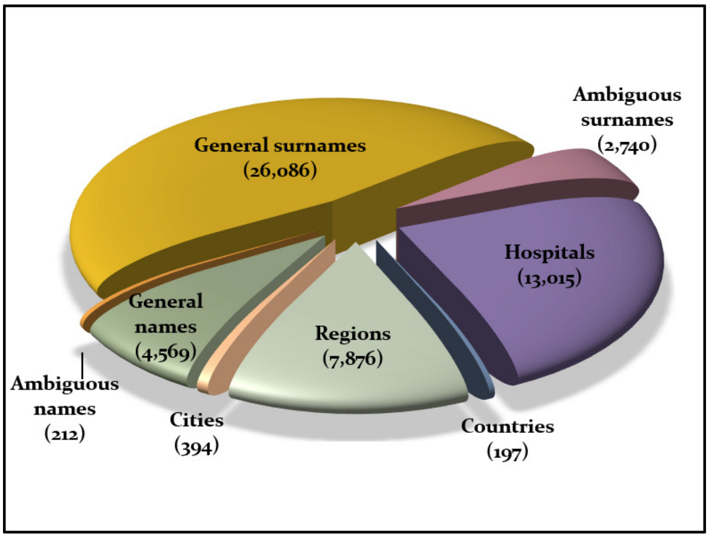
The raw data used in this thesis have been described in Section 3.2; however, the EMRs are free text and cannot be used directly. The format should be standardised before entering the data in the ICD coding proposals. For this purpose, several pre-processes have been applied in order to facilitate the subsequent representation of the records.

Firstly, anonymisation has been necessary for the use of real data from health centres, so we have designed our own proposal for immediate application in Section 3.3.1. Besides, one of the keys when processing a medical record is the correct detection of the boundaries of sentences and words due to the wide variety of formats. We have therefore implemented a tokeniser adapted to the domain and languages present in the corpora described in Section 3.3.2. Finally, a lexical standardization should be included to try to reduce the noise in the text. Section 3.3.3 addresses this issue by focusing on the Spanish language for which more data are available. Those pre-processing steps have been detailed below.

3.3.1 Anonimization

A medical record is a written and ordered description of all the information about a patient that is relevant to the final judgement of a diagnosis. Therefore, it may

Figure 3.15: *Statistical description of the lists of entities used in anonymisation.*



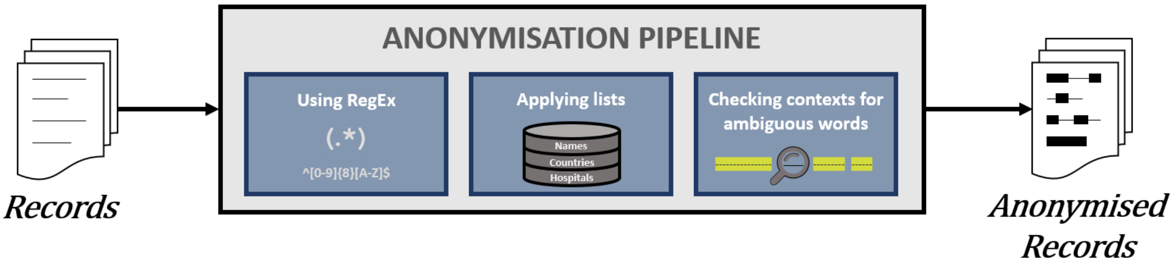
contain data related to the physiology, mental state, social and work environment, and habits, both of the patient and of his/her relatives; in short, any information that may influence the causes and evolution of conditions and diseases (Piqueras, 2009). In this way, any record from health centres contains a great quantity of sensitive information.

The collection, processing, and transfer of such information are highly regulated in most countries to prevent inappropriate use. Current privacity policies severely limit the access to clinical data for epidemiological, public health, research or educational purposes. For this reason, exploring ICD collections requires excluding all data identifying the patient. The death certificates from the *Causes of Death* corpus and the EMRs from the *CodiEsp* corpus have been generated exclusively for a shared task and have therefore been previously anonymised. In contrast, the *HUFA* records store the raw data, with all the information relating to the patients. It has therefore been necessary to implement anonymisation methods for processing such records.

Health Insurance Portability and Accountability Act (HIPAA) identified the entities susceptible to the identification of patients and established a standard for anonymisation in the United States of America (USA). The MEDDOCAN (Marimon et al., 2019) task organised by the *Plan de Impulso de las Tecnologías del Lenguaje* (Spanish Language Technologies Promotion Plan) has recently intended to establish a similar standard. Nevertheless, given the lack of consensus on the information to be anonymised in Europe, a compromise was reached with the hospital to anonymise a set of selected entity types: names and surnames, dates and times, personal identifiers such as the ID number, addresses and regions, telephone numbers, and health centre names. We have implemented a rule-based approach[3] because of the lack of Spanish clinical data

[3] The software can be downloaded at `https://zenodo.org/record/5148968#.YQRIQ477SUk`

Figure 3.16: *Pipeline for anonymisation process.*



sets at the beginning of this thesis. To this end, the statistics of the most frequent names, surnames, hospitals, countries, regions, and cities were previously obtained from the *Instituto Nacional de Estadística* (Spanish National Statistical Institute, INE)[4]. After manual examination, words with other meanings in the domain were separated, creating two sets of lists: those with a single, general meaning and those clinically ambiguous. Figure 3.15 shows the number of entities included in every lists.

Figure 3.16 provides an overview of the methods involved. Three steps have been applied during anonymisation. First, Regular Expression (RegEx) techniques have been applied for the identification of numerical data such as times, dates and personal identifiers. Then, lists of words with general meanings have been used to directly remove such information from the records. Lastly, the context is used to decide on ambiguous words by identify certain nearby expressions or words in order to avoid removing information excessively. In this way, names such as *Dolores* (meaning *pains* in Spanish) are anonymised if they are preceded by other sensitive information such as surnames, or other expressions are found suggesting that the word refers to the name of the patient, a relative, or the doctor, such as "*the patient*", "*his mother*", "*Mrs.*", and "*Dr.*".

All anonymised entities have been replaced by tags corresponding to each type. After applying the anonymisation process to the whole *HUFA* reports, a difference of 14,954,689 tokens is observed when excluding the tags, which means a decrease of 8 per cent. Although this may seem a high number, it should be noted that all reports include frequent date references and a header with numerous personal details. Still, this loss of information is expected to be irrelevant for coding purposes.

3.3.2 Sentence boundary detection and tokenization

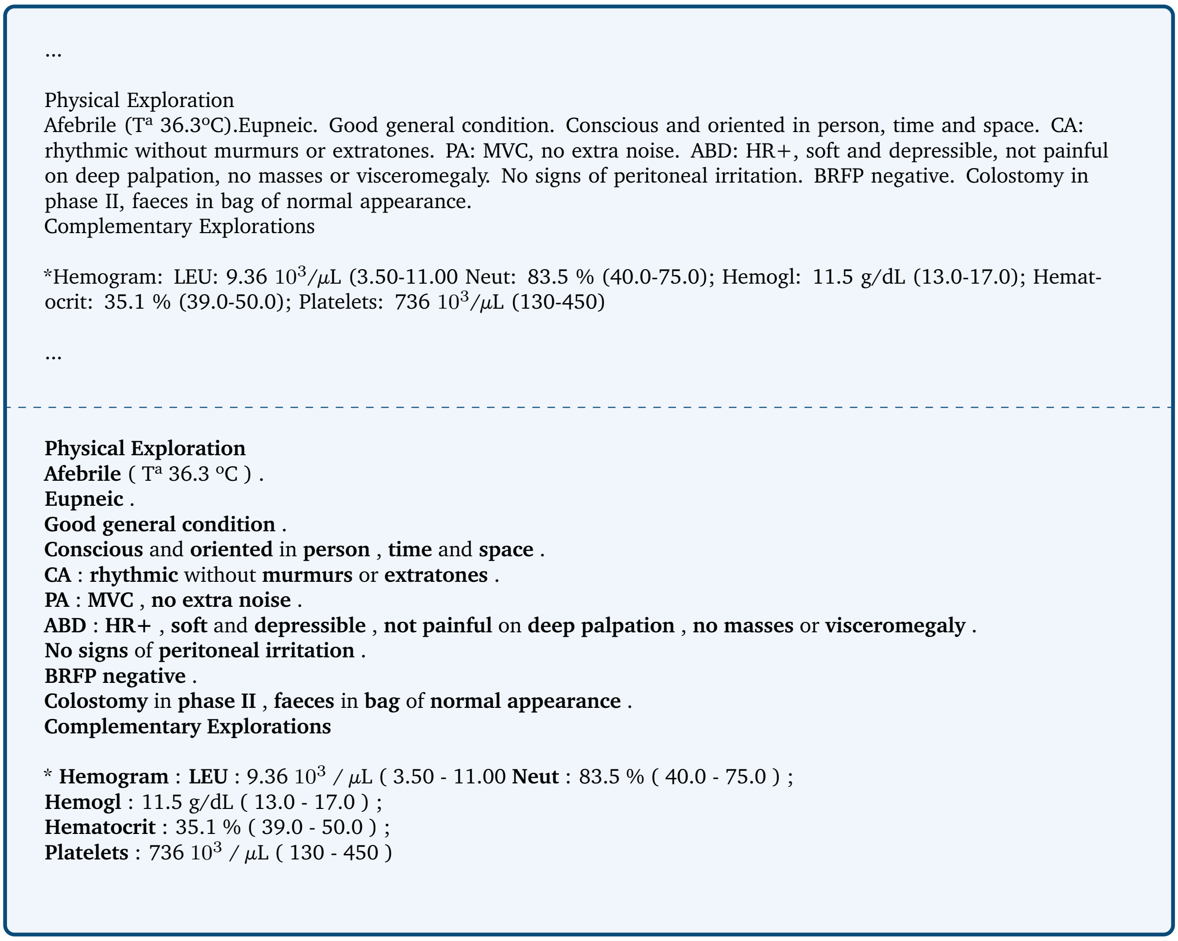
The content of the records does not reflect a conventional narrative structure, such as news, literary stories, and scientific articles. Instead, clinical text includes both short sentences and long paragraphs. Sometimes it adopts the schematic format,

[4] `https://www.ine.es/`

with numerous enumerations and incomplete syntax, and other times it uses a very descriptive and verbose format. Besides, the presence of grammatical errors and incoherent formatting, as well as the lack of punctuation marks or the indiscriminate use of capital letters to highlight some parts of the text, hinder the correct identification of sentences and words.

The breakdown of the text into coherent fragments is essential to represent the information more accurately. This process, also called tokenization, is one of the key components in NLP approaches. Although popular supervised tokenizers demonstrate good performance in general-purpose domains, they are not well suited to the flexibility of formatting that clinical reports present. Neither there is a large volume of publicly available annotated clinical documents on which to implement a supervised tokenizer. For this reason, we have proposed a customized rule-based tokenizer.

Figure 3.17: *Translated example of the tokenization of two paragraphs of the report shown in Figure 3.12. The original text is shown at the top, while the tokens separated by spaces are shown below.*



...

Physical Exploration
Afebrile (T^a 36.3°C).Eupneic. Good general condition. Conscious and oriented in person, time and space. CA: rhythmic without murmurs or extratones. PA: MVC, no extra noise. ABD: HR+, soft and depressible, not painful on deep palpation, no masses or visceromegaly. No signs of peritoneal irritation. BRFP negative. Colostomy in phase II, faeces in bag of normal appearance.
Complementary Explorations

*Hemogram: LEU: 9.36 $10^3/\mu$L (3.50-11.00 Neut: 83.5 % (40.0-75.0); Hemogl: 11.5 g/dL (13.0-17.0); Hematocrit: 35.1 % (39.0-50.0); Platelets: 736 $10^3/\mu$L (130-450)

...

Physical Exploration
Afebrile (T^a 36.3 °C) .
Eupneic .
Good general condition .
Conscious and **oriented** in **person** , **time** and **space** .
CA : **rhythmic** without **murmurs** or **extratones** .
PA : **MVC** , **no extra noise** .
ABD : **HR+** , **soft** and **depressible** , **not painful** on **deep palpation** , **no masses** or **visceromegaly** .
No signs of **peritoneal irritation** .
BRFP negative .
Colostomy in **phase II** , **faeces** in **bag** of **normal appearance** .
Complementary Explorations

* **Hemogram** : **LEU** : 9.36 10^3 / μL (3.50 - 11.00 **Neut** : 83.5 % (40.0 - 75.0) ;
Hemogl : 11.5 g/dL (13.0 - 17.0) ;
Hematocrit : 35.1 % (39.0 - 50.0) ;
Platelets : 736 10^3 / μL (130 - 450)

Regular expressions have been used to identify all sentences and words in records. The idea is to separate possible punctuation marks adjacent to the words, such as commas in enumerations and dots at the end of sentences. However, certain heuristics have been included to avoid punctuation marks pertaining to the words themselves, such as dots separating decimals and hyphens from compound words. Hence, compound words such as “*medical-surgical*”, “*β-carotene*”, and “*NT-proBNP*” remain as a single token.

Figure 3.17 shows the output of the tokenisation process when applied to a fragment of the example from Section 3.2.3. As shown, the defined set of rules are able to correctly separate most of the text.

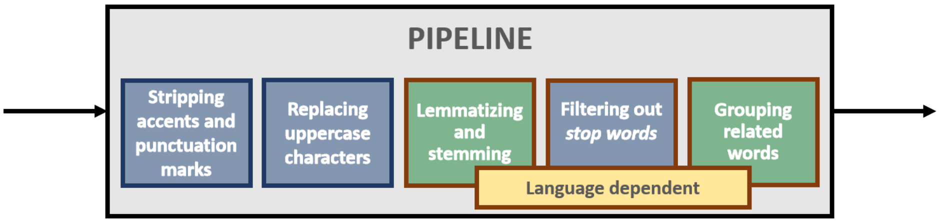
3.3.3 Lexical standardisation

The records used in this thesis are written in Romance (French, Italian and Spanish) and Uralic (Hungarian) languages. So no data have been used in English, for which there are many clinical-domain tools. For this reason, basic NLP processes have been designed and implemented instead of using widely popularised packages such as Natural Language Toolkit (NLTK) (Porter et al., 1980) or spaCy[5].

Typos, acronyms, and abbreviations are frequent in the records, which makes the content very diverse. Grammatical mistakes are also common in medical writing (Lai et al., 2015a), such as incorrect placement of accent marks. In addition, the clinical domain is characterised by a very broad and specific vocabulary, full of synonyms and possible variations. To all this should be added the large number of derivations and word forms in the Romance and Uralic languages.

Lexical diversity complicates the subsequent representation of the reports as it tends to increase the dimensionality of the features, which can lead to problems of scattering. Given the preceding aspects, the pipeline shown in Figure 3.18 has been implemented to prepare the data for the proposals. Although some of these pre-processes have not been performed for certain experiments, the overall details are provided below.

Figure 3.18: *Pipeline for pre-processing reports before representation.*


PIPELINE
Stripping accents and punctuation marks
Replacing uppercase characters
Lemmatizing and stemming
Filtering out *stop words*
Grouping related words
Language dependent

[5] https://spacy.io

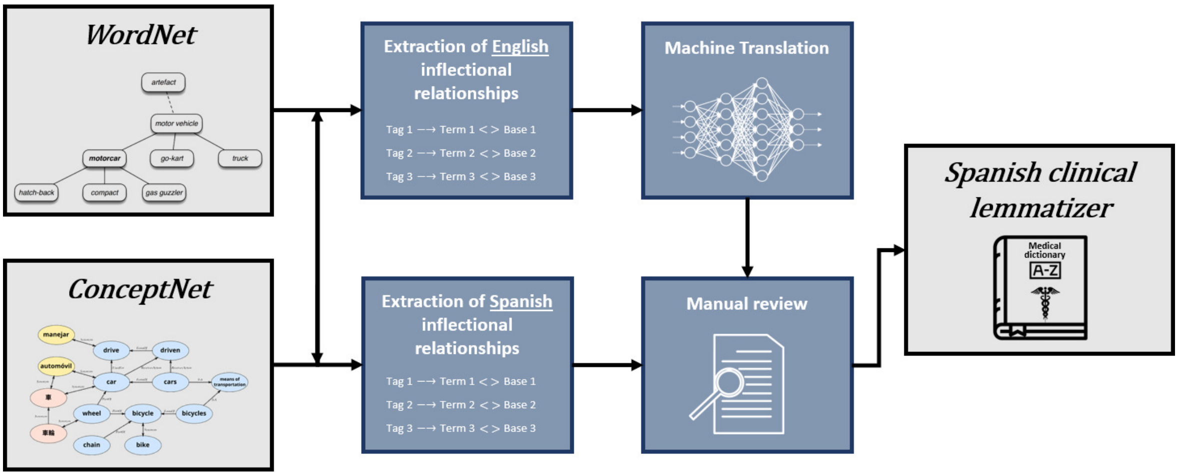
Stripping accent marks and puntuaction marks Punctuation marks do not usually provide much information in this domain given the frequent lack of syntax. Thus, the customized tokenization applied to identify the tokens has also been used to detect punctuation marks for immediate elimination. As far as accent marks are concerned, the lack of coherence often leads to the presence of different forms of the same word, accented and unaccented. For this reason, a character conversion is performed removing all accent marks, e.g., the token "*clínico*" and its incorrect forms ("*clinico*", "*cliníco*", and "*clinicó*") would be transformed into "*clinico*". This approximation may result in information loss leading to some ambiguity; however, an overall improvement in representation is expected.

Replacing uppercase characters The use of capital letters in the clinical domain is common to highlight expressions, and even whole sentences. A character transformation is again performed to avoid different representations of the same word. In this way, the fragment "*CLINICAL JUDGMENT*" would be replaced by the expression "*clinical judgment*". Although information on Named Entities may be lost, it is expected that this transformation will bring more quality to subsequent representations.

Stemming and lemmatization Stemming and lemmatisation transform the inflectional and derived forms of words into the common basic forms. These processes are widely used in NLP to reduce lexical diversity. As for the differences, stemming exploits heuristic rules to eliminate derivational affixes, while lematization combines morphological and lexical knowledge to return dictionary forms (lemmas).
The first one groups derivationally related words into abbreviated forms called stems, such as *infection*, *infectious*, *infected*, and *infectiousness* which are transformed into *infect*. Although the use of general rules provides greater coverage, it is also more likely to confuse lexically similar but semantically distinct words. This type of failure is very frequent in the Spanish clinical domain, e.g., the Spanish root *col* would gather both *col* (*cabbage*), *colar* (*to strain something*), and *cola* (*tail* or *glue*). On the contrary, the second one is focused on linking inflectional forms. Its reliance on dictionaries reduces the scope but ensures greater accuracy.
A Spanish lemmatizer built from WordNet (Miller, 1998) and ConceptNet (Speer, Chin, and Havasi, 2017) has been used to standardize the text with a broad coverage. Figure 3.19 shows an overview of the creation process. First, Spanish and English relationships between words have been extracted from both ontologies. Although Spanish entries have been prioritised, ML techniques have been applied to the English resources by extending the available sources. Then, a semi-automatic revision using lexical similarity techniques has produced the final resource, with more than 841,144 entries and 75,951 roots.
Lexical disambiguation does not seem to be particularly relevant to the ICD coding

task, so the most frequent lemma has been retrieved instead of analysing grammatical function. The use of this knowledge-based lemmatizer has been preferred to other tools based on supervised models, such as spaCy for general domain and IxaMed (Atutxa et al., 2018) for clinical domain, as we have noticed that it provide greater in the experimental tests.
A lemmatization followed by a stemming process has been applied in order to exploit both attributes. The idea is to retrieve the lemmas of the words available in the dictionary. All the words not found by the lemmatizer are subsequently stemmed. The conventional stemmer supplied by the NLTK library (Porter et al., 1980) has been used.

Figure 3.19: *Pipeline for the creation of the lemmatizer.*


WordNet
artefact
motor vehicle
motorcar
go-kart
truck
hatch-back
compact
gas guzzler
ConceptNet
manejar
drive
driven
automóvil
car
cars
means of transportation
車
wheel
bicycle
bicycles
車輪
chain
bike
Extraction of English inflectional relationships
Tag 1 ⟶ Term 1 < > Base 1
Tag 2 ⟶ Term 2 < > Base 2
Tag 3 ⟶ Term 3 < > Base 3
Machine Translation
Extraction of Spanish inflectional relationships
Tag 1 ⟶ Term 1 < > Base 1
Tag 2 ⟶ Term 2 < > Base 2
Tag 3 ⟶ Term 3 < > Base 3
Manual review
Spanish clinical lemmatizer
Medical dictionary
A-Z

Filtering out words A general trend in NLP is the removal of very common words that hardly add any meaning to the text. Numerous stop word lists have been published for the general domain; however, such lists have been found not to be totally effective because of the introduction of ambiguity. Certain words that are normally not very meaningful occupy a very specific semantic space in the clinical domain, e.g., *haber* is the verb *To be* and coexists with the *Haber* syndrome. Another example would be the Spanish determinant *les*, often used as an acronym for *Systemic Lupus Erythematosus*.

Grouping related words The use of synonyms and related words is one of the factors that introduces greater diversity. For this reason, the replacement of words with similar meanings has been applied by exploiting customized lists of related words.

On the one hand, pertainyms[6] from Wordnet and ConceptNet have been extracted in the same way as lemmas. Manual rather than semi-automatic inspection has been carried out to check for consistency of meanings, as related words may not share the same root. An example would be the group for *allergy* and *allergic*, or *mouth* and *oral*. A total of 2,167 groups with 17,915 words have been collected.

Lexical similarity techniques have also been used to extract variants of expressions from the ICD manuals. The method exploits the redundancy of terms for certain descriptions. After a subsequent manual examination, 607 groups with 3,084 words have been extracted, such as *glottis*, *glottic*, *nephrosis*, and *nephrotic*.

Finally, the SNOMED CT ontology has been used to extract in-domain synonyms. To this end, only equivalent concepts with a maximum of 3 words excluding stop words have been selected. 980 single-word synonyms have been collected in addition to 23,639 interchangeable expressions with 2 or more words. Examples of synonyms are "*platelet*" and "*thrombocyte*"; "*malaria*", "*paludism*", and "*plasmodiosis*"; "*oesophagogastropexy*" and "*gastro-oesophagopexy*". In terms of expressions, examples include "*hepatotomy*" or "*liver incision*"; "*premature eruption of the tooth*" or "*premature dentition*"; "*strongyloidiasis*", "*strongyloidiosis*", or "*strongyloides infection*"'.

3.4 Evaluation

Evaluation needs to be properly defined before analysing the results of any proposal. For this purpose, which aspects should be compared with the reference data and the way to measure their similarity are established.

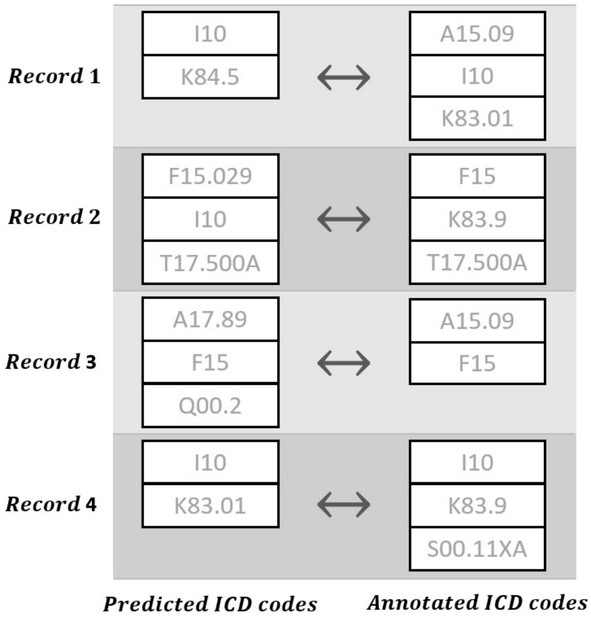
In our case, the reference data are the ICD-10 codes annotated by the specialists according to the criteria described in Section 3.2 for each corpus, while the results are the codes predicted for each record by the proposal. Hereby, the evaluation function aims at quantifying how similar the predicted code sets are to the annotated code sets. An example of the predicted and reference codes for four records is shown in Figure 3.20. Since this similarity or score will be used for decisions on approaches, it is relevant to define the method of computation.

Such scores measure the performance of a particular approach on a specific data set, so the values depend on both factors: approach and data set. Therefore, scores individually does not provide information about the quality of a method because these also depend on the properties of data sets. The results obtained need to be contextualised by relativising the scores by means of baselines.

Below are detailed both the evaluation functions capturing all the decisions that have been established to compare the predicted and annotated code sets, and the

[6] Pertainyms are words, usually adjectives, which can be defined as "of or pertaining to" another words.

Figure 3.20: *Example of the predicted and gold standard subsets of codes for a set of records.*



baselines describing the reference performance scores.

3.4.1 Evaluation functions

Part of the core work of this thesis has been to participate in ICD-10 competitions. In fact, as discussed in Section 3.2, data collections *Causes of Death* and *CodiEsp* have been acquired for experimentation with the participation in shared tasks. Initially, we started using the same metrics as these competitions, assessing the same aspects and using **exact matche**s. But as this study has evolved, the evaluation metrics have been gradually improved to capture all the implicit features of the ICD-10 task, from **partial matches** exploiting the hierarchical structure to the inclusion of **propensity** to weight the frequency.

Exact matches

The most widespread evaluation metric in classification is F-score (F_β), and consequently Precision (P) and Recall (R). These are the metrics used in the CLEF eHealth 2018 competition, and the starting point in the evaluation of the results of this thesis. The conventional metrics are based on exact matches, judging a system capable of predicting all the codes in records. In this context, Precision is the percentage of relevant codes among the predicted ones, while Recall is the percentage of relevant codes among all the possible relevant ones. Both metrics have been defined in Equations 3.1 and 3.2 in terms of True Positives (TP), False Positives (FP) and False Negatives (FN). In this context, True Positives are the relevant predicted codes, False Positives are the non-relevant predicted codes, and False Negatives are the relevant non-predicted codes. The sum $TP + FP$ is the number of predicted codes and is determined by the approach, while the sum $TP + FN$ is the number of associated codes in the reference collection. Finally, F-Score is the harmonic mean of Precision and Recall as described in Equation 3.3.

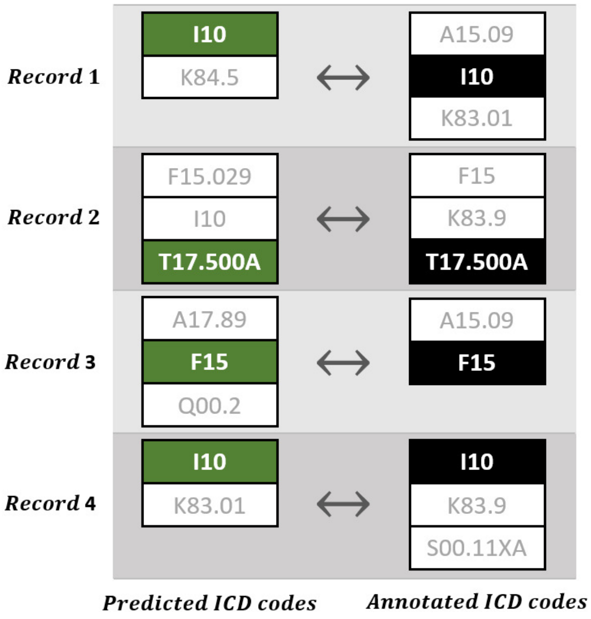
$$P = \frac{TP}{TP + FP} \tag{3.1}$$

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \tag{3.3}$$

Figure 3.21 shows an example of an evaluation based on exact matches for ICD-10 coding. The example comprises 4 TP or predicted matches in dark green color (*F15*, *I10*, and *T17.500A*), 6 FP or predicted non-matches (*A17.89*, *F15.029*, *K83.01*, *K84.5*, and *Q00.2*), and 7 FN or relevant non-matches (*A15.09*, *F15*, *K83.01*, *K83.9*, and *S00.11XA*), involving 0.4 Precision, 0.36 Recall, and 0.38 F-measure.

Figure 3.21: *Example of exact matches between the predicted and gold standard subsets of codes for a set of records. Exact matching codes have been highlighted in dark green for the predictions and in black for the gold standard. Otherwise, unmatched codes are blank.*



There are different ways to group successes and failures when dealing with multiple codes. Micro-average score provides a measure of the overall performance of the system regardless of minority codes. Conversely, macro averages involve the harmonic mean of code performance so that the hits are weighted with the inverse of frequency.

$$P_{macro} = \frac{\sum_{i=1}^{N_{rel}} P_i}{N_{rel}} \tag{3.4}$$

$$R_{macro} = \frac{\sum_{i=1}^{N_{true}} R_i}{N_{true}} \tag{3.5}$$

Equations 3.4 and 3.5 define macro averages of Precision (P_{macro}) and Recall (R_{macro}), where P_i and R_i are the score of each code, N_{rel} is the number of relevant (or predicted) codes, and N_{true} is the number of true (or gold standard) codes in the reference collection. Macro-average F-Score would again be the harmonic mean of the previous scores, as specified in Equation 3.3. Both micro and macro averages expecting significant differences in extreme imbalance scenarios such as the one exhibited by ICD coding. Micro-average score will match the performance of the majority classes as they comprise most instances, while macro-average score will ensures equal representation for all codes promoting less frequent ones.

The macro-average would yield a Precision score of 0.31, a Recall score of 0.36, and an F-score of 0.33 in the previous example (Figure 3.21). The macro-average Precision (P_{macro}) would be computed as the mean of the Precision scores of the 8 predicted codes, while the macro-average Recall (R_{macro}) would be the mean of the Recall scores of the 7 gold standard codes. Both values would be null for all codes except for *F15*, *I10*, and *T17.500A*, which would reach Precision values of 1.0, 0.67, and 1.0, and Recall values of 0.5, 1.0, and 1.0 respectively.

These functions are less meaningful if we focus on ICD-10 coding as a Extreme Multi-label Text Classification problem. The XMTC assumes an unbalanced distribution over a huge set of labels, with an innate predisposition to produce false positives as the number of potential labels is far greater than the number of labels truly associated with the instances. In terms of ICD-10 coding, each record has at most a few dozen associated codes out of thousands. This means that it is less likely and therefore much more difficult to hit, or predict TP, than not to miss, or avoid FN. This reduces the relevance of FN.

As mentioned, our purpose is to recommend codes for supporting ICD specialists, so it is more important to propose useful codes for the annotator, even if the predictions include less relevant codes, than to develop a system with restrictive tendencies to not suggest any code at all. The way to prioritise hits over misses is to deal with outputs as code rankings instead of sets and measure hits over fixed ranges, which length is denoted by the constant K. By freezing K, we keep constant the sum $TP + FP$, which was previously conditioned by the approach, so Precision (P) only varies as a function of TP. As a consequence, P also provides information on Recall (R) for a particular collection, since the sum $TP + FN$ was conditioned by the data set: the higher P, the higher R. Hence, the information provided by P, R and F-Score is redundant when comparing the performance of different approaches on the same collection. Precision at the top K codes ($P@K$), or the number of relevant codes in the K first predicted code, is a complete indicator to quantify the number of relevant codes. $P@K$ is defined in Equation 3.6, where r is a binary array and i indicates the presence or absence of the i suggested code in the gold standard.

$$P@K = \sum_{i=1}^{K} \frac{r(i)}{K} \tag{3.6}$$

Although the Precision estimation is usually complemented by the Recall and F scores to quantify the correlation between annotated and predicted codes, this is not necessary when fixing the number of predicted codes. Instead, other scores are more significant in this context, such as the Discounted Cumulative Gain (DCG) at K ($DCG@K$) and the normalized Discounted Cumulative Gain (nDCG) at K ($nDCG@K$). Both ($DCG@K$) and ($nDCG@K$) measure the distribution of those relevant codes giving more importance to the top positions. Equations 3.7 and 3.9 describe the metrics $DCG@K$ and $nDCG@K$, where r is the same binary array and $|REL|$ is the number of relevant codes up to position K.

$$DCG@K = \sum_{i=1}^{K} \frac{r(i)}{log_2(i+1)} \tag{3.7}$$

$$IDCG@K = \sum_{i=1}^{|REL|} \frac{r(i)}{log_2(i+1)} \tag{3.8}$$

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{3.9}$$

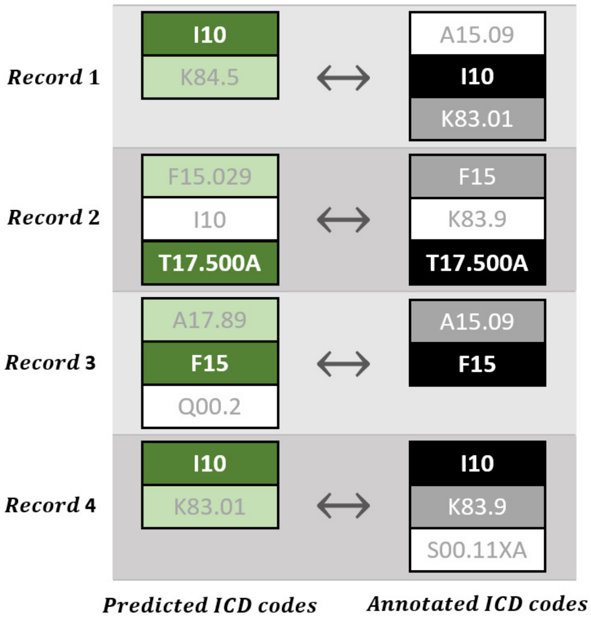
If we take as an example the records and predictions from Figure 3.21, we could measure the Precision, Discounted Cumulative Gain, and normalized Discounted Cumulative Gain at the first two codes ($K = 2$). $P@2$ would be 0.37 as only 3 TP out of 8 predicted codes would be considered. In turn, $DCG@2$ is computed as 1.0, 0.0, 1.58, and 1.0 values for records 1 to 4. $IDCG@2$ is the same calculation but considering all codes as relevant, so the value for each document would be 2.58. Therefore, the global $nDCG@2$ score would be $\frac{1.0+0.0+1.58+1.0}{2.58 \cdot 4}$, i.e 0.35.

Partial matches

ICD hierarchy introduces dissimilarities between codes, resulting in differences between levels or parent dependencies. The classification is designed so that codes that share the same parents share common features. Thus, it is not the same to recommend a code from the same group, close in the hierarchy tree, than a totally different code. Besides, often the same diagnosis is slightly nuanced and mapped into different codes. It is therefore recommended the introduction of distances in the estimation of failures and successes.

The hierarchy of the ICD is reflected in the nomenclature of the code identifiers themselves, whereby the more characters two codes share, ordered from left to right, the closer they are in the structure. Figure 3.22 shows an example of evaluation considering the partial matches of the codes. These partial matches are shown in light green in the predictions and in grey in the gold standard.

Figure 3.22: *Example of partial matches between the predicted and gold standard subsets of codes for a set of records. Exact matching codes have been highlighted in dark green for the predictions and in black for the gold standard. Partial matching codes have been coloured in light green for the predictions and in grey for the gold standard. Finally, unmatched codes are blank.*



The idea of measuring the similarity of codes in terms of the number of characters they share lies in the concept of Information Content (IC). The more characters identifying a diagnosis, the deeper it is in the hierarchical tree, and therefore, the greater the specificity and amount of information it contains. Following this line of thought, a modification of the above metrics has been proposed.

Similarity values between pairs of codes are calculated exploiting the hierarchical structure as proposed in Jia et al. (2019). Equation 3.10 deals with the IC of the code 1 ($IC(i)$), code 2 ($IC(j)$), and LCS ($IC(LCS(i, j))$). The IC has been established as the number of characters, considering that the size of the final CIE-10-ES codes can range from 3 to 7 characters ($IC \in [3, 7]$). As can be verified, similarity based on the information content of common and individual codes preserves coherence, as two codes depending on the same parent are closer the deeper they are, i.e., the more specificity, the smaller the difference.

$$C(i, j) = \frac{2 \cdot IC(LCS(i, j))}{IC(i) + IC(j)} \tag{3.10}$$

Then, the amount of similarity shared by two sets of codes is defined by the Similarity Positive (SP) (Equation 3.11). SP would reflect the fraction of predictions that matched. It can be calculated as the maximum weight matching in a bipartite graph $G = (V, E)$, where the vertices are the union of two subsets $V = V_1 \cup V_2$. V_1 are the predicted codes, V_2 are the gold standard codes, and E represent the edges between both subsets, which have a cost based on the code similarity $C_{i,j}$. Such maximization is defined in Equation 3.11, where N_{rel} is the number of predicted codes, N_{true} is the number of codes in the gold standard, and $X_{i,j}$ is a binary value indicating the assignment of code i to code j. As a constraint, there must be only one positive value of X for each i. The Hungarian method proposed by Munkres (1957) has been used for the optimization.

$$SP = max \sum_{i=1}^{N_{rel}} \sum_{j=1}^{N_{true}} C_{i,j} X_{i,j} \tag{3.11}$$

Dissimilarity is not symmetric, and therefore depends on directionality. Thus we can distinguish between Positive Dissimilarity ($\hat{S}P$) and Negative Dissimilarity ($\hat{S}N$). $\hat{S}P$ is the predictions-goldstandard dissimilarity and can be defined as the fraction of unmatched predictions, while $\hat{S}N$ involves the goldstandard-predictions dissimilarity and is the fraction of unpredicted codes. Both are defined in Equations 3.12 and 3.13, where N_{rel} and N_{true} are the number of predicted and gold standard codes again.

$$\hat{S}P = N_{rel} - SP \tag{3.12}$$

$$\hat{S}N = N_{true} - SP \tag{3.13}$$

New metrics based on those similarity values between the predicted and gold standard code sets are explored, such as the Similarity Precision (P_S), Similarity Recall (R_S), and Similarity F-Score (F_S). P_S has been published in Almagro et al. (2020). Equations 3.14, 3.15, and 3.16 provide the respective definitions. P_S is the predicted hit rate, R_S is the estimated hit rate, and F_S is still the harmonic mean between predicted and estimated hit rates. These scores can be directly adapted to the ranking output by only substituting the variable N_{rel} by K. P_S is the same as P when there are no partial similarities because SP would be the sum of the cost functions of the code pairs that match exactly (i.e., TP), and therefore, $\hat{S}P$ would be FP and $\hat{S}N$ would be FN. Thus, the difference $P - P_S$ provides an idea of the percentage of partial overlap of code, excluding exact code matches.

$$P_S = \frac{SP}{SP + \hat{S}P} \tag{3.14}$$

$$R_S = \frac{SP}{SP + \hat{S}N} \tag{3.15}$$

$$F_S = (1 + \beta^2) \cdot \frac{P_S \cdot R_S}{\beta^2 \cdot P_S + R_S} \tag{3.16}$$

Following the example in Figure 3, codes *F15*, *I10*, and *T17.500A* would have an exact match, so their similarity would be 1. In contrast, codes *F15.029* and *K83.01* would partially match codes *F15* and *K83.9* respectively, both with a similarity of 0.67. The rest of the codes would not have a partial match as the requirement of a minimum of three characters in common is not satisfied. Therefore, SP would be $1.0 + 0.0 + 0.67 + 0.0 + 1.0 + 0.0 + 1.0 + 0.0 + 1.0 + 0.67$, i.e., 5.33, while $\hat{S}P$ and $\hat{S}N$ would be 4.67 and 5.67. Finally, P_S would be 0.53; R_S, 0.48; and F_S, 0.51. Repeating the same example with $K = 2$, P_S, R_S, and F_S would yield values 0.43, 0.39, and 0.41.

Propensity

As mentioned repeatedly, there are codes more frequent than others, or with more abstract descriptions, which causes inequalities in the coding time. As it is not easy to measure the complexity of descriptions automatically, we have focused on the frequency of the annotations. We have assumed that given the volume of possibilities and the need to constantly consult manuals, those diagnoses to which the coders are familiar with will be easier to remember and code, while those diagnoses with only a couple of annotations will require more regular consultations. This leads to the introduction of Propensity score matches.

The Propensity (p_c) represents the marginal probability for a relevant code (c) to be found in a particular record, so that the higher the frequency, the higher the Propensity. We have relied on the work of Jain, Prabhu, and Varma (2016) to define the Propensity of each code. Authors have modelled the Propensity as a sigmoidal function of $logN_c$ in Equations 3.17 and 3.18, where N is the size of the gold standard, N_c is the number of records annotated with the code c in the gold standard, and A and B are configurable parameters in the order of 10^{-1} (we will typically use 0.55 and 1.5, as recommended in the paper).

$$C = (logN - 1)(B + 1)^A \tag{3.17}$$

$$p_c \equiv P(y_c = 1|y_c^{\cdot} = 1) = \frac{1}{1 + Ce^{-Alog(N_c+B)}} \tag{3.18}$$

For example, the Propensity associated with a code appearing in 1,000 of 3,500 records would be $p_c = 1,000/3,500$, i.e 0.28. For this reason, we have transformed the propensity into a distance that works as a corrective factor to reduce the weight of very frequent codes. Hence, we have applied the conversion $1 - p_c$.

We have defined the Propensity Scored True Positives ($PSTP$), Propensity Scored False Positives ($PSFP$), and Propensity Scored False Negatives ($PSFN$) in Equations 3.19, 3.20, and 3.21. $PSTP$ is the amount of information provided by the matched predictions, $PSFP$ is the amount of information not provided by the matched predictions, and $PSFN$ is the amount of missing information. Based on such values, the Propensity Scored Precision (PSP), Propensity Scored Recall (PSR) and Propensity Scored F-Score (PSF) have been proposed in Equations 3.22, 3.23, and 3.24.

$$PSTP = TP \cdot (1 - p_c) \tag{3.19}$$

$$PSFP = FP \cdot (1 - p_c) \tag{3.20}$$

$$PSFN = FN \cdot (1 - p_c) \tag{3.21}$$

$$PSP = \frac{PSTP}{PSTP + PSFP} \tag{3.22}$$

$$PSR = \frac{PSTP}{PSTP + PSFN} \tag{3.23}$$

$$PSF = (1 + \beta^2) \cdot \frac{PSP \cdot PSR}{\beta^2 \cdot PSP + PSR} \tag{3.24}$$

If we use the Propensity scores from Table 3.7 and the matches from Figure 3.21, $PSTP$ for I10, F15, and T17.500A codes would be $2 \cdot 0.65$, $1 \cdot 0.88$, and $1 \cdot 1$, i.e., 1.3, 0.88, and 1 respectively (3.18 overall). In turn, $PSFP$ would be $1 \cdot 0.95 + 1 \cdot 1 + 1 \cdot 0.65 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1$, i.e., 5.6, and $PSFN$ would be $2 \cdot 1 + 1 \cdot 1 + 1 \cdot 0.88 + 2 \cdot 0.92 + 1 \cdot 1$, i.e., 6.72. Hence, PSP, PSR, and PSF would yield values 0.36, 0.32, and 0.34 respectively. As can be seen, these values reflect worse performance compared to the example in Figure 3.21, as the impact of frequent code hits, such as I10, is reduced in Precision while rare unrecovered codes aggravate Recall.

	I10	F15	K83.9	K84.5	Q00.2	K83.01	A15.09	A17.89	F15.029	T17.500A	S00.11XA
Frequency	350	120	80	50	20	4	3	2	1	1	1
Propensity	0.65	0.88	0.92	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 3.7: *Example of propensity scores calculated for each code from the example in Figure 3.20 for a collection of 1,000 documents.*

3.4.2 Baseline

Reference scores are necessary to relativise the results. For example, getting 90% correct on the task of binary word domain classification, medical or general domain, may be an acceptable result if we are testing a large volume of words from both classes, but it may be a non-significant result if 95% of the tested words fall into the general domain. In particular, the presence of extreme imbalances makes baseline design important.

Baselines consist of simple heuristics or simple, trivial solutions aimed at achieving the minimum scores to be exceeded. We proposed the heuristic of assigning the N most frequent codes to each record, where N is the average number of codes per record in each collection. This baseline has been evaluated on the corpora described in Section 3.2 with all the metrics described in Section 3.4.1. Reference scores are shown in Table 3.8. N is 4 for the *Causes of Death* corpus, 14 for the *CodiEsp* corpus, and 10 for the *HUFA* corpus. In this case, the best K value would be N so any score calculated at K will be the same as the one estimated for all predictions.

In terms of micro values, P shows the percentage of matching codes in the prediction, which is composed of the most frequent codes in this case. In turn, PS estimates the percentage of partial matches in the predictions, while PSP indicates the percentage of matches by assigning higher weights to minority codes. In addition, PSP_S focuses on the partial match by promoting also the percentage of matches corresponding to the minority codes. For example, the hit rate varies in the FR subset from the *Causes of Death* corpus as follows: 9.41%, 16.10%, 9.08%, and 15.94% for exact, partial, weighted exact, and weighted partial matches in the predictions respectively.

In contrast, R estimates the percentage of retrieved codes among the annotations. Its corresponding R_S, PSR, and PSR_S values focus on the partial matches, weighted exact matches, and weighted partial matches from among all annotations. The CodiEsp corpus shows the percentages 13.76%, 20.94%, 8.50%, and 13.42% for the above-mentioned values. Macro-averaged values show the weighted average of the above values for each code, thus providing an idea of the distribution of these scores around frequent and infrequent codes. Since the number of codes is often large and the matches accumulate for the most frequent codes, the averages tend to produce comparatively low values, e.g., the micro-averaged precision in the HUFA corpus is

Score		***Causes of Death* corpus**				***CodiEsp* corpus**	***HUFA* corpus**
		FR	HU	IT	All		
Micro	P	9.41	26.28	16.37	8.96	14.51	14.63
	R	9.16	22.82	15.24	8.12	13.76	14.46
	F	9.28	24.43	15.78	8.51	14.13	14.55
	P_S	16.10	35.17	25.73	21.49	22.26	22.54
	R_S	15.68	30.54	23.96	19.47	20.94	22.28
	F_S	15.89	32.70	24.82	20.43	21.58	22.41
	PSP	9.08	25.64	15.14	6.75	14.31	13.30
	PSR	3.04	6.42	6.47	2.94	8.50	4.46
	PSF	4.55	10.26	9.07	4.09	10.66	6.68
	PSP_S	15.94	34.49	24.72	20.35	22.08	21.12
	PSR_S	5.56	9.06	11.08	9.11	13.42	7.51
	PSF_S	8.24	14.36	15.30	12.58	16.69	11.08
Macro	P	0.02	0.05	0.07	0.01	0.14	0.02
	R	0.22	0.20	0.44	0.14	0.92	0.19
	F	0.04	0.08	0.12	0.02	0.24	0.05
	P_S	0.03	0.07	0.11	0.03	0.21	0.04
	R_S	0.22	0.20	0.44	0.14	0.92	0.19
	F_S	0.06	0.10	0.18	0.05	0.34	0.07
	PSP	0.00	0.00	0.02	0.00	0.07	0.00
	PSR	0.03	0.02	0.10	0.02	0.50	0.04
	PSF	0.00	0.01	0.03	0.00	0.12	0.01
	PSP_S	0.00	0.01	0.02	0.00	0.11	0.01
	PSR_S	0.03	0.02	0.10	0.02	0.50	0.04
	PSF_S	0.01	0.01	0.04	0.01	0.18	0.01
Ordering	$nDCG$	26.70	57.45	44.08	22.56	49.18	48.13
	$nDCG_S$	53.05	72.51	64.60	61.13	58.45	62.42
	$PSnDCG$	26.62	56.86	43.76	22.63	48.03	45.56
	$PSnDCG_S$	52.47	71.39	63.63	63.21	56.27	58.22

Table 3.8: *Evaluation of the baseline on collections with different metrics.*

0.2%. Last but not least, the ordering values quantify the distribution of matches in the output rankings, i.e., higher values indicate that matches are placed in the top positions.

Differences in code distributions can be appreciated by examining the scores yielded by the baselines. The high values of the micro-averaged scores indicate a greater concentration frequent codes in the instances, while the low values of the macro-averaged scores show a greater dispersion of minority codes in the instances. Thus, the subset HU shows a higher presence of the most frequent codes (around 26% P) than the other subsets of the corpus *Causes of Death*. In turn, the CodiEsp and

HUFA corpora show the same concentration of frequent codes but are different at low frequencies. The difference of 0.14 over 0.02 in the macro-averaged P indicates that the records in the CodiEsp corpus are associated with a lower diversity of minority codes.

3.5 Discussion and concluding remarks

As introduced in the chapter, the elements necessary for proper ICD-10 coding experimentation have been addressed: the raw input data to be coded (consisting of the records and annotations), NLP pre-processing to prepare the data format, and evaluation function dealing with the hierarchy and distribution of the codes.

Data We have described all data collections used in this thesis in Section 3.2. Table 3.9 summarises the main features of each corpus. The degree of difficulty in predicting ICD-10 codes varies greatly depending on the type of record and the requirements surrounding coding. In particular, although the three collections contain ICD-10 coded records, the coding task faces different constraints.

Hence, although the *Causes of Death* corpus contains records in different languages, the lexical diversity rate is scarce. Regarding the criteria, the certificates has been coded with the international version of the ICD-10, which limits the number of codes to only 4 digits, excluding procedures. Moreover, death certificates are not verboses but consist of a few lines, so no sophisticated techniques are required to identify relevant information.

In contrast, the Clinical Case Coding in Spanish Shared Task (CodiEsp) corpus includes clinical case studies, which are longer documents with greater lexical diversity. This collection has been annotated with the CIE-10-ES non-final codes, so there are over 98,000 potential diagnoses and 87,000 possible procedures. Coding has been done specifically for the task with some simplifications, assigning all codes that have a possible relationship to textual evidence. In this way, organisers do not regard negations or medical suspicions, nor do they apply the ICD rules for exclusions or inclusions.

Finally, HUFA corpus consists of complete hospital records coded by CIE-10-ES specialist, so there are no lexical or size restrictions. Coding has not been performed specifically for the creation of this collection, but these are real examples of the task. Thus, CIE-10-ES coding has been applied considering the contexts and rules, with over 71,000 final diagnoses and 75,000 final procedures.

All three corpora have been useful to this research for some experimental purpose; however, the main corpus is the set of EHRs from HUFA. It is the only corpus with data collected in hospitals rather than designed for a particular task, in addition to being the largest in terms of volume (see the digital volume in Table 3.9).

	Languages	Record size	Lexical diversity rate	Coding level	ICD criteria	Code number	Coverage	Digital volume
Causes of Death	FR,HU,IT	Few lines	0.2	Line	Entity detection	14,000	35%	41 MB
CodiEsp	ES	Few paragraphs	22.4	Sentence	Entity detection	185,000	2%	9 MB
HUFA	ES	Many paragraphs	1,076.3	Document	Reasoning + rules	147,000	9%	1,243 MB

Table 3.9: *Comparison of features between collections.*

Pre-processes All these collections represent more than one gigabyte of unstructured textual information, in free format, which is not easily processable. For this reason, different pre-processes have been described which will be decisive to achieve quality data representations that improve the performance of the proposals. In particular, we have implemented an anonymisation process based on regular expressions and gazetteers; a tokeniser also based on rules and adapted to clinical text; and a lexical normalisation process with our own tools adapted to the domain.

Evaluation The annotation criteria described for each corpus offer an idea of how the coding function to be modelled works, which is essential for choosing the appropriate metric.

Some of the experiments using the corpora from the shared tasks have been evaluated with P, R, and $F - Score$ in order to compare the results with those of the rest of the participants. Although we consider that such metrics are not the best options to quantify the similarity between predictions and the gold standard, organizers have established them as the official metrics of the competitions.

We have focused on the hierarchical structure and the non-uniformity of the code set, as they characterise the challenge of ICD-10 coding. Hence, new evaluation metrics have been proposed to quantify successes and failures by estimating the distances of the codes within the ICD hierarchical structure and the amount of information provided by the predictions to the annotators. In addition, ranking-based metrics contextualised to the XMTC such as $P@K$ and $nDCG@K$ have been used, which only deal with the first few codes. Several K values can be computed to evaluate different ranges, but all decisions made should be based on the average number of codes per document as a general rule, e.g., the average number of codes per record is 10 in the HUFA corpus. So it makes sense in this case to evaluate approaches on the ability to correctly recommend the top 10 codes.

CHAPTER

4

NOT ENOUGH EXAMPLES? EXPLORING UNSUPERVISED APPROACHES

Content

This chapter presents unsupervised ICD coding approaches to address the non-representability of most codes when dealing with examples. The idea is to exploit knowledge-based representation to ensure the inference of codes missing in the available records, which conventional supervised approaches fail to reach. To this end, the coding challenge has been tackled as an Information Retrieval (IR) rather than a classification task, so that alternative knowledge-based representations of codes have been explored for providing EHRs matches.

The following objectives have been pursued:

- Generate quality representations for ICD codes.
- Explore IR methods based on lexical matching.
- Explore semantic-based IR methods by introducing concept relationships in similarity estimation.
- Compare the performance of IR proposals with supervised baselines.

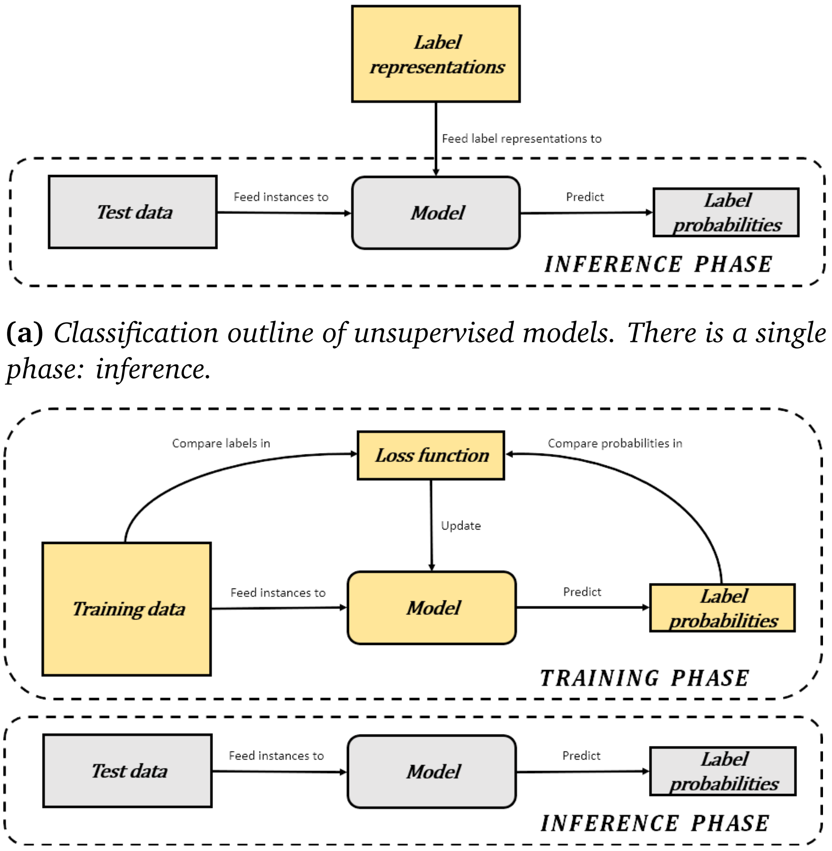
4.1 Introduction

The current trend in computer science in solving complex tasks is the use of supervised learning methods by exploiting data containing examples to model an objective function by means of pattern learning, an area that encompasses Machine Learning (ML). However, as discussed in Section 1.2.2, the comprehensiveness of the ICD entails a huge number of diagnostic heuristics that are not often reflected in patient samples. The high specificity results in an increased likelihood of assigning very diverse codes, often not previously coded in a particular health institution.

Given the amount of possible codes compared to the volume of admissions and the huge differences in prevalence, a large percentage of diagnoses and procedures are underrepresented, or even non-represented, in the generated EHRs collections. For example, since the implementation of the CIE-10-ES in Spain in 2016, an average of 1,000 new codes per year have been registered in the hospital discharge section of HUFA (2017 and 2018 estimates), almost one-fifth of the different codes collected per year. This rate is not sufficient to produce a training data set representative for most codes before the release of the next, broader, and more specific ICD revision.

Figure 4.1 shows the outlines of the main classification methods we have explored: unsupervised methods that directly exploit label representations to establish similarities with instances, and supervised methods that exploit labelled data to capture patterns between instances and labels. As mentioned in Section 1.2.2, the absence or scarcity of examples for a large part of the nomenclature prevents the use of discriminative learning to identify the corresponding patterns, so it would be desirable to explore alternative representations based on expert knowledge.

Figure 4.1: *Overview of the two main types of classification explored in this research.*



(a) *Classification outline of unsupervised models. There is a single phase: inference.*

(b) *Classification outline of supervised models. There are two phases: training and inference.*

In this chapter, we have explored different unsupervised methods by approaching coding as an IR task while comparing them with supervised baselines in order to answer the following Research Question: "*Is it possible to approach ICD-10 coding using unsupervised techniques in a way that can be a competitive alternative to supervised methods?*" (**RQ 2**). The idea is to analyse the performance of such proposals and examine the potential benefits in comparison with supervised approaches. For this purpose, the evaluation has been conducted on the multiple data collections presented in Chapter 3, analysing the differing response of the methods according to the complexity of the task. Although proposals on all three collections have been evaluated, the most relevant results are those for HUFA corpus as the aim of the thesis is to explore possible improvements on real corpora, as mentioned in Section 1.4. Overall, EHRs are addressed as sets of diagnostic evidence in which those relevant to the corresponding ICD codes should be identified. The proposals relies on two foundations, detailed as follows:

- Lexical similarity. The set of descriptions and terminology from the ICD has been established as the relevant information, so that matching clinical expressions increase the probability of assignment of the corresponding code. Additionally, the enrichment of the representations by terminology extraction from another set of EHRs has been explored.
- Semantic similarity. Due to variations in granularity within the ICD, lexical matching limits coverage considerably. An approach that exploits the "Is A" hierarchical tree from SNOMED CT in order to identify more general concepts through specific ones has been explored.

4.2 Related Works

This section aims to further examine unsupervised classification methods in the clinical domain and to contextualise the unsupervised approaches for ICD coding described in Chapter 2, so that the reader is provided with an overview of the purpose of the approaches described.

4.2.1 Introduction

Clinical Information Extraction is one of the most demanded tasks given the need to structure the information within the EHRs. Today, there are numerous concept classifications for EHR assistance purposes, such as SNOMED CT, MeSH, UMLS, ICD and International Classification of Functioning, Disability and Health (ICF) among others. As mentioned above, the availability of clinical data is limited due to the high sensitivity involved (Moen et al., 2015), so that many authors have therefore preferred to explore unsupervised approaches. Thus, instead of learning to identify labels from examples, the mapping between features and labels is fixed using some knowledge base. Labels have been characterised by representations using external knowledge with the aim of establishing direct mappings for features. The association of features to labels can be done by means of structured knowledge such as definitions and terms from ontologies, or via distributed knowledge captured in large corpora.

As reported by Stanfill et al. (2010), clinical coding has been addressed mainly with NLP and Information Retrieval (IR) approaches. Research has focused on NLP techniques to deal with the semantics of specific concepts, with approaches focused on entity detection and normalisation. Different authors have released a variety of Named Entity Recognition (NER) tools for the detection and identification of clinical concepts, e.g., MedLEE (Friedman et al., 2004) and MetaMap (Aronson, 2001) use parsers for coding UMLS concepts. Other elaborate proposals such as cTAKES (Savova et al., 2010) involve POS Tagging, parser, NER, and negation detection processes. In this line, Xu et al. (2010) explore MedEx, a clinical tool based on semantic taggers and parsers. Another approaches such as the one proposed by Liu et al. (2013) have focused on normalisation and a combination of regular expressions and match rules.

Given the limited coverage (Pradhan et al., 2014) and portability issues involved (Carroll et al., 2012), IR systems, often supported by NLP techniques, have been explored for more abstract concepts. The IR methods in dealing with EHRs try to identify the most relevant entries (ICD codes) for satisfying an information request or query (the text extracted from EHRs). Some of the most notable approaches in the domain include EMERSE (Hanauer et al., 2015), Léon Bérard Cancer Center System (Biron et al., 2014), StarTracker (Gregg et al., 2003), and EliIE (Kang et al., 2017).

As for the EHR categorisation, the concepts are typically indexed as the entries to be retrieved and the records are transformed into queries. Both concepts and documents are represented with the same features by vectorising text so that the similarity between the two sets can be estimated. Vectorisation is usually done on the assumption of independence between words, i.e., Bag-of-Words. The complexity of EHRs requires more complex IR systems to deal with highly domain-specific terminology, processing morphology such as inflection and derivation, synonyms, and homographs. In addition to NLP pre-processing, it is common to use ontologies and implicit domain knowledge for query expansion (Zhu et al., 2013). Other authors have relied on the use of word embeddings to tackle lexical diversity (Banerjee et al., 2015).

SOTA research on unsupervised categorisation of EHRs, such as the one proposed by Wang et al. (2019), recognises three main types of approaches according to the representations and the similarity method: geometric, probabilistic and semantic methods.

Geometric approximations rely on measuring the lexical overlap between documents and labels by comparing sparse vectors projected onto a Vector Space Model (VSM). In contrast, probabilistic approximations are based on calculating the probabilities of each label given the terms in the document. Finally, semantic approaches exploit relations between meanings to measure the similarity between labels and documents, either via ontology or distributed semantics. Each one is described in more detail below.

4.2.2 Geometric IR methods

Geometric techniques are based on lexical sparse vectors, which are based on the Independence Principle. Hence, records are represented as n-dimensional vectors, with each dimension corresponding to an individual term. Relationships between terms are not implicitly included. One of the most popular proposals is the association of SNOMED CT entities by means of similarities in a VSM based on Term Frequency-

Inverse Document Frequency (TF-IDF) (Ruch et al., 2008a; Yu, Berry, and Bisbal, 2011). The creation of the TF-IDF space around concepts rather than terms has also been explored to deal with granularity mismatches (Aleksovski and Sevenster, 2010; Koopman et al., 2012a,b; Zuccon et al., 2012). Exploiting both approaches, Liu et al. (2019a) propose Cohort Retrieval Enhanced by Analysis of Text from Electronic Health Records (CREATE). This system simultaneously performs queries on structured and unstructured data, extending coverage with such a combination. For this purpose, two indexes are built, one consisting of the conceptual definitions from clinical ontologies such as ICD, UMLS, and SNOMED CT, and another one composed of the EHR texts to which we associate concepts identified using cTAKES. In this way, the relevance score of the concepts given a query is calculated as the sum of the scores returned by both indices.

4.2.3 Probabilistic IR methods

The probabilistic methods consist of estimating the probability of association of each term to the labels. For example, Wang et al. (2009) apply χ^2 test to associate adverse events to identified drugs and Banerjee et al. (2015) predict adverse drug events by means of co-occurrence statistics, using ontologies and word embeddings for query expansion. Zhu et al. (2013) compare probabilistic-based systems, such as Dirichlet LM and Markov Random Field (MRF), with a geometric one involving textual and conceptual spaces. Authors use MeSH to perform a query expansion and MedTagger as a NER for identifying the concepts. Mirhosseini et al. (2014) also explore multiple approaches (TF-IDF-based VSM, BM25 (Jones, Walker, and Robertson, 2000), and LMs) for identifying SNOMED CT entities, achieving the highest results with the VSM. In a similar way, Wang et al. (2019) evaluate multiple unsupervised methods on a released clinical test collections for categorising EHRs. The authors compare TF-IDF-based VSM, BM25, Dirichlet LM, MRF, and CREATE. Again, TF-IDF-based VSM methods provide the best results, with a remarkable difference.

4.2.4 Semantic IR methods

The semantic methods usually rely on the relationships between entities within an ontology or context dense vectors such as word embeddings. For example, Albitar, Fournier, and Espinasse (2014) exploit the hierarchical tree of UMLS to calculate the similarity between documents according to the similarity between concepts. The similarity between concepts is estimated by means of different ontology-based measures, such as wup (Wu and Palmer, 1994) and lch (Leacock and Chodorow, 1998). Instead, Moen et al. (2015) explore TF-IDF-based and distributional semantic-based VSM, varying the reference vocabulary. Authors use the weighted sum of vectors at the level of clinical note and patient episode. In turn, Das et al. (2020) explore an IR method on clusters of related clinic papers using BERT-based representations. Unsupervised similarity with word embeddings has proven to be effective for short texts. In contrast, vector weighted averaging is no longer effective for long texts as task-relevant information is blurred with the more abundant non-relevant information (Wei and Eickhoff, 2018).

4.2.5 Discussion

VSM methods yield positive results in the unsupervised approach SOTA for document classification, so we have focused on this type of models. The main idea is to transform both records and labels into vectors to estimate the similarity between them using some function, tipically some geometric operation such as cosine. Such vectors can be based on lexical information such as TF-IDF representations or on semantic information such as concepts from an ontology.

Most unsupervised proposals for ICD coding rely on statistical information and label representations composed of sets of expressions manually detected in the records, such as the approaches proposed by Arifoğlu et al. (2014), Park et al. (2019), and Pérez et al. (2018). In this line, Rizzo et al. (2015) evaluate VSM, BM25 and LM on the ICD coding task, while Goldstein, Arzumtsyan, and Uzuner (2007) compare a TF-IDF-based VSM with a supervised model. The tendency of these proposals is to only show global metrics, without providing a deeper analysis of the results. Therefore, it is proposed to analyse in more detail the possible contributions of VSM methods in an extreme ICD distribution.

As for the semantic proposals, SOTA proposals focus on a semantic representation based on dimensionality reduction, either through RI (Henriksson, Hassel, and Kvist, 2011), LSA (Jatunarapit, Piromsopa, and Charoeanlap, 2016), or word embeddings (Moen et al., 2015). The length of the documents hampers unsupervised embedding-based methods, so we have decided to explore similarity-based methods between clinical concepts. Instead of using exact matches, hierarchical relationships have been exploited to deal with differences in granularity. In contrast to the approach suggested by Albitar, Fournier, and Espinasse (2014), we have developed a method of comparison between sets of concepts that considers all similarities, not just maximum values.

4.3 Lexical similarity

One of the most elementary approaches in ICD coding is the direct search of the terminology defined in the standard. In fact, there are multiple look-up software that rely on search engines to retrieve the ICD descriptions. Nevertheless, the simplicity of the method precludes implementing an NLP approach as it would result in almost zero coverage given the huge lexical variability of the domain that the nomenclature does not capture, as pointed out in Section 1.2.2.

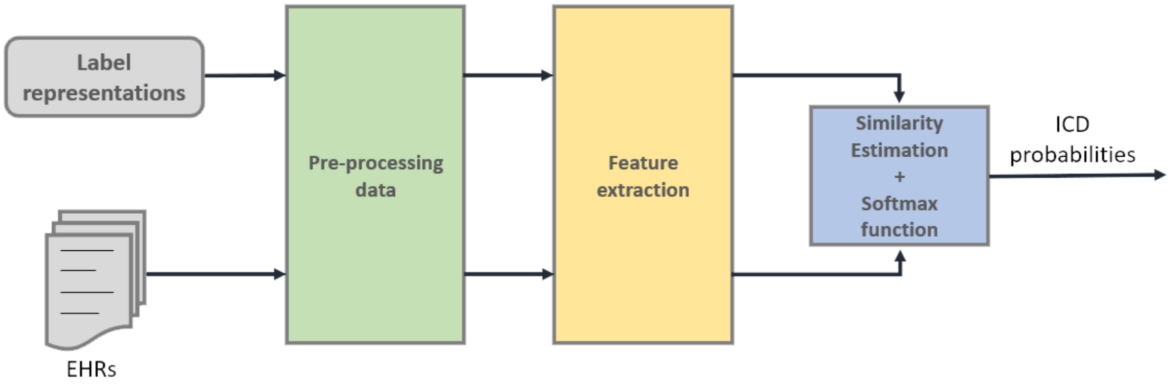
This section explores the widely extended IR approach based on transforming documents and labels into vectors of the same Euclidean space, so that similarity can be estimated geometrically. In particular, it is proposed to project both instances and codes into a lexical space, with as many dimensions as terms in the vocabulary. The aim is to compare the performance of evidence-based IR methods with supervised methods that learn the rules implicit in the encoding.

The unsupervised method used, the supervised models to compare the performance, and the experimentation carried out are described below.

4.3.1 Unsupervised method

The main foundations of the unsupervised approach are code representation, feature selection and similarity calculation. Figure 4.2 shows schematically the inference of the probabilities of the ICD codes for each incoming EHR according to the similarity-based unsupervised proposal. The idea is to represent the codes using descriptions or a set of terms in a similar way to the documents, capturing label and instance features in a uniform way. Thus, codes can be simply ranked with a relevance function to quantify the proximity (i.e., similarity) to the documents, exploiting the homogeneity between labels and EHRs. Ideally, getting code representations as close to the documents as possible maximises the correlation between the estimated similarity and code relevance.

Figure 4.2: *Overview of the unsupervised similarity-based ICD coding approach.*



The components are detailed below.

Label representations

The proposed unsupervised methods rely on code representations to find similarities with instances. One possible scenario would be to use features based on the descriptions and terminology defined in the standard, which would imply a reduced coverage due to the differences in granularity between the semantics of ICD and clinical record languages. In contrast, the explicit use of textual evidence annotated in records would be more in line with the required specificity. The availability of annotated ICD expressions is often associated with the research domain as health professionals tend to code at the document level. For this reason, any technique for extracting ICD terminology from EHRs may also be useful for a more general solution. All these alternatives are detailed below.

ICD terminology The tenth version of the CIE does not have an electronic format that focuses on the digital accessibility of the data. Instead, practically the bulk of the information is distributed in an text document (PDF format) targeted for human readers. This standard comprises two elements with complementary information: Alphabetical Index and Tabular List. The first section contains guided indications of coding by means of the relevant associated terminology. Figure 4.3 shows an example of several entries, where the words "*Abdomen*" and "*Abdominal*" are related to multiple codes, so that may refer to R10.0 or K55.1 depending on the "*acute*" and "*angina*" specifications respectively. As can be seen, different guidelines such as "*see*" and "*see also*" are included for the coders.

Alternatively, Tabular List contains the codes with all the attributes, such as descriptions, inclusions, exclusions, and explanatory notes, among others. In addition to the main descriptions, some codes have additional information. For example, Figure 4.4 illustrates the entry for the code A06, with inclusion and exclusion rules. In turn, "*Acute amebiasis*" and "*Intestinal amebiasis NOS*" are secondary descriptions for the code A06.0. As one may note, Tabular List offers a general definition of the codes, while Alphabetical Index provides greater specificity through precise clinical terms. The Tabular List and Alphabetical Index of the 2016 release of the CIE-10-ES contain about 12,500 and 16,500 different terms respectively, with an approximate overlap of 50%. For example, the code H53.8 is referenced in Tabular List as "*Other visual disorders*", which encompasses all sight-related disorders not specified in the other codes under the same branch. In turn, Alphabetical Index includes specific instances of the code such as "*Visual impairment*", "*Toxic amblyopia*", "*Blurred vision*", "*Polyopia*", and "*Visual disorientation syndrome*".

Despite the complementarity, the main diagnostic and procedural descriptions defined in Tabular List are the only electronic CIE-10-ES information accessible in a structured format. For this reason, we have processed the complete standard in PDF format automatically by extracting the individual information by means of regex and

Figure 4.3: *English example of the content of Alphabetical Index. The Spanish example can be found in Appendix F.*

Aarskog's syndrome Q87.1

Abandonment - *see* Maltreatment

Abasia (-astasia) (hysterical) F44.4

Abderhalden-Kaufmann-Lignac syndrome (cystinosis) E72.04

Abdomen, abdominal - *see also* condition
- acute R10.0
- angina K55.1
- muscle deficiency syndrome Q79.4

Abdominalgia - *see* Pain, abdominal

Abduction contracture, hip or other joint - *see* Contraction, joint

Aberrant (congenital) - *see also* Malposition, congenital
- adrenal gland Q89.1
- artery (peripheral) Q27.8
- - basilar NEC Q28.1
- - cerebral Q28.3
- - coronary Q24.5

...

Figure 4.4: *Example of the content of Tabular List. The original, untranslated example can be found in Appendix G.*

A06 Amebiasis

Includes infection due to Entamoeba histolytica
Excludes1 other protozoal intestinal diseases (A07.-)
Excludes2 acanthamebiasis (B60.1-)
Naegleriasis (B60.2)

A06.0 Acute amebic dysentery
Acute amebiasis
Intestinal amebiasis NOS

A06.1 Chronic intestinal amebiasis

A06.2 Amebic nondysenteric colitis

A06.3 Ameboma of intestine
Ameboma NOS

...

organising it in a structured way. As a result, we have reconstructed Tabular List and Alphabetical Index in plain text in an accessible way[1] to use the terminology tree of Alphabetical Index and secondary descriptions and examples from Tabular List.

[1] Tabular List and Alphabetical Index can be found `https://zenodo.org/record/5148885#.YQQ9Io77SUk`

ICD annotations Annotated ICD evidences are the closest representations to the text within the records to be coded, but the availability of word-level annotations is not always possible, e.g., HUFA coding is at document level so records are not tagged with the exact expressions that coders have relied on to annotate a code. Both the *Causes of Death* and *CodiEsp* corpora (created specifically for research purposes) are annotated at the text fragment level, so that these textual evidences can be used to search for similar expressions in other EHRs corpora. While the use of labelled data to characterise codes is more in the nature of a supervised method, the absence of learning allows for greater coverage.

Annotations provide specific clinical vocabulary for codes with abstract descriptions covering many concepts. For example, the CodiEsp annotations add nearly 5,300 different words to the representations based exclusively on the standard terminology, introducing nearly 2,000 new terms into the vocabulary. A frequent expression of the above code (H53.8) in the records is "*Visual acuity deficit*", which is not covered by the nomenclature.

Terminology extraction Since ICD evidences are not always available, such as HUFA corpus scenario, different terminology extraction methods have been explored to identify the most relevant terms in the coded EHRs. As with the annotation-based representations, these techniques rely on labelled data, so it cannot really be claimed to be an unsupervised method when used on the same corpus.

KLD, χ^2, and $DICE$ are common information Gain-based measures used for computing affinity and expanding queries in IR (Matos-Junior et al., 2012). We have used Kullback-Leibler Divergence (KLD) to identify the terms most closely related to each code using the associated and non-associated documents. In this case, KLD measures the difference between the probability distribution functions of the relevant codes and the entire code set for each term, so the greater the divergence, the greater the term-code correlation. Equation 4.1 describes the distance between the probabilities of relevance $P(t|c)$ and $P(t|c \cup \hat{c})$ of the term t to the code c. $P(t|c)$ is estimated as the frequency of the term t in the documents associated with the code c divided by the total frequency of the term. In turn, $P(t|c \cup \hat{c})$ is computed as the frequency of the term t in all documents in the collection also divided by the overall frequency.

$$KLD(t,c) = P(t|c) \cdot \ln \frac{P(t|c)}{P(t|c \cup \hat{c})} \tag{4.1}$$

Terminology extraction is able to gather acronyms such as "*HTA*" for code I10 (*Essential primary hypertension*), or expressions related to causes and symptoms. For example, the terms identified as relevant for code A02.0 (*Salmonella enteritis*) are "*salmonella*", "*food*", and "*deposition*", which causes the infection, and "*diarrhoea*", "*vomiting*", and "*abdominal pain*", which are symptoms of the disease.

Feature selection

After the corresponding pre-processing described in Section 3.3, a BoW approach has been explored, assuming independence between meanings and dropping syntactic word order. One of the most widespread measures for scoring the relevance of words in IR is TF-IDF. Equation 4.2 describes the TF-IDF value estimation, where $f_{t,d}$ is the frequency of the term t in the document d, T_d is the number of terms in the document d, N is the number of documents, and N_t is the number of documents containing the term t.

$$TF - IDF(t,d) = TF(t,d) \cdot IDF(t) \tag{4.2}$$

$$TF(t,d) = \frac{f_{t,d}}{\max_{i=1}^{T_d} f_{i,d}} \tag{4.3}$$

$$IDF(t) = \log\big(\frac{N}{N_t}\big) \tag{4.4}$$

We have computed the IDF values on the code representations, so that both the normalised frequencies of the code and EHRs terms are weighted with such values. Also, the vocabulary is fixed during the calculation to D unique words, only considering the words from codes. In this way, both codes and documents are represented by sparse vectors of dimension D, with each component being the TF-IDF value of the individual terms.

Similarity estimation

Vocabulary-based high-dimensional vectors constitute a linear space, so it is possible to directly apply geometric measures to quantify the proximity between vectors. Moreover, TF-IDF values measure term significance progressively, creating a continuous space. The TF-IDF-based VSM has repeatedly proved to be one of the most effective techniques in Information Retrieval (IR) (Wang et al., 2019), so we have focused on this method. Ideally, the more terms a code and a document share, the more affinity there is between them, i.e., relevance can be measured as the similarity of their TF-IDF components. Since the Term Frequency (TF) component is the normalised frequency that depends on the length of the text representing the code or document, it is better to compare vectors by orientation or angular rather than spatial distance.

We have used cosine to compute the similarity between codes (v_1) and EHRs (v_2). Cosine similarity is one of the most widespread measures in NLP for the estimation of similarity between feature vectors (see Equation 4.5). Finally, a softmax function is applied to all similarities for estimating code probabilities. Equation 4.6 shows the calculation of the probability of the code i for a dataset with L codes, where s_i is the code similarity value.

$$SIM_{cos} = \frac{v_1 \cdot v_2}{||v_1||\ ||v_2||} \tag{4.5}$$

$$p_i(s_i) = \frac{e^{s_i}}{\sum_{j=1}^{L} e^{s_j}} \tag{4.6}$$

4.3.2 Supervised methods

Two basic supervised classification methods, often used as baseline in advanced systems, have been proposed to compare performance with unsupervised methods. KNN and SVM methods have been implemented based on the same TF-IDF representation for the EHRs.

KNN This classification method assumes that documents with close feature vectors will have similar associated code sets, so the codes for a new document are inferred from the votes of the k nearest neighbours. In this case, the same cosine similarity estimation as in Equation 4.5 has been used. The final set of code probabilities is computed by applying the softmax function (see Equation 4.6) to the code frequency in the selected subset of training documents.

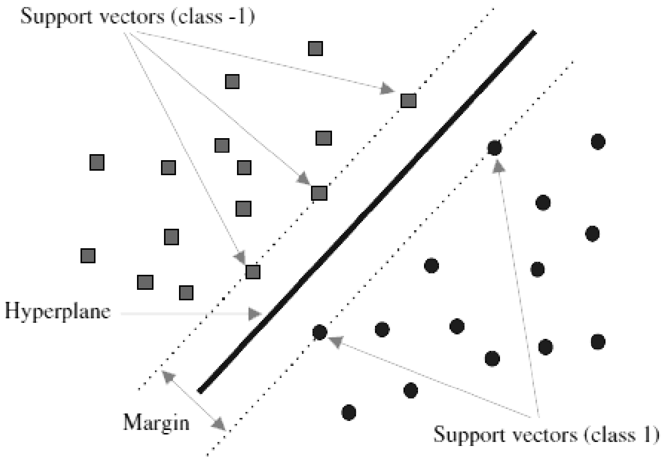
SVM SVM is a ML method based on the optimisation of the hyperplane that linearly separates instances of different classes (see Figure 4.5). In case they are not linearly separable, the algorithm searches for hyperplanes of higher dimensionalities by applying kernel functions. The optimisation is performed by maximising the margin or the distance between the hyperplane and the nearest data points, which ensures better generalisation during learning. Equation 4.7 describes the function that is required to be minimised using L2 optimisation, where n is the number of instances, x_i and $y_i \in (-1, 1)$ are the features and labels of the i-th instance, w is the normal vector to the hyperplane, $\frac{b}{||w||}$ indicates the offset of the hyperplane from the origin, and λ is a parameter limiting the size of the margin. Thus, $w^T x_i - b$ would be the output of the i-th instance. In this case, $y_i = 1$ if the corresponding ICD code is present in the i-th instance, and $y_i = -1$ if it is missing.

$$\left[\frac{1}{n}\sum_{i=1}^{n} max\Big(0, 1 - y_i(w^T x_i - b)\Big)\right] + \lambda||w||^2 \tag{4.7}$$

4.3.3 Experimentation

The performance of the methods proposed above has been evaluated on the test data sets from the three corpora described in Section 3.2. The settings used and the results

Figure 4.5: *Outline of the SVM function.*



obtained are described below.

Experimental settings

The same settings have been used for each corpus, attending to the particularities of each language, with the exception of the unsupervised method based on enriched descriptions. Instead of indexing annotated textual evidence, KLD has been applied to extract terminology from the training dataset of the HUFA corpus. The proposed settings are detailed below:

- The unsupervised method has been applied on each corpus using the terminology contained in the ICD standard as code representation (**IR-T** setting). Annotations made by coders on other documents have also been used to directly characterise codes (**IR-A** setting) for the corpora of *Causes of Death* and *CodiEsp*. The combination of both representations is reflected in **IR-TA** setting. As for the *HUFA* corpus, a code representation based on the extraction of the 30 most related terms estimated by means of KLD (**IR-K** setting) has been used, as well as its combination with the conventional representation (**IR-TK** setting) has been explored. It should be noted that information related to coded examples is being used when introducing code annotations or KLD terms, so these methods are therefore not strictly unsupervised approaches. However, we have retained these proposals in the unsupervised section for clarity, since these theoretically have the ability to infer any code by using information that is not derived from examples. Throughout the results and discussion we have referred to the above settings as unsupervised methods, pointing to this capacity.

- The supervised methods have been separately trained on the three corpora with a single setting. In the case of the SVM (**SVM** setting), the popular Radial Basis Function (RBF) kernel has been used with a coefficient of 2e-7, λ has been set at 1, and the optimisation limit has been fixed at less than 1e-3 improvements in loss. A linear kernel (often recommended for text classification) has not been applied because of the predominant noise in the domain. As for the KNN (**KNN** setting), the 30 closest documents have been used for voting the associated codes.

Results

Table 4.1 collects the micro- and macro-averaged scores for each setting estimated on the *Causes of Death* data sets. As expected, supervised methods achieve significantly higher micro-averaged scores than unsupervised ones on the corpus *Causes of Death* due to the combination of a relatively manageable ratio of examples per class and the short length of the instances (compact expressions). In particular, SVM method seems to identify more codes with F-Score values of 86, 89, and 53% for the IT, HU, and FR data sets respectively, outperforming all other methods. It is followed by KNN with F-Score values that are reduced by 20 to 50% due to more limited learning. However, the difference in the ordering scores is not so high, with the unsupervised method **IR-TA** outperforming the svm in all values for the FR data set. This indicates that the SVM produces rankings with many matches but not in the top positions; in contrast, the unsupervised method produces fewer matches but placed in the top positions.

The difference between micro- and macro-averaged scores is considerably smaller for the unsupervised methods (around an 8% decrease on average in Precision) than for the supervised methods (a drop of 26% and 63% for the SVM and KNN methods respectively) as they retrieve a greater diversity of codes. Official terminology representations yield the poorest performance due to limited lexical overlap with records, which tend to contain less general clinical concepts. In contrast, the annotations (**IR-A**) are more in line with clinical observations. The combination of both representations (**IR-TA**) leads to an improvement of 12% and 9% in micro- and macro-averaged scores respectively, surpassing KNN in F-Score. It is worth noting the poorer Recall of all methods on the FR data set, which may be due to the fact that it involves a set of records with a greater diversity of years. This implies that different versions of the codes are mixed, which suposes a greater inconsistency in the data affecting similarity or learning.

In turn, Table 4.2 shows the results for the CodiEsp and HUFA corpora. Although supervised methods achieve the best performance on the CodiEsp corpus, annotation-

		Causes of Death														
		IT					**HU**					**FR**				
		IR-T	IR-A	IR-TA	SVM	KNN	IR-T	IR-A	IR-TA	SVM	KNN	IR-T	IR-A	IR-TA	SVM	KNN
Micro	P	15.86	24.87	29.58	**96.94**	42.45	31.44	33.09	35.76	**97.39**	78.35	19.64	23.17	28.15	**96.03**	43.92
	R	14.51	22.58	27.19	**77.47**	34.02	25.56	27.71	30.25	**82.33**	65.26	16.93	22.29	26.71	**36.76**	26.72
	F	15.15	23.67	28.33	**86.12**	37.77	28.20	30.16	32.77	**89.23**	71.21	18.19	22.72	27.41	**53.17**	33.22
	P_S	29.90	36.30	41.90	**97.87**	51.33	43.82	49.43	50.79	**99.39**	84.61	31.06	33.10	38.24	**98.28**	60.27
	R_S	27.38	33.07	38.58	**78.27**	41.13	36.11	41.64	43.23	**82.93**	70.49	27.31	31.87	36.35	**37.68**	36.70
	F_S	28.59	34.61	40.17	**86.98**	45.67	39.59	45.20	46.70	**90.42**	76.91	29.07	32.47	37.27	**54.48**	45.62
	PSP	16.05	22.67	29.38	**96.36**	31.77	22.79	23.68	26.37	**96.98**	61.63	11.97	21.45	20.57	**94.54**	37.26
	PSR	10.89	20.26	22.94	**74.20**	28.55	24.17	28.77	31.05	**75.72**	51.53	12.22	17.87	21.26	**32.48**	23.73
	PSF	12.98	21.40	25.76	**83.84**	30.07	23.46	25.98	28.52	**85.04**	56.13	12.09	19.50	20.91	**48.36**	28.99
	PSP_S	29.34	33.44	40.83	**97.52**	38.78	34.40	41.70	42.62	**98.34**	70.67	23.27	31.28	31.09	**96.14**	52.20
	PSR_S	20.56	30.36	32.82	**75.48**	35.47	35.82	46.50	46.66	**77.09**	60.04	24.37	26.17	31.64	**33.49**	33.72
	PSF_S	24.18	31.83	36.39	**85.10**	37.05	35.10	43.97	44.55	**86.43**	64.92	23.81	28.50	31.36	**49.68**	40.97
Macro	P	13.12	24.06	25.75	**72.04**	19.93	24.70	31.26	32.88	**72.76**	22.33	20.37	23.25	26.44	**69.26**	15.97
	R	8.79	20.20	19.18	**62.00**	14.77	22.94	28.60	30.32	**57.31**	18.96	13.70	13.43	18.09	**37.65**	12.47
	F	10.53	21.96	21.99	**66.65**	16.97	23.79	29.87	31.55	**64.12**	20.51	16.38	17.02	21.48	**48.78**	14.00
	P_S	17.00	29.57	29.76	**73.34**	21.27	30.96	37.44	38.60	**75.98**	24.34	25.18	27.32	30.39	**70.90**	18.15
	R_S	11.07	24.93	22.50	**62.72**	15.44	29.31	35.13	36.34	**59.80**	20.02	19.65	16.90	22.68	**38.84**	13.65
	F_S	13.41	27.05	25.63	**67.62**	17.89	30.11	36.25	37.43	**66.93**	21.97	22.07	20.88	25.97	**50.19**	15.58
	PSP	10.78	21.08	21.99	**67.19**	16.69	21.20	27.79	29.09	**68.11**	17.43	15.13	16.81	19.75	**67.38**	12.09
	PSR	7.47	18.47	16.91	**58.64**	13.03	21.06	27.12	28.39	**54.20**	14.89	11.36	10.42	14.65	**36.53**	10.00
	PSF	8.83	19.69	19.12	**62.62**	14.63	21.13	27.45	28.74	**60.36**	16.06	12.98	12.87	16.82	**47.37**	10.95
	PSP_S	13.45	25.56	25.02	**68.41**	18.12	26.53	33.05	33.99	**71.46**	19.09	18.76	19.51	22.63	**68.69**	13.74
	PSR_S	9.03	22.40	19.48	**59.30**	13.56	26.69	32.76	33.61	**56.74**	17.82	16.01	12.85	18.15	**40.96**	10.90
	PSF_S	10.81	23.87	21.90	**63.53**	15.51	26.61	32.91	33.80	**63.25**	18.43	17.28	15.49	20.15	**51.32**	12.16
Ordering	$nDCG$	41.84	61.73	67.55	**96.68**	42.45	73.18	73.04	76.52	**96.21**	78.22	48.45	55.32	**67.26**	52.17	37.92
	$nDCG_S$	52.66	72.95	75.99	**98.48**	51.39	84.33	84.29	86.99	**98.50**	86.80	59.99	65.86	**74.04**	54.47	45.97
	$PSnDCG$	41.70	61.55	67.18	**96.68**	42.45	73.26	72.85	76.38	**98.68**	78.22	48.44	55.30	**67.09**	52.17	37.92
	$PSnDCG_S$	52.28	72.97	75.90	**98.48**	51.37	83.60	84.21	85.23	**97.99**	86.80	59.71	65.41	**73.75**	54.47	45.97

Table 4.1: *Evaluation of the performance of the unsupervised and supervised lexical-based methods for the Causes of Death corpus. The highest values per metric are in bold for each corpus. Similarly, the highest unsupervised values per metric are underlined.*

based similarity techniques yield competitive per-code results in terms of macro-averaged values. The lower ratio of examples per class and the greater length of the documents than in the *Causes of Death* corpus hamper the learning of relevant patterns for a large percentage of codes, so that the distance to the best unsupervised approach is shortened. For example, SVM get more than 100% improvement in micro-averaged F-Score over **IR-TA** but only 15% in macro-averaged F-Score, i.e., an increase in the predictive ability of a small group of codes rather than a generalised gain is produced, so that the overall matches are duplicated but relate to the same codes. Again, the results of KNN are considerably behind those achieved by SVM. As for the differences in the label representation, the difference between the search for similarities based on the official descriptions and the annotations provided by the coders is accentuated in the CodiEsp corpus. In fact, the official terminology seem to provide relatively minor information to the annotations, as one can notice with the small improvement between setting **IR-A** and **IR-TA** (4 and 11% in micro- and macro-averaged scores).

As for the HUFA corpus, supervised methods only beat unsupervised ones in

		CodiEsp					HUFA				
		IR-T	IR-A	IR-TA	SVM	KNN	IR-T	IR-K	IR-TK	SVM	KNN
Micro	P	13.18	22.29	23.41	**56.46**	25.34	5.38	26.91	27.83	**35.25**	29.90
	R	3.01	16.47	16.95	**34.29**	11.10	1.86	19.49	20.43	**30.73**	26.44
	F	4.91	18.95	19.66	**42.67**	15.44	2.77	22.60	23.56	**32.84**	28.06
	P_S	20.55	30.58	31.86	**62.13**	35.21	12.51	35.72	36.69	**41.55**	38.73
	R_S	4.95	23.46	23.92	**39.98**	17.84	4.47	26.41	27.42	**40.16**	36.60
	F_S	7.97	26.55	27.32	**48.65**	23.68	6.58	30.37	31.39	**40.84**	37.63
	PSP	13.26	21.38	22.59	**54.52**	25.88	4.83	27.68	28.59	**30.11**	26.46
	PSR	2.35	13.63	14.11	**29.46**	6.16	2.20	24.99	**26.07**	22.58	18.73
	PSF	3.99	16.65	17.37	**38.25**	9.95	3.02	26.27	**27.27**	25.81	21.93
	PSP_S	20.68	29.58	31.12	**60.41**	35.84	11.92	35.74	36.77	**37.18**	34.00
	PSR_S	3.87	19.83	20.41	**35.10**	10.20	5.58	32.66	**33.79**	31.25	26.29
	PSF_S	6.51	23.74	24.65	**44.40**	15.88	7.60	34.13	**35.22**	33.96	29.65
Macro	P	4.39	11.94	13.15	**15.08**	1.64	2.31	35.69	**36.37**	8.34	6.15
	R	3.81	13.04	14.00	**18.18**	2.24	2.35	40.99	**41.72**	8.95	7.78
	F	4.08	12.47	13.56	**16.49**	1.89	2.33	38.16	**38.86**	8.63	6.87
	P_S	5.66	14.02	15.88	**16.22**	2.22	3.96	39.47	**39.93**	12.20	8.20
	R_S	5.10	15.59	17.09	**19.16**	2.73	4.13	44.32	**44.92**	12.25	9.27
	F_S	5.37	14.76	16.46	**17.57**	2.45	4.04	41.75	**42.28**	12.22	8.70
	PSP	3.56	9.35	10.43	**13.53**	1.14	2.07	37.27	**37.96**	6.60	5.18
	PSR	3.25	10.62	11.53	**16.57**	1.47	2.32	43.89	**44.55**	7.46	6.32
	PSF	3.40	9.95	10.95	**14.90**	1.28	2.19	40.31	**40.99**	7.00	5.69
	PSP_S	4.60	11.06	12.73	**14.58**	1.53	3.56	40.58	**41.09**	9.28	6.94
	PSR_S	4.32	12.73	14.12	**17.47**	1.79	4.05	46.91	**47.46**	9.74	8.16
	PSF_S	4.46	11.84	13.39	**15.89**	1.65	3.79	43.52	**44.04**	9.50	7.50
Ordering	$nDCG$	15.68	55.99	55.45	**82.54**	51.89	11.53	59.60	62.49	**70.39**	61.77
	$nDCG_S$	24.27	66.08	65.36	**85.42**	59.16	19.89	69.24	**73.41**	72.44	68.79
	$PSnDCG$	15.64	55.19	54.60	**77.26**	50.85	11.53	59.90	62.71	**64.43**	58.48
	$PSnDCG_S$	24.01	65.23	64.51	**79.84**	58.99	19.55	69.84	**73.59**	67.73	68.13

Table 4.2: *Evaluation of the performance of the unsupervised and supervised lexical-based methods for the CodiEsp and HUFA corpora. The highest values per metric are in bold for each corpus. Similarly, the highest unsupervised values per metric are underlined.*

exact and partial matches for frequent codes as the scores decrease substantially by assigning greater relevance to minority codes (propensity-scored and macro-averaged metrics). Despite the greater availability of examples, the length of the records (an average length almost three times the average length from the CodiEsp corpus) poses a challenge that may require more complex models. Similarly, the difference between KNN and SVM is smaller than in the other corpora.

The terminology-based setting (**IR-T**) reflects poorer results compared to the other corpora due to the greater difficulty of the task. Conversely, KLD shows the best results in macro-averaged values for all matches. KLD is more effective than SVM because it is not a discriminative learning method requiring many examples, i.e., despite using examples to extract terms, the KLD-based proposal involves statistical techniques that do not demand optimizing parameters, providing more coverage. Finally, in the case of **IR-TK**, the contribution of ICD terminology to the identified expressions is modest.

4.4 Semantic similarity

Although the semantic granularity in the ICD is heterogeneous as it differs from one code to another, the general tendency of the codes is to group a wide variety of clinical concepts. Some of the specific clinical concepts are explicitly in the terminology included in the standard, while others are intrinsically described as part of the meaning of the official descriptions. Such differences in specificity between ICD descriptions and medical observations in records involve less lexical overlap, which complicates the search for similar representations between codes and records. For this reason, it has been necessary to use either the more specific terminology accompanying the codes or the associated expressions from the records to increase coverage.

In the previous section we explored simpler supervised and unsupervised methods with a special focus on multiple representations. Alternatively, in this section we explore a semantic method with the aim of relating general clinical concepts to less abstract ones. For this purpose, we follow the same outline as Figure 4.2, extending the feature selection and similarity method to handle meanings.

Label representations

We have used the code representations described in Section 4.3.1: the official descriptions and terminology provided by the nomenclature, as well as annotations or expressions extracted from the records.

Feature selection

The pre-processing described in Section 3.3 has been applied to code representations and record texts. Instead of using all the information from documents in aggregate by transforming the full text into term frequencies, the creation of context windows to handle local information has been proposed. In this way, documents are processed as sets of fragments of length N words, reducing the jumbling of information caused by an BoW approach.

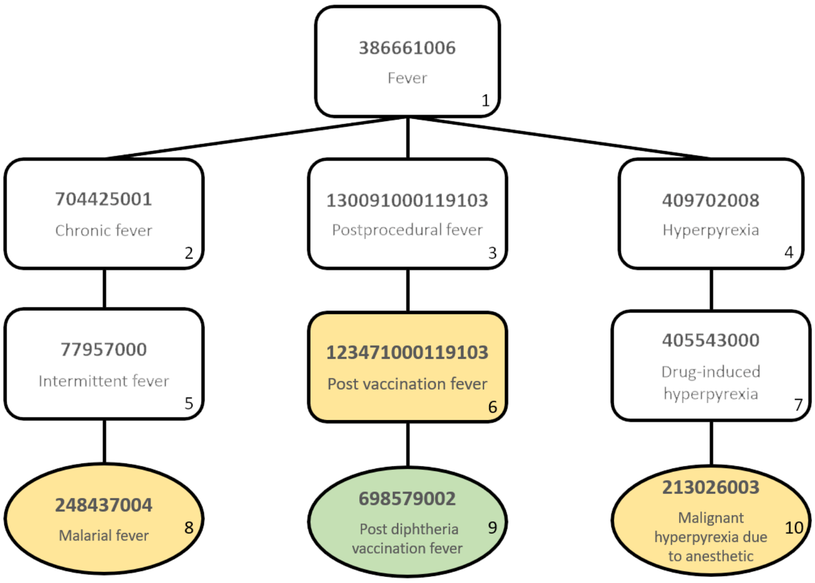
In order to transcend word syntax, we have applied NER to identify the SNOMED CT concepts present in both code representations and record fragments. Since not all relevant words correspond to a clinical concept defined in SNOMED CT, the final features are composed of a frequency vector of recognised entities $\vec{e}$ of size E, where E is the total number of entities, and another frequency vector of words not belonging to any identifiable concept $\vec{v}$ of size V, where V is the total number of words.

Similarity estimation

Although the preprocessing exploits synonymy relationships in SNOMED CT, there is no information on specificity relationships. For example, the concepts 762690000 (*Classical Hodgkin lymphoma*) and 17788007 (*Acute myelocytic leukaemia*) are not interchangeable but both are a type of neoplasm, so they share common features. In this case, both are related in the "Is A" hierarchical tree provided by the SNOMED CT ontology, sharing the same parent: code 400177003 (*Neoplasm and/or hamartoma*). Although SNOMED CT defines other types of relationships, such as "Finding Site" and "Causative agent", the "Is A" tree distributes most of the concepts in a graph while providing a semantic meaning to the distance.

To exploit the SNOMED CT semantic tree, we have proposed to measure the similarity between entity sets by means of a pairwise assignment maximisation problem, being the quantification of the closeness between individual components measured by means of the lin similarity (Lin et al., 1998). lin similarity uses the IC of the common parent and both entities to measure proximity (similar to Equation 3.10), so that the closer the two entities are, the more attributes they share. Equation 4.8 describes the IC-based similarity, where $IC(c)$ is set as the deepth of the concept c in the tree and LCS is the LCS, i.e., the first parent common to both concepts.

Figure 4.6: *Example of a comparison between SNOMED CT concept sets.*



For example, the IC values, or the depth in the tree in this case, for the previous entities 762690000, 17788007, and 400177003 would be 9, 9, and 3. In this way, the similarity value between *Classical Hodgkin lymphoma* and *Acute myelocytic leukaemia* would be $\frac{2 \cdot 3}{9+9} = 0.33$.

$$lin(c_1, c_2) = \frac{2 * IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{4.8}$$

In order to extend this idea to the comparison of entity frequency vectors, we have broken down each concept into the sum of its parents in such a way that the new vectors $\vec{u}$ contain path information for all concepts. Equation 4.9 describes this transformation, with $\vec{e}$ and $\mathbf{M}_p^\top$ being the entity frequency vector of size E and the transpose of the path matrix of size E x E, where each column comprises the co-occurrence of parents of the corresponding entity. Finally, similarity SIM is calculated in Equation 4.10 as the relative percentage of common parents, where the numerator represents the portion shared by both vectors, $\vec{u}_a$ and $\vec{u}_b$.

$$\vec{u} = \vec{e} \times \mathbf{M}_p^\top \tag{4.9}$$

$$SIM(\vec{u}_a, \vec{u}_b) = \frac{2||\vec{u}_a - relu(\vec{u}_a - \vec{u}_b)||^2}{||\vec{u}_a||^2 + ||\vec{u}_b||^2} \tag{4.10}$$

For example, Figure 4.6 shows a tree with 10 nodes and two sets of concepts $s_a = 248437004, 123471000119103, 213026003$ (shown in orange) and $s_b = 698579002$ (shown in green). Before comparing both sets, the representations $\vec{u}_a$ and $\vec{u}_b$ are first constructed. To this end, we have assigned one-hot vectors to nodes using the position shown in the corner of each element (Figure 4.6). Thus, $\vec{e}_a$ and $\vec{e}_b$ include ones in the positions corresponding to the concepts in sets and the rows of the matrix M_p represent all the nodes above each concept (from 1 to 10), as illustrated in Equations 4.11, 4.12, and 4.13. After vector multiplication, $\vec{u}_a$ and $\vec{u}_b$ contain the intermediate nodes of the entire sets (Equations 4.14 and 4.15). The function $relu(\vec{u}_a - \vec{u}_b)$ would yield the difference between the vectors $\vec{u}_a$ and $\vec{u}_b$ but excluding negative values (Equation 4.16). Finally, SIM is estimated by calculating the square module of the vectors (Equation 4.17).

$$\vec{e_a} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1\end{bmatrix} \tag{4.11}$$

$$\vec{e_b} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix} \tag{4.12}$$

$$M_p = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix} \tag{4.13}$$

$$\vec{u}_a = [3, 1, 1, 1, 1, 0, 1, 0, 0, 0] \tag{4.14}$$

$$\vec{u}_b = [1, 0, 1, 0, 0, 1, 0, 0, 0, 0] \tag{4.15}$$

$$relu(\vec{u}_a - \vec{u}_b) = [1, 0, 1, 0, 0, 0, 0, 0, 0, 0] \tag{4.16}$$

$$SIM(\vec{u}_a, \vec{u}_b) = \frac{2 \cdot 2}{8 + 3} = 0.36 \tag{4.17}$$

As for the word frequency vectors, the similarity is estimated as in Section 4.3.1. The final similarity is a weighted average of both similarities.

4.4.1 Experimentation

The semantic-based method has been tested only on the CodiEsp and HUFA corpora, since associating codes to the less verbose expressions found in the certificates from the *Causes of Death* corpus requires more lexical rather than meaning knowledge, implying less complexity.

The settings used and the results obtained are described below.

Experimental settings

The results of the lexical-based settings **IR-TK** and **SVM** from the previous section (Section 4.3.3) have been exhibited in order to analyse the differences with the proposed method. As for the semantic-based approach, a single setting (**S**) has been applied on each corpus by using the label representations that performed best in the lexical similarity approach: the combination of ICD descriptions and coder annotations in the CodiEsp corpus, and the ICD descriptions enriched by KLD terminology for the HUFA corpus.

Results

Table 4.3 shows the results of the semantic-based method together with the other settings described above.

S is the setting that beats the rest in both micro and macro-averaged values in the CodiEsp corpus, with an F-Score value of 49.20. Although there is a relevant improvement in micro-averaged Precision compared to the best lexical-based setting (about 23% respect to SVM), the biggest increase is in macro-averaged Precision, which rises by more than 100%. It is worth noting that the score difference between the prediction of exact and partial matches is small (around 10%), which means that

		CodiEsp			**HUFA**		
		IR-TA	S	SVM	IR-TK	S	SVM
Micro	P	23.41	**69.15**	56.46	27.83	31.49	**35.25**
	R	16.95	**38.19**	34.29	20.43	24.66	**30.73**
	F	19.66	**49.20**	42.67	23.56	27.66	**32.84**
	P_S	31.86	**75.52**	62.13	36.69	39.25	**41.55**
	R_S	23.92	**42.85**	39.98	27.42	30.89	**40.16**
	F_S	27.32	**54.67**	48.65	31.39	34.57	**40.84**
	PSP	22.59	**68.97**	54.52	28.59	**31.05**	30.11
	PSR	14.11	**35.24**	29.46	26.07	**28.71**	22.58
	PSF	17.37	**46.64**	38.25	27.27	**29.84**	25.81
	PSP_S	31.12	**75.51**	60.41	36.77	**38.50**	37.18
	PSR_S	20.41	**39.94**	35.10	33.79	**35.89**	31.25
	PSF_S	24.65	**52.24**	44.40	35.22	**37.15**	33.96
Macro	P	13.15	**32.67**	15.08	36.37	**38.71**	8.34
	R	14.00	**27.82**	18.18	41.72	**43.27**	8.95
	F	13.56	**30.05**	16.49	38.86	**40.86**	8.63
	P_S	15.88	**34.96**	16.22	39.93	**41.04**	12.20
	R_S	17.09	**29.37**	19.16	44.92	**45.47**	12.25
	F_S	16.46	**31.92**	17.57	42.28	**43.14**	12.22
	PSP	10.43	**29.63**	13.53	37.96	**39.56**	6.60
	PSR	11.53	**25.80**	16.57	44.55	**45.18**	7.46
	PSF	10.95	**27.58**	14.90	40.99	**42.19**	7.00
	PSP_S	12.73	**31.69**	14.58	41.09	**41.94**	9.28
	PSR_S	14.12	**27.18**	17.47	47.46	**47.59**	9.74
	PSF_S	13.39	**29.27**	15.89	44.04	**44.59**	9.50
Ordering	$nDCG$	55.45	62.12	**82.54**	64.41	66.67	**70.39**
	$nDCG_S$	65.36	69.47	**85.42**	67.79	69.40	**72.44**
	$PSnDCG$	54.60	62.23	**77.26**	64.41	**66.71**	64.43
	$PSnDCG_S$	64.51	69.49	**79.84**	67.79	**69.39**	67.73

Table 4.3: *Evaluation of the performance of the unsupervised semantic-based method for the CodiEsp and HUFA corpora.*

the erroneous predicted codes tend to pertain to totally different branches from those of the codes assigned by the coders. In contrast to lexical methods, the semantic macro-averaged scores increase up to 90%, reaching an F-Score value of 30.05 and correctly predicting a large percentage of different codes, including minority codes. Again, SVM yields results with more matches in the first positions, which is reflected in the higher nDCG values. In general, supervised methods tend to prioritise predictions better than unsupervised ones.

While the semantic-based method is a promising approach to the task of identifying ICD entities in reports (corpus CodiEsp), its performance does not offer the same improvements in non-academic clinical coding as reflected in the HUFA corpus. One can observe the 17% and 10% increase in micro-averaged F-Score for the semantic-based method on exact and partial code matches compared to the purely lexical-based method. The improvement is smaller when it involves the macro-averaged values (5% and 2% in F-Score for exact and partial matches). Despite the increase in both Precision and Recall for unsupervised approaches, the SVM method still achieves the best micro-averaged results for exact and partial matches: an F-Score value of 32.84 (more than 15% higher). In contrast, SVM is outperformed by the semantic-based method on propensity-scored and macro-averaged values (more than 15% and 370% respectively), indicating that discriminative learning focuses on the most frequent codes, while the unsupervised method practically predicts majority and minority codes equally. In this case, the unsupervised method exceeds the $PSnDCG$ and $PSnDCG_S$ values compared to the SVM, achieving that minority codes are placed higher in the rankings.

4.5 Discussion and concluding remarks

After a review of the state of the art of unsupervised approaches for the classification of clinical texts, we have proposed two methods based on lexical matching and semantic similarity. Both have been evaluated on different corpora and compared with supervised baselines. In the following, we discuss the differences in performance according to the different attributes of the corpora and the possible benefits of these approaches for answering Research Question 2 (see Section 1.3).

Diversity of performance The variation in the performance of supervised baselines and unsupervised lexical and semantic matching-based methods has been explored in three corpora of differing complexity. As discussed in Section 3.2, *Causes of Death* consists of short, unverbose sentences containing the ICD entities, so coding resembles a NER task, requiring more lexical than semantic processing. In turn, the number of example per code is relatively large. Otherwise, *CodiEsp* includes longer and more verbose paragraphs, so it needs more semantic interpretation, but all the entities present are codified. In this case, the number of example per code is scarce. Finally, *HUFA* comprises the longest documents and not all the entities that appear are coded, only those that are related to or derive from the causes of hospitalisation. The number of examples per code is considerably reduced, decreasing the macro-averaged values. With such differences in mind, we notice that supervised lexical-based methods achieve the best results on data sets with many examples per code and short records, such as in the case of death certificates. As document length increases, supervised methods lose effectiveness as they require many more examples than available to identify the relevant information among the noise. Searching for annotations similar to those provided by coders seems to be an option to consider, although terminology extraction through examples shows even more promising results. In addition, non-academic coding of extensive hospital reports is a considerably complex task, where there is potential for improvement on the results offered by the direct evidence search (IR method).

As for the semantic approach in which multiple words and meanings by means of one NER are addressed for each sentence, a significant improvement in the detection of ICD entities in long texts is observed. In the same way, we observe that higher results are obtained for the CodiEsp corpus, whose associated codes are only based on decontextualised matching, outperforming the SVM by more than 15%. The proposal has lost effectiveness with the entities of the HUFA corpus, for which a non-pure matching criterion is needed. In this case, the similarity-based approach is unable to outperform the SVM in terms of micro-averaged scores, reaching 10% lower values. In contrast, does outperform any other approach in macro-averaged values as it has the ability to predict codes not represented in examples; consequently, matches are not so tightly clustered in the most popular codes.

Feasibility assessment Unsupervised methods do not distinguish relevant information within long documents, in contrast to supervised models that learn to discern relevant information. Furthermore, some papers concerning the alignment of ICD and SNOMED CT (Rodrigues et al., 2015, 2017) point out the complexity of establishing direct correspondences, as the two standards define diseases differently. ICD requires a higher level of abstraction that is hard to reach by non-learning methods. Therefore, assignments proposed by unsupervised methods, which lack reasoning processing to filter out entities related to the patient's hospitalisation, usually fail to outperform the micro-averaged results achieved by supervised models in official coding (*HUFA*). Despite this lack of final code selection criteria, unsupervised methods do produce a significant improvement in macro-averaged values, i.e., such methods predict minority codes better than the supervised ones.

We have not detected any purely unsupervised approach in the SOTA that is able to achieve results comparable to supervised approaches when dealing with long texts.

Nor have we implemented any method able to outperform the proposed supervised baselines in terms of micro-averaged F-Score. Although the semantic similarity-based approach is close to the KNN in performance, it is surpassed by 18% by SVM, a method that we consider a baseline given the characteristics of the task. Therefore, we do not consider unsupervised methods competitive compared to supervised ones, which would be part of the answer to the Research Question "*Is it possible to approach ICD-10 coding using unsupervised techniques in a way that can be a competitive alternative to supervised methods?*" (**RQ 2**). However, the huge improvement in macro-averaged values confirms that unsupervised methods do not accumulate the success rate on certain codes, but are able to predict both frequent and underrepresented codes. This functionality is interesting to complement the results.

Coding in health institutions is a complex task that requires some learning to discriminate relevant information from large amounts of data. While supervised models are able to accurately predict more instances, corresponding to the most frequent codes, the contribution of unsupervised methods focuses on coverage, especially of minority codes. The results therefore appear complementary, and in principle some ways of combining both types of approaches could be explored (Chapter 7). Again, in answer to RQ 2 (see Section 1.3), unsupervised methods are required to supplement the supervised approaches with alternative representations based on expert knowledge in order to deal with label sparsity.

In view of the results achieved by the supervised models, it is worth investigating different supervised techniques adapted to label imbalance, which has been addressed in the next chapter (Chapter 5).

CHAPTER

5

DEALING WITH UNBALANCED DATA

Content

This chapter presents different proposals focused on ICD distribution to deal with imbalance and scalability. We have explored two different perspectives: data augmentation for boosting the representation of rare codes and reducing imbalance, and Extreme Multi-label Text Classification (XMTC) algorithms for dealing with imbalance and also tackling scalability.

The following objectives are planned:

- Enrich the representation of non-common codes using clinical ontologies.
- Explore the impact of Machine Translation on data augmentation.
- Conduct a comparative analysis of the behaviour of the different types of XMTC proposals.

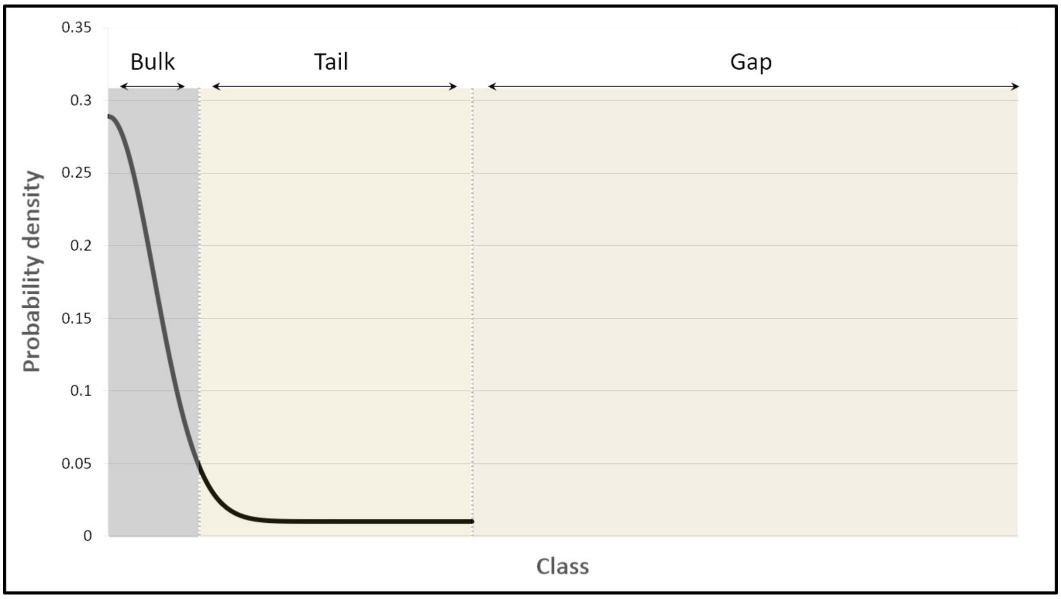
5.1 Introduction

More often than not, real data exhibit biases that predispose to unreliable Machine Learning models, resulting in training data not sufficiently representative for test data. Biases can arise mainly as class imbalance and covariate shift, with ICD data sets typically including both types, as discussed in Section 1.2.2. On the one hand, the difference in disease prevalence leads to the predominance of specific codes over the rest. On the other hand, such data sets are always subject to variations such as the emergence of new diseases, changes in causal factors, and modifications in definitions, introducing fluctuations in new data collections.

Figure 5.1 shows the characteristic data trend of the ICD collections: an extreme label distribution. Codes follows an exponential rather than uniform tendency, with a small percentage of diagnoses much more frequent than others (the bulk), a larger percentage with low presence (the tail), and a majority of codes absent from the set (the gap). Thus, a representative set of EHRs from a hospital will typically be associated with hundreds of common CIE-10-ES codes and thousands of more specific and rarer ones, resulting in a representational gap for tens of thousands of codes reflected in the CIE-10-ES nomenclature.

The difference between the average value of instances in the bulk and the tail represents the strong variation between the frequency of codes, which causes imbalance. The average height of the tail in Figure 5.1 indicates the low number of examples for those codes, which results in data sparsity, while the width of the tail shows the high diversity of classes, resulting in label sparsity and scalability issues. Such characteristics constitute an extreme classification problem. The main difficulty in extreme classifications is that the generalisation and prediction of the most common classes (codes in our case) are favoured over the rest while penalising the rarer ones, at the same time as requiring faster and smaller models than typical classifications.

Figure 5.1: *Outline of an extreme distribution.*



Therefore, novel techniques are needed to deal with such distributions and promote the prediction of less common codes.

In this chapter, we have variations in the training data set via data augmentation methods and algorithms with specific objective functions to address scalability and imbalance issues. The aim is to address the Research Question "*Which techniques can increase the predictive capacity of ICD-10 codes with fewer instances while improving overall system performance? How and how much can the computational complexity of the task be reduced?*" (**RQ 3**). To this end, the improvement of the predictive performance of the models on codes with limited number of examples and decrease in the computational complexity associated with the huge label space and text length have been explored through the following proposals:

- Data augmentation methods. Clinical ontologies and general Machine Translation models have been used to expand the number of training instances with the aim of introducing more lexical diversity to improve generalisation during learning. In particular, we have focused on the least represented codes, which lack sufficient examples to cover the whole range of related synonyms and lexical derivations.

- XMTC proposals. A comparative analysis among traditional approaches and the different families of XMTC algorithms applied to the ICD coding of large EHRs has been carried out. This study also aims to evaluate the impact of document length in addition to the extreme number of labels, as this is directly linked to the number of parameters, i.e., the volume of the models, which is a major factor in scalability and learning.

5.2 Related Works

This section aims to provide an overview of label imbalance in supervised methods and the main SOTA proposals to reduce the negative effect during learning. To do so, the reader is introduced to the SOTA methods of data augmentation and XMTC.

5.2.1 Introduction

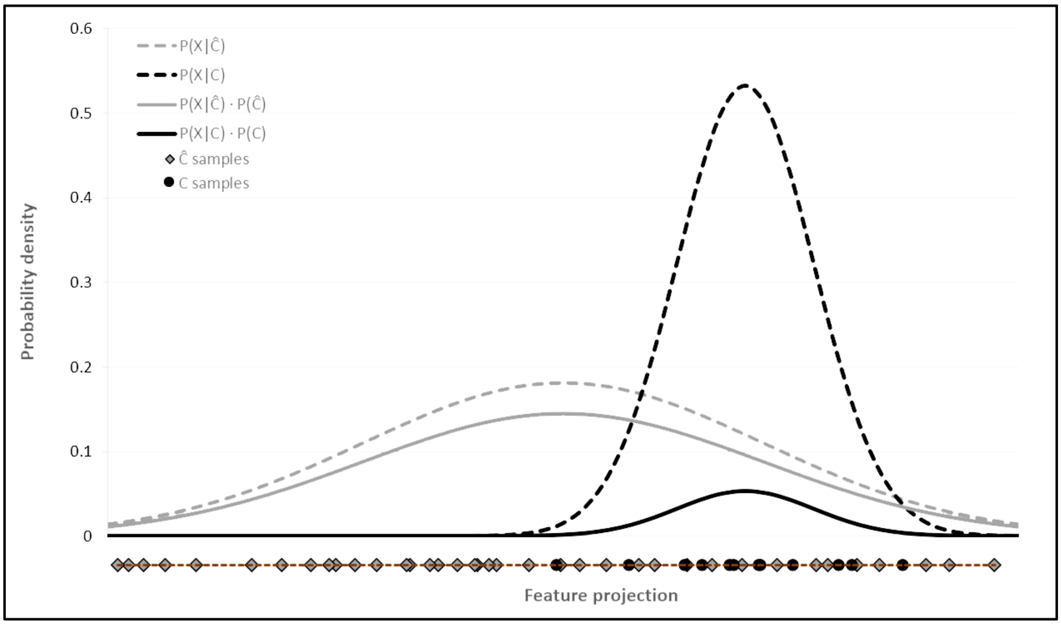
Extreme classification is a relatively recent research topic, but one that is prevalent in many domains. It consists of tagging the data with the most relevant subset among a large number of labels, where the average number of instances per label is low, at most a few orders of magnitude. For this reason, ranking metrics are typically used for the evaluation of such proposals. Data with so many classes tend to follow an exponential distribution, which hampers the learning of Machine Learning algorithms. The main challenges to overcome are the lack of generalisation of the least represented labels, which comprise most of them, and the exponential increase in computational complexity as a function of the number of labels, leading to scalability issues.

Regarding the first point, ML models tend to misclassify minority labels compared to majority labels in extreme distributions, as minority label features are often ignored and processed as noise. For example, Figure 5.2 shows the prior probability distributions of the presence and absence of a particular label in a unbalanced representative data set of 50 data points. In this case, 38 out of 50 data points do not match the label, resulting in 76% of negative instances for the label, or $\hat{C}$. The dashed curves are the probabilities of the features given the presence ($P(X|C)$) or absence ($P(X|\hat{C})$) of the label C, while the non-dashed curves represent the same values weighted by the probability of the presence ($P(X|C) \cdot P(C)$) or absence ($P(X|\hat{C}) \cdot P(\hat{C})$) of the label itself. In turn, the data points or instances are represented in one dimension by pointing out their coincidence or non-coincidence with the label in black or grey respectively. As can be noticed according to Figure 5.2 and the Bayes rule in Equation 5.1, the probability of the label given a data point ($P(C|X) \propto P(X|C) \cdot P(C)$, black curve) is always lower than the probability of not being the label ($P(\hat{C}|X) \propto P(X|\hat{C}) \cdot P(\hat{C})$, grey curve), which means that the highest possible accuracy achieved by a classifier would be accomplished by never matching the label.

$$P(\hat{C}|X) = \frac{P(X|\hat{C}) \cdot P(\hat{C})}{P(X)} > \frac{P(X|C) \cdot P(C)}{P(X)} = P(C|X) \qquad (5.1)$$

As for the second point, the number of possible label combinations is exponential to the number of labels, totalling 2^L possible candidates of label subsets. Let N be

Figure 5.2: *Sample distribution for a class C, where the x-axis represents one-dimensional features and the y-axis indicates the density of probabilities. $P(C)$ is the class probability, $P(\hat{C})$ is the non-class probability, C samples are the instances associated with the class, $\hat{C}$ samples are the instances not associated with the class, $P(X|C)$ is the feature probability given the class, and $P(X|\hat{C})$ is the feature probability not given the class.*



the number of instances and D be the number of features per instance, then the computational cost for training and prediction would be $\mathcal{O}(ND2^L)$. For a corpus like HUFA, with about 36,000 coded documents, around 300,000 unique tokens, and almost 14,000 different codes, the computational cost is of the order of $10^{4,224}$. The motivation for searching for techniques that transform learning and/or inference into sublinear processes arises in order to adapt the training and prediction times to the task and ensure model volumes become manageable.

The different techniques to deal with these extreme imbalance and scalability challenges rely on varying the marginal probability distributions for the training data or the objective functions in models. Traditionally, increasing the number of relevant examples by repeating information has effectively been employed in unbalanced data sets (Chawla, Japkowicz, and Kotcz, 2004; Cheng et al., 2016). Nevertheless, oversampling is not effective in the case of extreme distributions for overcoming imbalance (He and Ma, 2013), as much information such as meaningful patterns and errors are underrepresented in the training data. Thus, new information needs to be introduced to support the existing data, so data augmentation techniques can be used alternatively (Heidari et al., 2019). Otherwise, the use of weights to score instances based on labels has been a popular technique for unbalanced problems, but extreme distributions exceed their limitations. Different mechanisms such as inter-label dependencies and feature overlap have been proposed to deal with both scalability and imbalance. Algorithms that focus on tackling both of these problems contextualised in extreme distributions are called XMTC approaches.

The most relevant types of proposals for data augmentation and XMTC methods are described below. A discussion section has been included to point out the SOTA in relation to ICD coding and to justify the proposals in the chapter.

5.2.2 Data augmentation

Data augmentation consists of expanding the training data with the synthetic generation from the available examples in order to achieve better generalisation. Such techniques introduce slight permutations on the original data that act as a regulariser to reduce overfitting, especially of majority classes.

Data augmentation in computer vision is relatively easy as it requires geometric transformations, which involve fewer constraints. However, text modification implies more complexity due to syntactic and semantic inter-dependencies between all elements, so that the valid options are strongly reduced; for example, only a few words can replace a particular word in a given context.

All changes applied to documents result in new information, of higher or lower quality depending on the type of permutation. The most widespread techniques, ordered from less to more complex, starting with low-level, or lexical, changes and ending with more abstract, or semantic, changes, are described below: random noise injection, lexical substitution, syntax-tree manipulation, back translation and generative techniques.

Random noise injection

Random noise injection is based on lexical unit insertions, replacements, and deletions, with the objective of introducing lexical variability. The changes rely exclusively on simple transformations, which are usually random as knowledge from external information sources is not considered. Character-level methods have been proposed by Coulombe (2018) to simulate spelling errors. Xie et al. (2017) applies word replacements based on the unigram frequency distribution, while Beneš and Burget (2020) and Wei and Zou (2019) introduces random word shifts to train more error-robust models. At the sentence level, Luque (2019) mixes sentence segments while maintaining polarity in a sentiment analysis task. Mixup method has also been explored by Guo, Mao, and Zhang (2019), performing a linear combination of the word embeddings for multiple instances.

Lexical substitution

Lexical substitution mainly relies on the use of synonyms from structured information stored in thesaurus, contextual information captured in word embeddings, and the range of probabilities generated by LM. Regarding the former, Coulombe (2018), Mueller and Thyagarajan (2016), and Zhang, Zhao, and LeCun (2015) use WordNet to search for synonyms of the selected words according to a geometric distribution of the semantic similarity provided by the ontology itself. In contrast, Wang and Yang (2015) exploit the KNN and Cosine Similarity techniques for finding similar candidates via word embeddings. Jiao et al. (2019) use Cosine Similarity on GloVe in a similar way for multi-term expressions, but substitute single words via BERT-style LMs. In this line, other authors (Anaby-Tavor et al., 2019; Fadaee, Bisazza, and Monz, 2017; Garg and Ramakrishnan, 2020; Kobayashi, 2018) use LMs for predicting word candidates by giving surrounding words. An alternative is proposed by Xie et al. (2019), who apply TF-IDF to replace less informative words.

Syntax-tree manipulation

Syntax-tree manipulation aims at paraphrasing sentences, using semantically similar expressions but with different syntax. For example, Coulombe (2018) create manual rules for transforming dependency trees, such as conversion from active to passive voice. In turn, Şahin and Steedman (2019) propose random rotations or deletions of elements in the sintax-based tree.

Back translation

Back translation methods are based on MT techniques to introduce paraphrases and synonyms. The idea is to exploit the asymmetry in translations supported by the differences between the lexicons and syntax of two or more languages to generate text with different words but maintaining the context. Two-step methods are used: a first step to translate the text into an auxiliary language, and a second step to return to the original language. As a result, paraphrased texts, with synonyms and noise, are often obtained. There are a large number of proposals in this regard (Aroyehun and Gelbukh, 2018; Ciolino, Noever, and Kalin, 2021; Coulombe, 2018; Sennrich, Haddow, and Birch, 2015a; Xie et al., 2019).

Generative techniques

Finally, ML have also been used to generate expressions and even whole sentences, thus achieving a higher degree of variability with respect to the original text. For example, multiple authors (Fader, Zettlemoyer, and Etzioni, 2013; Hou et al., 2018; Jia and Liang, 2016; Narayan, Reddy, and Cohen, 2016) generate rules based on context-free grammars from different corpora, which are subsequently applied to construct sentences with the same structure. Other proposals such as the one proposed by Kafle, Yousefhussien, and Kanan (2017) focused on generating whole sentences via LMs. Following this line, Anaby-Tavor et al. (2019) and Kumar, Choudhary, and Cho (2020) use labels as initial words for generating sentences.

5.2.3 Extreme Multi-label Text Classification

In view of the low scalability of processing every possible subset of labels, independence between labels has traditionally been assumed by ignoring dependencies in order to deal with a linear rather than exponential cost to the number of labels. Hence, multi-label classification tasks have been tackled mainly either with algorithms adapted to multiple outputs, such as KNN, decision trees, and neural networks, or by splitting the problem into binary classifications using an OvR strategy. Even so, a linear cost remains unfeasible for most tasks given an extreme distribution. Besides, these perspectives lack inherent mechanisms to deal with the imbalance and do not exploit label co-dependency information. Different authors have made proposals focusing on any of these aspects.

For example, boosting techniques are designed to reduce bias and variance via sequential ensembles of weak models that focus at each iteration on learning the previously misclassified examples. It achieves the promotion of minority classes, as their instances tend to be the harder-to-classify examples from which overfitting is avoided and generalisation is improved. One of the most widespread methods is Adaptive Boosting (AdaBoost) (Freund and Schapire, 1997), which iteratively modifies the sample distribution by fitting the weights of each instance according to the remaining errors.

There have also been proposals that focus on leveraging dependencies between labels to reduce computational complexity, such as Dependency-LDA (Rubin et al., 2012). This proposal is an adaptation of the LDA for capturing word probabilities for groups of labels, and it has been specifically designed to manipulate large amount of rare labels. Dependency-LDA is based on learning multinomial distributions of topics over labels and distributions of labels over words. Hence, the assignments of label-tokens to topics are estimated for a new document during prediction. It requires fewer examples to model as it is generative rather than discriminative, which favours minority classes.

The major advances, however, have been made through the XMTC proposals, which are generally designed to improve the learning of tail labels. XMTC consist of scalability-focused algorithms with the aim of assigning the most relevant subset of labels from a huge category space to each instance. These algorithms are often applied to classification tasks where traditional approaches are not able to effectively

	Training time	Prediction time	Model size	Accuracy
Label independence-based methods	X	X	X	✓
Label embedding-based methods			✓	X
Probability Label Tree-based methods		✓		

Table 5.1: *Summary of common attributes of each type of XMTC method. A check mark indicates strengths and an X indicates weaknesses.*

model the huge unbalanced label distributions due to the computational complexity arising from the scale and heterogeneity of frequencies. Some of the data sets that are typical of this challenge are *Amazon-670K* (McAuley and Leskovec, 2013), *Amazon-3M* (McAuley, Pandey, and Leskovec, 2015; McAuley et al., 2015), *EURLex-4.3K* (Chalkidis et al., 2019a,b), *Wiki10-31K* (Zubiaga, 2012), *Delicious-200K* (Wetzker, Zimmermann, and Bauckhage, 2008), and *WikiLSHTC-325K* (Partalas et al., 2015). The main algorithms proposed to deal with this type of data sets are based on three main foundations:

1. Label independence-based methods, which are based on the reduction of instances or features for classifiers assuming no label co-dependencies.

2. Label embedding-based methods, which exploit the compression of the dimensionality of the label space using embeddings.

3. Probability Label Tree-based methods, which partition the feature or label space with decision trees.

Label embedding and Probability Label Tree (PLT) exploit label co-dependencies by reducing the final number of possible combinations of subsets, while the former approaches try to reduce the training data or features by identifying only the relevant features or data points. Table 5.1 shows the qualities of each family: label independence-based methods achieve the best accuracy at the expense of high computational complexities; label embedding-based methods produce worse results but the best parameter reduction; finally, PLT-based methods can range in training times, with smaller and larger models, but all focused on providing a fast response time during prediction.

Label independence-based methods

The main idea of methods that assume independence between labels is to provide a personalised learning per class, training without any information linking the labels, except for the features themselves. Most proposals focus on One-vs-Rest strategies, training one classifier per label, which reduces the computational complexity to

$\mathcal{O}(NDL)$. In this line, the extreme proposals tackle scalability by using effective subsampling methods for reducing the number of instances N or different regularisation techniques for eliminating features D. Although they do not exploit the information of co-dependencies between labels, they tend to obtain the best predictive accuracy. However, they alone do not reduce the computational complexity beyond a linear relationship with the number of labels, which sometimes prevents their usability for the problems. For this reason, the recent trend is to use them in tree- and embedding-based approaches.

In line with boosting techniques Chen and Guestrin (2016) propose XGBoost, which is a gradient boosting decision tree algorithm that focuses on scalability. Broadly speaking, it uses gradient descent function to optimise the remaining errors while incorporating different memory and computation optimisation techniques, such as a sparsity-aware mechanism to determine the direction of the trees by making the computation linear to the number of non-missing entries, a weighted quantile sketch algorithm for optimally splitting the feature space, and parallelisation and compression techniques.

Alternatively, Babbar and Schölkopf (2017) suggest DiSMEC, which relies on parallelisation methods and L2 regularisation to exploit sparsity via pruning of weights. In contrast, PPDSparse (Yen et al., 2017) focus on subsampling in addition to L_1 regularization to preserve feature sparsity and reduce volume. To this end, the number of negative instances[1] per label is limited, so that each label is trained with all the corresponding examples and some of the instances not related to the label. PPDSparse is 100x faster at training than DiSMEC. Another proposal is Slice (Jain et al., 2019), which relies on negative subsampling by applying Approximate Nearest Neighbour Search methods over a trained generative model to identify hard-to-classify negative instances, i.e., the unlabelled ones which tend to be misclassified. Besides, Babbar and Schölkopf (2019) promote tail labels introducing a Hamming loss with L_1-regularization mechanisms.

In a different line, other authors have explored adapted multi-label classifiers, notably neural networks, with different focus mechanisms for promoting personalised learning for each label. While reducing computational complexity to only $\mathcal{O}(ND)$ is achieved, it is often at the expense of a considerable degradation in performance. For example, Liu et al. (2017) explore XML-CNN, a proposal which uses dynamic max pooling mechanisms for capturing richer information from different regions of the documents, an adapted loss function to increase the weight of relevant retrieved labels, and a fully-connected layer to compress the features. Although the authors achieve competitive results compared to other binary approaches, training and prediction times are far from those of the following algorithms.

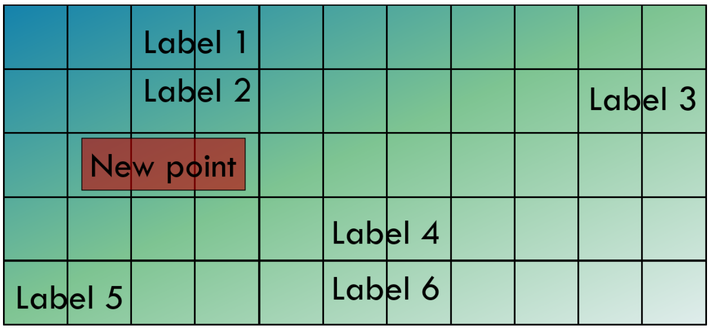
[1] We refer to positive instances for a label in the context of binary classification when examples are tagged with that label, and negative instances when they are not.

Label embedding-based methods

The label space in an extreme distribution is too sparse, so the idea of label embedding-based methods is to exploit label sparsity and correlations to compress the number of labels from L to L', producing new, denser spaces. Thus, the resulting computational complexity would be $\mathcal{O}(NDL')$, making the training and prediction tractable by assuming a low-rank label matrix, i.e., there are many dependencies and few linearly independent classes. For this purpose, those methods linearly transform the high-dimensional label vectors into low-dimensional ones reducing the effective number of labels.

Figure 5.3 shows a graphical representation of this label space. When predicting, the labels closest to the suggested vector in the reduced space given the new point are chosen. In the example it would be labels 1, 2, and maybe 4 or 5. This type of model achieves reduced volumes, optimising the memory. As a counterpart, the more exponential the label distribution, the less dependencies will be involved and the more independent labels there will be, leading to greater information loss and accuracy errors.

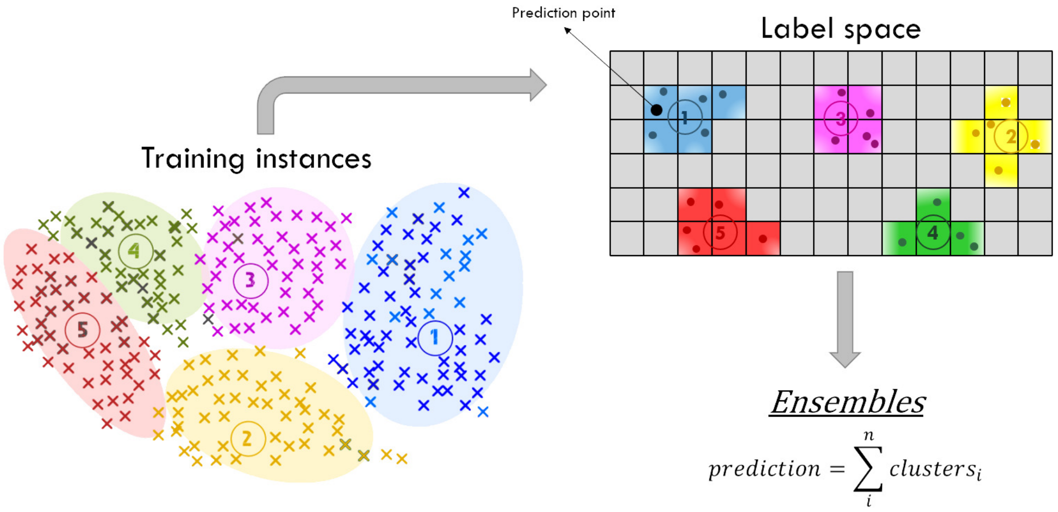
Figure 5.3: *Graphical representation of the reduced label space.*



There are a lot of proposals that use random projections or canonical correlation analysis of label co-occurrences for complexity reduction (Balasubramanian and Lebanon, 2012; Bi and Kwok, 2013; Chen and Lin, 2012; Cissé et al., 2013; Ferng and Lin, 2011; Kapoor, Viswanathan, and Jain, 2012; Mineiro and Karampatziakis, 2015; Tai and Lin, 2012; Weston, Bengio, and Usunier, 2011). Alternatively, Yu et al. (2014) approaches reduction as a task of learning a new global linear space.

One of the most popular proposals is SLEEC (Bhatia et al., 2015), which is illustrated in Figure 5.4. It first performs a clustering of data points to split up the generation of label embeddings with the aim of parallelising. Local non-linear $\hat{L}$-dimensional embeddings are learnt from the original L-dimensional label vectors while preserving pairwise distances between the closest ones for each cluster. At prediction time, the approach performs a KNN search for projecting a novel document in all the $\hat{L}$-dimensional embedding spaces. Finally, the prediction is the result of the ensemble of all clusters.

Figure 5.4: *Outline of the approach SLEEC.*



Tagami (2017) propose an extension of SLEEC (AnnexML) by applying a KNN graph for partitioning the data points and learning embeddings with a ranking objective for replicating the graph into the reduced label space. Wadbude et al. (2017) also extend SLEEC by using the word2vec algorithm for learning the label embeddings, thus introducing co-occurrence information.

Regarding other proposals, RobustXML (Xu, Tao, and Xu, 2016) focuses on tail labels by decomposing the label matrix into tail and non-tail components before generating the label embeddings. Yeh et al. (2017) generate joint feature and label embeddings by learning a feature-aware deep latent subspace. To this end, authors combine deep canonical correlation analysis and autoencoders with label-correlation sensitive loss functions. In this line, Zhang et al. (2018) produce label embeddings using DeepWalk method over the label co-ocurrence graph first, and then generate non-linear embedding in both feature and label spaces simultaneously. Finally, authors apply clustering on the low-dimensional space via k-means.

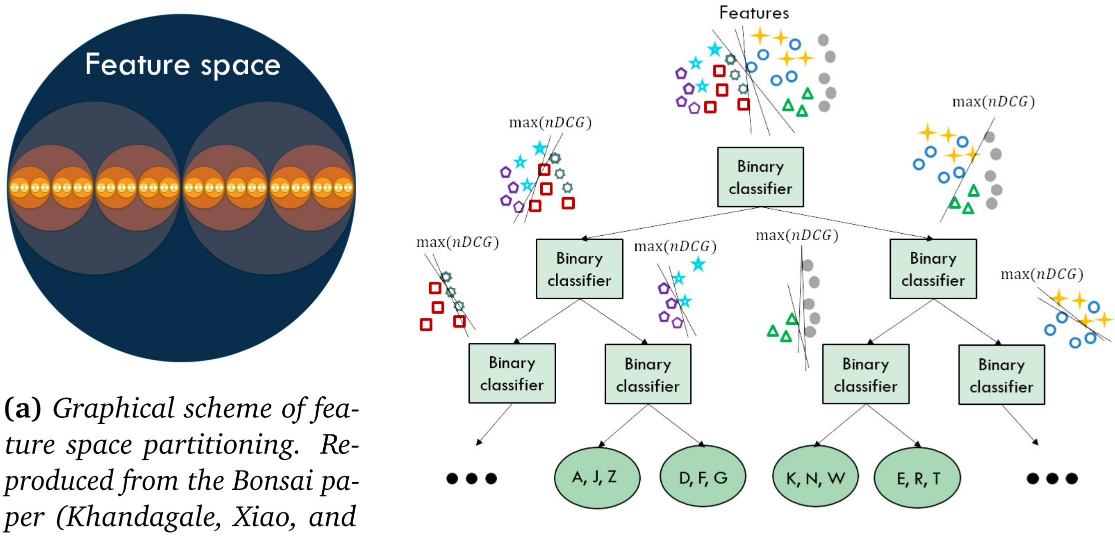
PLT-based methods

Probability Label Tree-based approaches are based on recursive partitioning of the feature or label space by means of binary classifiers, which reduces the computational complexity to $\mathcal{O}(ND\,logL)$. In this way, some approaches reduce the feature space by exploiting overlaps or patterns common to groups of labels, while others segment the label space based on the representation of co-occurrences between labels. This achieves low prediction times, which is one of the most desirable attributes for real-world applications. By contrast, errors propagate in cascades, so that faults occurred at the top of the trees are carried downwards, and as a result, ensembles of many inaccurate trees are required to produce robust approaches.

One of the most widespread early proposals was LPSR (Weston, Makadia, and Yee, 2013), which relies on a multi-label base classifier and a decision tree composed of k-means feature partitions only for predicting. For each new instance, the decision tree is traversed, arriving at a leaf with a subset of labels. The classifier is then used for predictions, limiting the result to the previous subset. Instead, Agrawal et al. (2013) propose a multi-label random forest method (MLRF) by applying an ensemble of trees based on feature splits according to the Gini index.

Another popular algorithm is FastXML (Prabhu and Varma, 2014), which is represented in Figure 5.5. It is based on hierarchical divisions that are recursively learnt by determining which labels should be assigned to the left or right in each child node until each leaf contains a small number of labels. Such decisions are supplied by binary SVM classifiers focused on nDCG-based objective, which split the label set into two subsets so that all the documents in a branch share similar label distribution.

Figure 5.5: *Outline of the FastXML approach.*



(a) *Graphical scheme of feature space partitioning. Reproduced from the Bonsai paper (Khandagale, Xiao, and Babbar, 2020).*

(b) *Scheme of the PLT structure.*

Jain, Prabhu, and Varma (2016) replace the nDCG-based loss by a function based on the propensity scored nDCG in PfastreXML to improve robustness against bias and promote tail label prediction. In turn, SwiftXML (Prabhu et al., 2018a) is designed as an extension of PfastreXML by jointly using the data point features and labels to split the label space. Similarly, Siblini, Kuntz, and Meyer (2018) perform a random projection onto lower dimensional spaces of label and feature vectors, using k-means to split the instances. The Probabilistic Label Tree (PLT) (Jasinska et al., 2016), instead, uses a class probability estimator per node to maximize F-measure and separate the data points associated with the labels from non-labelled data points. In addition, Jernite, Choromanska, and Sontag (2017) design an algorithm for building a tree optimized with Stochastic Gradient Descent (SGD) and focused on node balance and purity. The objetive function for yielding easily separable partitions relies on maximizing the difference between label and global conditional distributions.

Another widespread algorithm is Parabel (Prabhu et al., 2018b), in which the label space is recursively divided into equal groups by k-mean clustering. Each label is represented by a unit vector whose direction is defined by the average of the features of all associated instances (the centroid). A negative subsampling method is adopted to reduce the negative examples (those not related to the labels) to only the ones associated to the most similar labels. As an extension, Bonsai (Khandagale, Xiao, and Babbar, 2020) has been proposed, which represents the label space as the combination of the features and the co-occurrences between labels. The main differences lie in the use of shallow instead of deep trees, with partitions of varying sizes. Shallow trees are more robust to cascade errors and seem to achieve better results. In a different line, Jalan and Kar (2019) explore DEFRAG, which uses agglomeration to effectively reduce the feature dimensionality. Authors divide the data point space and combine all the feature vectors per cluster; then, they produce new features for each data point by estimating one feature per cluster as a function of the distance to the centroid; finally, a hierarchical clustering is performed over the reduced features.

The significant step forward in deep learning comes with the proposal of the architecture AttentionXML. You et al. (2018) explore a label tree-based deep learning model which uses hierarchical shallow probabilistic trees focused on tail labels. Such trees are based on the same partitioning according to the sum of BoW features as Parabel approach (Prabhu et al., 2018b). Each leaf node consists of a multi-label BiLSTM with an attention mechanism to capture label-specific features. You et al. (2019) extend the AttentionXML approach by improving scalability. To do this, the set of labels is divided into smaller subgroups, which are handled as tags in an overall tree. Subsequently, each subgroup is treated independently with its own trees. Alternatively, Medini et al. (2019) propose a strategy (MACH) based on hashing for ensemble of groups of classifiers trained on random subsets of labels.

Chang et al. propose X-bert (Chang et al., 2019) and the extended model X-Transformer (Chang et al., 2020), two approaches based on hierarchical bonsai-style label partitions. Labels are represented by the concatenation of the embedding of the corresponding wikipedia category and the TF-IDF-weighted aggregation of the associated instances, which in turn are represented by the average of the word embeddings. BERT, RoBERTa, and XLNet are trained on the label clusters to match the new instances. Finally, a OvR classifier per label is trained using negative subsampling to reduce computer complexity. Jiang et al. (2021) point out the limitations in computational complexity and model size of the approach X-transformer, and extend the same outline by introducing the following changes: initial clustering based on k-means over the normalised sum of sparse text features; inclusion of the same hidden bottleneck layer used in XML-CNN to project word embeddings into a low dimension; and negative sampling that includes instances from the most similar clusters.

Dahiya et al. (2019) propose an scheme (Dahiya et al., 2019, 2021) focused on short text and based on multiple XMTC mechanisms previously proposed by other authors. DeepXML uses intermediate features learnt from a surrogate task, sub-linear partitions of the label space, negative sampling methods, transfer mechanisms to generate final features, and OvR classifiers trained on the shortlisted labels. Following such an outline, the most recent proposals DECAF (Mittal et al., 2021a), GalaXC (Saini et al., 2021), and ECLARE (Mittal et al., 2021b) have emerged.

5.2.4 Discussion

The publication of textual data augmentation proposals in the clinical domain has been limited by the reduced availability of dedicated resources. For example, Dhrangadhariya et al. (2021) uses Google Translation[2] to introduce lexical changes by extending the set of pathology reports in the classification of the grade of prostate cancer. As regards ICD coding, Velichkov et al., 2020 do not use any external resources, but perform noise injection by randomly modifying letters. In this line, Biseda et al. (2020) explore shuffling the sentences within documents before feeding a BERT-style model. Instead, Garcıa-Santa and Cetina (2020) use a generative model trained on MIMIC-III and PubMed for producing new text for the entities identified in the goldstandard. In turn, Ollagnier and Williams (2020) explore the replacement of random words with WordNet, which is a general-purpose lexical source, and the use of a LM to complete parts of sentences. To the best of our knowledge, there are no published approaches involving domain-specific Lexical Substitution or Back Translations techniques applied to ICD coding.

There is also hardly any literature on the application of XMTC techniques for ICD coding even though such approaches seem to be well-suited to the task. To the best of our knowledge, we have published the first proposal applying such techniques in ICD coding, exploring an ensemble of conventional and XMTC methods (Almagro et al., 2020). Chalkidis et al. (2020) compare the performance of Parabel and Bonsai models, based on Bag-of-Words, with AttentionXML model, which achieves a 15% improvement using vector representations. In turn, the impact of transfer learning using BERT-style models is analysed, with results that do not surpass the performance achieved by AttentionXML, partly due to the limitation of the input length. Alternatively, Zhang, Liu, and Razavian (2020) explore an approach based on BERT and the multi-label

[2] https://translate.google.com

attention output layer from AttentionXML. Given the absence of a broader study on the impact of traditional versus extreme distribution-focused algorithms on ICD coding, a comparative analysis of the different families of XMTC is proposed, with a special focus on the characteristic length of the EHRs. Therefore, both models based on bags of words, generally robust in long texts, and vector representations, more efficient in short texts, are evaluated.

5.3 Data augmentation

One of the main problems of ICD coding is the scarcity of examples for a large number of codes, i.e., the lack of representation for all those labels constituting the tail of the extreme distribution. In that sense, data augmentation is one of the lines of research that is attracting more interest for overcoming this lack. However, the high specialisation of clinical texts complicates such techniques, which is reflected in the few SOTA proposals published in the clinical domain.

This section proposes the use of clinical ontologies to exploit synonymy relations in order to generate new examples by introducing random permutations into existing ones. At the same time, the application of machine translation methods to exploit paraphrases produced by lexical differences between languages as an element of variation has also been explored. While ontologies provide domain-specific changes to relevant information, offering greater generalisation capacity, the general-purpose MT methods aim to introduce variations in non-relevant information, providing greater robustness to noise.

The following is a brief description of the proposed methods for data augmentation, the model used to assess the impact, and the analysis of the results.

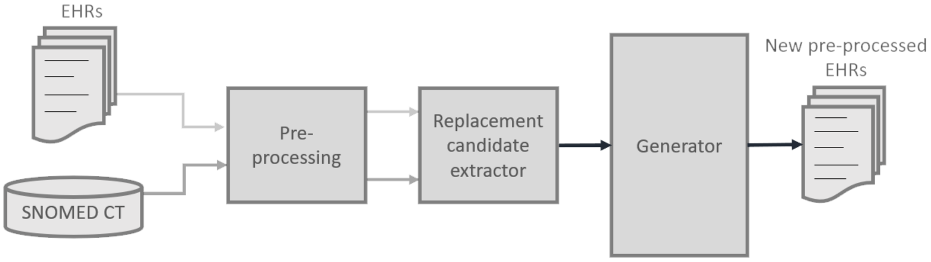
5.3.1 Data augmentation methods

Two data augmentation techniques have been explored: Lexical Substitution based on domain-specific knowledge and Back Translation based on a general-purpose MT model.

Lexical Substitution

SNOMED CT have been used as the main ontology as it provides more than 120,000 different expressions for 50,000 clinical concepts. Unlike the groups of related words used in the preprocessing (see Section 3.3) where precise replacements were preferred to avoid information loss, the multiple expressions associated with a concept have not been used directly as they tend to overlap information introducing small lexical variations. Replacements are made at lexical chunk level. The proposed Lexical Substitution method has been illustrated in Figure 5.6.

Figure 5.6: *Scheme of the proposed Lexical Substitution method for EHRs augmentation.*



Pre-processing First, the pre-processing described in Section 3.3 has been applied to both EHRs and SNOMED CT descriptions.

Replacement candidate extractor Then, the non-overlapping information between pairs of descriptions associated with the same SNOMED CT concept is grouped as long as both descriptions share common information and differ on some term. For example, the concept ***292223000*** corresponds to the expressions "*Adverse reaction to mitozantrone*" and "*Adverse reaction to mitoxantrone*", so the non-overlapping information would be "*mitozantrone*" and "*mitoxantrone*". Similarly, "*cutis laxa with bone dystrophy*", "*cutis laxa with osteodystrophy*", and "*cutis laxa with joint laxity and retarded development*" describe the concept ***73856006***, but we would only be interested in connecting the words "*bone dystrophy*", "*osteodystrophy*", and "*joint laxity and retarded development*". In this way, lexical diversity is increased with greater coverage. It is assumed that the possible loss of information due to less precise but random replacements in a small percentage of the document is minimal. In total, 45,626 pairs of interchangeable expressions have been produced. Once the processed expressions are collected, exact matches are searched for in the EHRs to retrieve a list of all possible substitutions.

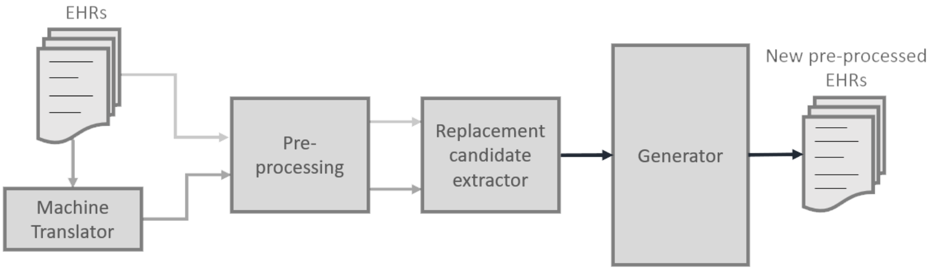
Generator Finally, new preliminary EHRs are produced with a maximum of 20% new information in the *Generator* module, limiting the maximum number of candidate documents to 15 permutations of the original one. In turn, the codes corresponding to the original documents are uniformly distributed among the candidates, proposing C_i as the target number of code associations (Equation 5.2), where f_i is the training frequency of the code C_i and N is the number of training instances. According to the function, the target number decreases exponentially with the frequency of codes, so that more instances are desirable for rare codes and hardly any for common codes, which already have enough examples for covering practically all lexical forms. α and β in Equation 5.2 are parameters heuristically fixed at 11 and 10. α is the asymptote towards which the function tends for the lowest values, so $alpha - 1$ is the maximum number of synthetic examples generated from an original example, while β determines the rate of reduction of the number of synthetic examples as a function of the code frequency. Once all codes have been assigned C_i times or there are no candidate documents for those codes pending to reach the number C_i, the remaining candidates are discarded.

$$C_i = \alpha - e^{\frac{\beta * f_i}{N}} \tag{5.2}$$

Back Translation

The MT model called Marian (Junczys-Dowmunt et al., 2018) and the auxiliary language English have been used to introduce paraphrases in the EHRs. On the one hand, Marian is one of the most widely released MT models in the research community that has demonstrated better performance in multiple language pairs. On the other hand, the use of English as the auxiliary language for translations is motivated by the fact that there are more English content associated with Spanish one than any other language. The proposed Back Translation method has been illustrated in Figure 5.7. It should be note that all replacements are made at the sentence level.

Figure 5.7: *Scheme of the proposed Back Translation method for EHRs augmentation.*



Machine translator First, the model Marian has been feed with the sentences within the EHRs for translating the documents from Spanish to English with the corresponding lexical flattening, and again from English to Spanish, thus modifying certain expressions.

Pre-processing The same pre-processing described in Section 3.3 has been used for both original and translated EHRs.

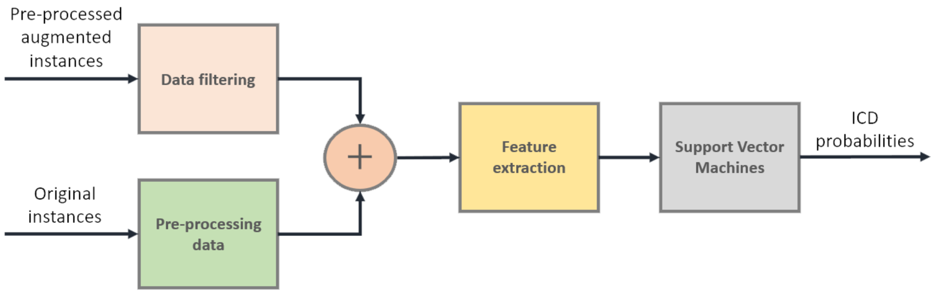
Replacement candidate extractor Next, the list of possible substitutions is produced by comparing the changes between original and translated sentences. If a sentence is modified by the MT model, then it is a candidate for replacing the corresponding sentence in the original example.

Generator Finally, the new EHRs are generated using the same Generator module as in Section 5.3.1.

5.3.2 Classification method

The proposed data augmentation methods aim to strengthen the representation of minority classes. However, 85% of EHRs contains the most frequent codes constituting 1% of the annotated ones, which means that it is not possible to increase the number of instances of minority classes without increasing the representation of majority classes. Therefore, a OvR approach is proposed, training each code separately with classifiers that measure the likelihood of code presence. In this way, it is possible to individually increase the number of positive instances without interfering with the training of the other codes. We have used SVMs in this experimentation as they provide a robust and relatively simple baseline with which to easily observe the impact of data augmentation. The scheme is illustrated in Figure 5.8.

Figure 5.8: *Outline of the proposed ICD classification for using data augmentation methods.*



Pre-processing The pre-processing applied on the original instances is the same as the one described in Section 3.3.

Data filtering Data filtering module works by applying negative sampling, i.e., discarding all but n_{neg} randomly selected instances where the corresponding code does not appear.

Feature extraction Label-specific features have been captured by transforming the pre-processed documents into Term Frequency-Bi-Normal Separation (TF-BNS) vectors (Forman, 2008). To this end, records have been first transformed into Bag-of-Words, also considering trigrams and bigrams; next, a χ^2 feature selection has been applied by reducing the dimensionality; and finally, TF-BNS values are computed for the remaining relevant elements. The steps are detailed below:

- **BoW transformation**. The text tokenised during pre-processing is encoded into a new vector $\vec{x}$, where each coordinate is associated with a specific term within the vocabulary and indicates the frequency in the text.

- **Feature selection**. Dealing with words as discrete categories results in widely sparse features, so a reduction in the number of words for the classification of each code has been implemented. For this purpose, we have measured dependencies between words and codes as more independent words are assumed to be less relevant for classification. Thereby, the idea is to limit the dimensionality of the instance representation vector by only focusing on the most dependent words per code.

  A word W_i is C_i code-independent if the joint probability is equal to the product of the independent probabilities, or $P(W_i, C_i) = P(W_i) \cdot P(C_i)$, which is equivalent to satisfy the joint condition of $P(W_i|C_i) = P(W_i)$ and $P(C_i|W_i) = P(C_i)$. How much the conditional probability of the word W_i given the code C_i $(P(W_i|C_i))$ and the individual probability of the word W_i $(P(W_i))$ differ, or the variation between the probability of the code C_i given the word W_i $(P(C_i|W_i))$ and that of the code C_i $(P(C_i))$, can be quantified by Chi-Square (χ^2).

  χ^2 is described in Equation 5.3, where e_{W_i} and e_{C_i} are binary indices to code the presence of the word W_i and the code C_i respectively (1 being present, and 0 being absent), while $O_{e_{W_i},e_{C_i}}$ and $E_{e_{W_i},e_{C_i}}$ are the number of observed instances and the number of expected ones with the word W_i and the code C_i being presence or absence, depending on indicator. In turn, the number of expected instances $E_{e_{W_i},e_{C_i}}$ is defined in Equation 5.4 as the frequency assuming that the word W_i and the code C_i are independent, where O is the total number of instances, $O_{e_{W_i}}$ is the number of observed instances ocurring or missing the word W_i, and $O_{e_{C_i}}$ is the amount of instances associated or not with the code C_i.

$$\chi^2(W_i, C_i) = \sum_{e_{W_i}=0}^{1} \sum_{e_{C_i}=0}^{1} \frac{(O_{e_{W_i},e_{C_i}} - E_{e_{W_i},e_{C_i}})^2}{E_{e_{W_i},e_{C_i}}} \tag{5.3}$$

$$E_{e_{W_i},e_{C_i}} = O \cdot P(W_i) \cdot P(C_i) = \frac{O_{e_{W_i}} \cdot O_{e_{C_i}}}{O} \tag{5.4}$$

		hypertension		**pressure**		**Total**
		Absence	Presence	Absence	Presence	(Presence and absence)
I10	Absence	8,700	100	7,800	1,000	8,800
	Presence	200	1,000	500	700	1,200
Total (Presence and absence)		8,900	1,100	8,300	1,700	10,000

Table 5.2: *Example of the number of instances for the calculation of χ^2 and BNS values, where the words ”hypertension“ and ”pressure“ are present or absent as a function of the code I10 in a sample of 10,000 EHRs. The last column shows the total number of EHRs associated with the code I10 with independence of the words, while the last row shows the total number of EHRs containing the words.*

An example is shown in Table 5.2, which includes the number of certificate lines in which the word ”*hypertension*“ appears or is missing, the number of instances associated or not with the code I10 (Essential hypertension), and the frequency of instances with both conditions. In this case, the expected frequencies and χ^2 values for ”*hypertension*“ and ”*pressure*“ would be:

$$\begin{aligned}
E_{0,0} &= \frac{8,900 \cdot 8,800}{10,000} = 7,832 & E_{0,0} &= \frac{8,300 \cdot 8,800}{10,000} = 7,304 \\
E_{0,1} &= \frac{8,900 \cdot 1,200}{10,000} = 1,068 & E_{0,1} &= \frac{8,300 \cdot 1,200}{10,000} = 996 \\
E_{1,0} &= \frac{1,100 \cdot 8,800}{10,000} = 968 & E_{1,0} &= \frac{1,700 \cdot 8,800}{10,000} = 1,496 \\
E_{1,1} &= \frac{1,100 \cdot 1,200}{10,000} = 132 & E_{1,1} &= \frac{1,700 \cdot 1,200}{10,000} = 204 \\
\chi^2(&hypertension, I10) = 7287.7 & \chi^2(&pressure, I10) = 1298.8
\end{aligned}$$

Those χ^2 values are high and it is therefore very likely that the words ”*hypertension*“ and ”*pressure*“ are not independent of the code I10 but have a strong dependence.

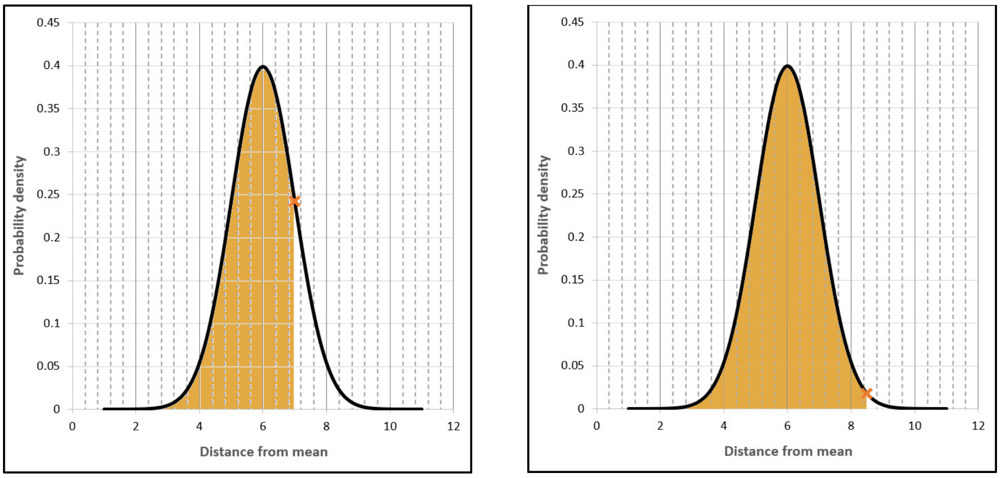
- **TF-BNS value estimation**. Once dimensionality has been reduced, the TF-BNS values for an instance are produced with the frequencies of the remaining words weighted by their Bi-Normal Separation (BNS) scores, which are described in Equation 5.5. Φ^{-1} is the inverse Normal cumulative distribution function, $P(W_i|C_i)$ is the probability of the word W_i given the code C_i, and $P(W_i|\overline{C}_i)$ is the probability of the word W_i given a code other than C_i. Both probabilities are estimated as the ratio of observed instances associated (Equation 5.6) or not (Equation 5.7) with the C_i code in which the word W_i is found.

$$BNS(W_i, C_i) = \left|\Phi^{-1}\big(P(W_i|C_i)\big) - \Phi^{-1}\big(P(W_i|\overline{C}_i)\big)\right| \tag{5.5}$$

$$P(W_i|C_i) = \frac{O_{W_i,C_i}}{O_{C_i}} \tag{5.6}$$

$$P(W_i|\overline{C}_i) = \frac{O_{W_i,\overline{C}_i}}{O_{\overline{C}_i}} \tag{5.7}$$

Figure 5.9: Φ^{-1} *values for the word ”hypertension“ within the Normal distribution according to the example from Table 5.2.*



(a) Φ^{-1} *value for the probability of the word W_i given the code C_i in a Normal distribution.*

(b) Φ^{-1} *value for the probability of the word W_i given a code other than C_i in a Normal distribution.*

BNS scores penalise words that are common in many codes, while promoting words that are not so prevalent in other codes. For example, we can calculate the BNS values for the two words in the example above detailed in Table 5.2, assuming a normal distribution with a mean of 6 and a standard deviation of 1. Figure 5.9 shows the Φ^{-1} values for the word ”*hypertension*“. Hence, the estimates would be:

$$
\begin{aligned}
&P(W_i|C_i) = \frac{1,000}{1,200} = 0.83 && P(W_i|C_i) = \frac{700}{1,200} = 0.58 \\
&P(W_i|\overline{C}_i) = \frac{100}{8,800} = 0.01 && P(W_i|\overline{C}_i) = \frac{1,000}{8,800} = 0.11 \\
&\Phi^{-1}(0.83) = 6.97 && \Phi^{-1}(0.58) = 6.21 \\
&\Phi^{-1}(0.01) = 8.28 && \Phi^{-1}(0.11) = 7.20 \\
&BNS(hypertension, I10) = 1.31 && BNS(pressure, I10) = 0.99
\end{aligned}
$$

Support Vector Machiness The SVM architecture is described in Section 4.3.2.

5.3.3 Experimentation

The impact of the proposed data augmentation methods have been evaluated on the HUFA data set. We have preferred the HUFA corpus for this experimentation because the instances are long documents with abundant information, so it is relatively easy to introduce modifications. Given the computational limitations, we conducted the experimentation on a random sample of 20% of the coded records from HUFA (see Table 3.6), for which experimentation can be carried out in a reasonable time using conventional algorithms. In total, we have collected 7,254 EHRs associated to 76,525 annotated codes, with 5,803 records for training and 1,451 records for testing.

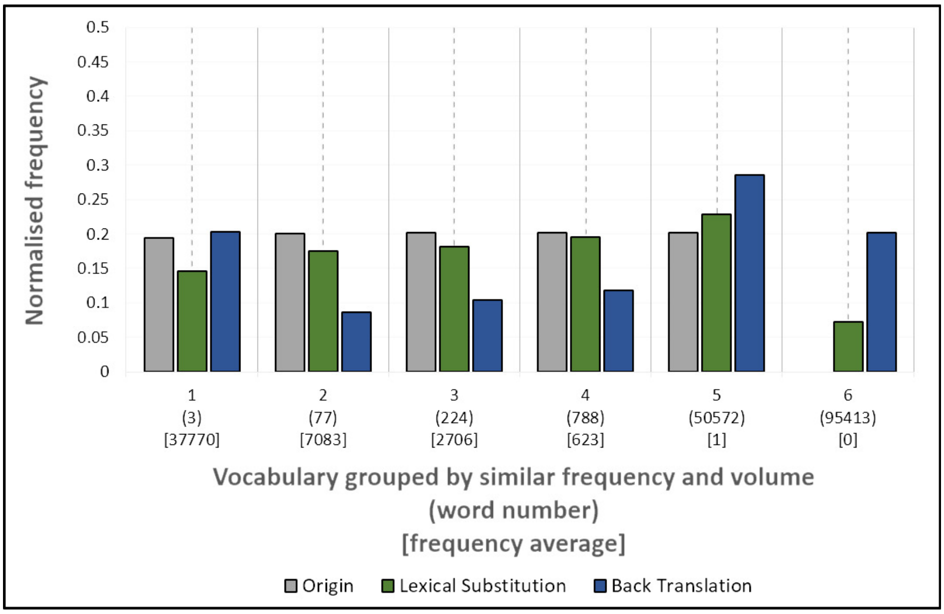
Table 5.3 shows the comparison of vocabularies after pre-processing the original and augmented training data set. Lexical Substitution increases the vocabulary by 20%, adding almost 6 times the number of records, which seems ideal conditions for a classifier: a lot of data with slight changes. On the other hand, Back Translation duplicates the vocabulary with a 4-fold increase in the number of documents, so that a priori we can suspect a worse impact on learning as it introduces excessive lexical diversity in a relatively reduced amount of data.

	Document number	Vocabulary size	Vocabulary increment	Word number	Word increment
Origin	5,803	51,664	0.00%	4,933,684	0.00%
Lexical Substitution	31,198	62,326	20.64%	24,412,572	394.81%
Back Translation	23,646	113,536	119.76%	14,462,535	193.14%

Table 5.3: *Overall vocabulary statistics resulting from the data augmentation methods. Vocabulary increment and Word increment columns are represented as percentages.*

Figure 5.10 shows the normalised vocabulary histrogram for the original (in grey) and augmented data sets (in green and blue). Each bar in the histogram represents a group of words with similar frequencies in the data set. In turn, the groups have been created with the same volume on the original data set; in fact, one can note that the first 5 groups (the sixth group concerns the oov words) have an aggregated relative frequency of 0.2. If one examines how the vocabulary frequency distributions change according to the modifications of each method, one can notice that Lexical Substitution introduces more infrequent vocabulary (group 5 exceeds 0.2) and OOV words (group 6 does not exist in the original data set) while reducing the relative volume of those words that are more frequent, e.g., group 1 drops to almost 0.15 (from 0.2). So we can state that Lexical Substitution produces a more specialised vocabulary if we assume that frequent words tends to be more generic. In contrast, Back Translation increases the relative volume of both very frequent (group 1) and new vocabulary (group 6) at the expense of the volume of the rest. Much of the new vocabulary corresponds to untranslated English words, which could be of benefit where English terminology is used.

Figure 5.10: *Vocabulary histogram after applying the data augmentation methods. Each bar groups words with similar frequencies in the data set.*



The settings implemented and the scores achieved in the different metrics are described and analysed below.

Experimental settings

The contribution of introducing synthetic instances derived from Lexical Substitution (**LS**) and Back Translation (**BT**) during learning has been analysed, with the **Baseline** approach being the same models trained on the original data set exclusively.

The number of negative instances for each code, n_{neg}, is heuristically fixed to 10 times the number of positive instances. In addition, the number of features selected by χ^2 has been set at 1,000. The SVM model has been set with a Radial Basis Function (RBF) kernel, coefficient of 2e-7, λ parameter of 1, and optimisation limit less than 1e-3 improvements in loss. We have used a non-linear kernel because of the large amount of noise found in the clinical records.

Results

Table 5.4 shows the results of incorporating the documents with the variations produced by SNOMED CT and the MT model.

As can be seen, the data augmentation method based on Lexical Substitution, which exploits domain-specific knowledge, outperforms all metrics when compared to the baseline. It increases micro-averaged Precision (P) by 10% and macro-averaged P by more than 15%,. A joint significant improvement of micro- and macro-averaged values means a better prediction of minority codes with no degradation in the performance of the most common codes. The gain is lower when assessing partial matches (P_S, R_S, and F_S), so Lexical Substitution improves accuracy and reduces partial failures. It also improves the position of the matches in the output code ranking per document, as can be seen in the increase of $nDCG$, $nDCG_s$, $PSnDCG$, and $PSnDCG_S$.

Otherwise, the Back Translation method only slightly increases micro averages while worsening the macro ones. In particular, micro-averaged P barely rises by 1% and P_S remains hardly unchanged. In contrast, macro-averaged P and P_S drops by more than 15% and 20% respectively. Empirically we have observed that Back Translation increases the vocabulary with non-existing words or words from another domain, which introduces a significant amount of noise. Lexical noise affects macro rather than micro measures in the case of extreme imbalance. It is worth pointing out that the scores corresponding to the position of matches are higher than the baseline but do not outperform Lexical Substitution except with $PSnDCG_S$. This may be because minority codes tend to be predicted with less confidence by the models, so that they tend to be at the bottom of the output rankings. In turn, Back Translation tends to reduce the prediction of minority codes and may eliminate those in lower positions.

Figure 5.11 shows macro-averaged F_S broken down into groups of codes with similar frequencies while preserving the same volume of all groups within the training set. As one can notice, the most significant improvements in Lexical Substitution are achieved with the minority code groups (exceeding the baseline in groups containing codes with training frequency between 1 and 185), thus confirming the effectiveness of the method in improving the learning of underrepresented codes. Conversely, Back Translation does not follow a clear frequency-related pattern: while there is an improvement in the first four groups, it only significantly exceeds the baseline in other three separate cases.

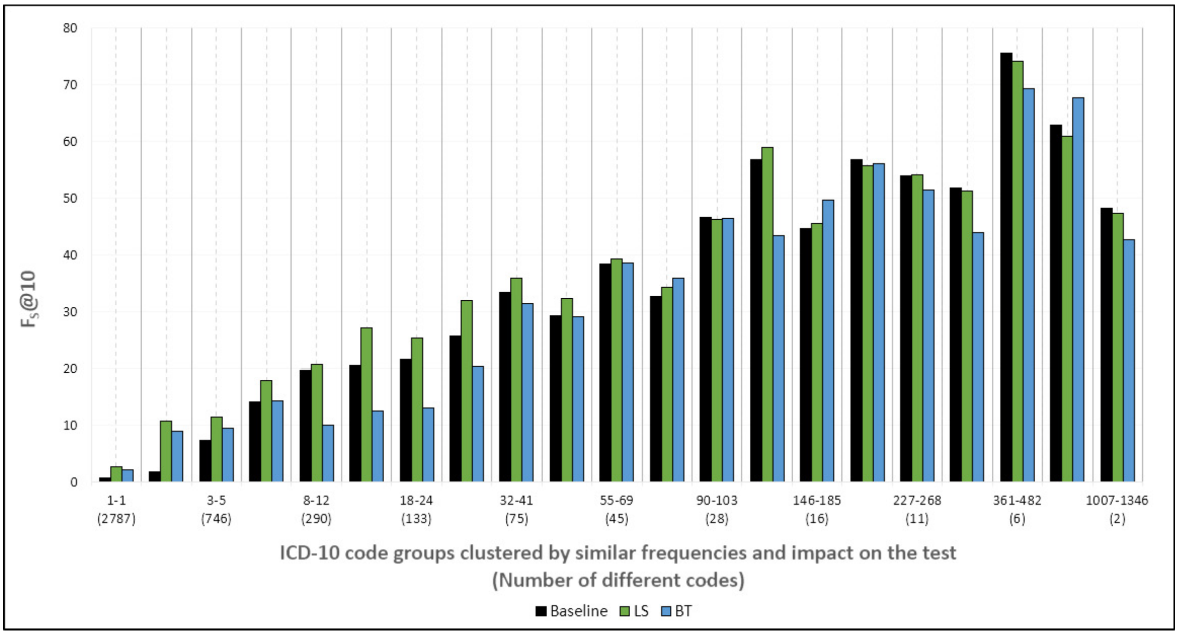
To summarise, the Lexical Substitution method based on SNOMED CT combined

		Baseline	**LS**	**BT**
Micro	P	38.11	**41.92**	38.58
	R	34.7	**35.64**	35.46
	F	36.32	**38.52**	36.95
	P_S	46.9	**50.89**	46.97
	R_S	42.87	**43.71**	43.41
	F_S	44.8	**47.03**	45.12
	PSP	32.62	**37.09**	37.24
	PSR	25.69	**26.47**	25.06
	PSF	28.74	**30.89**	29.96
	PSP_S	42.22	**46.92**	46.79
	PSR_S	33.95	**34.56**	32.50
	PSF_S	37.64	**39.80**	38.36
Macro	P	9.26	**10.78**	7.76
	R	9.91	**11.35**	7.84
	F	9.57	**11.06**	7.80
	P_S	13.51	**15.83**	10.66
	R_S	13.87	**15.41**	9.97
	F_S	13.69	**15.62**	10.31
	PSP	7.21	**8.70**	6.02
	PSR	7.98	**9.50**	6.08
	PSF	7.58	**9.08**	6.05
	PSP_S	10.8	**13.11**	8.42
	PSR_S	11.43	**13.16**	7.84
	PSF_S	11.11	**13.14**	8.12
Ordering	$nDCG$	76.57	**77.79**	77.10
	$nDCG_S$	82.86	**84.74**	84.04
	$PSnDCG$	70.64	**71.86**	71.66
	$PSnDCG_S$	76.38	77.08	**78.24**

Table 5.4: *Evaluation of the impact of Lexical Substitution (LS) and Back Translation (BT) methods for data augmentation.*

with a OvR strategy and a negative subsampling technique is effective in improving the representation of minority classes resulting in a generalised improvement of all metrics. The Back Translation method based on the Marian model also seems to significantly improve the representation of codes with up to 5 training instances. However, the translations seem to introduce noise in most of the training, resulting in a generalised worsening of the scores.

Figure 5.11: *Similarity F-Score (F_S) at 10 for both data augmentation techniques. The scores are shown as a percentage.*



5.4 Extreme classification

Section 5.3 pointed out the complex ICD distribution comprising very narrow and pronounced heads, constituted by a small group of over-represented codes, and huge tails composed of under-represented codes. This imbalance causes learning problems for the underrepresented codes, as they tend to be overridden by the vast volume of underrepresented codes from the head. Another way to deal with imbalance is to modify the objective function to promote minority codes, and this is precisely what extreme algorithms do. To this end, a large percentage of the authors have focused on exploiting inter-label dependencies to increase representativeness in addition to the subsampling techniques that many of them apply.

Besides, training algorithms on data sets annotated with thousands of classes involves huge computational costs, often unfeasible for the time demanded by the task. In particular, EHRs tend to be descriptive and informative, which is attached to a larger number of features and poses an additional challenge. Dealing with scalability is therefore another main bottleneck in ICD coding. Among the diverse proposals for reducing computational complexity are learning parallelisation (generally assuming independence between classes), reduction of non-relevant instances, feature compression, and exploitation of co-dependencies to reduce the space of possible subsets.

		Features					Foundations				
		BoW	KLD	TF-IDF	TF-BNS	Word embeddings	Similarity estimation	Probability distribution	OvR	Clustering	Parameter sharing
Unsupervised method	**SIM**		✓	✓			✓				
Label independency	**B-SVM**				✓				✓		
	AdaBoost				✓				✓		
	B-MLP				✓				✓		
	B-LSTM					✓			✓		
	RCNN					✓					✓
Label dependency	**KNN**			✓			✓				
	D-LDA	✓						✓			
XMTC-Label independency	**B-XGBoost**				✓				✓		
	XML-CNN					✓					✓
XMTC-Label embeddings	**SLEEC**			✓			✓			✓	
	AnnexML			✓			✓			✓	
XMTC-PLT	**FastXML**			✓						✓	
	Parabel			✓						✓	
	Bonsai			✓						✓	
	AttentionXML			✓		✓			✓	✓	✓
	DeepXML			✓		✓			✓	✓	✓
	DECAF			✓		✓			✓	✓	✓

Table 5.5: *Overview of the features and foundations of each method.*

This section proposes a comparative study of the different techniques employed in multi-label classification, adapting them to the coding task and analysing the relative training and prediction times. We have focused on the approaches that best deal with the imbalance and computational times rather than model space as we assume that memory problems are easier to solve. In such a study, we explore whether XMTC algorithms are the most efficient option to tackle ICD coding using ML techniques. Below we describe each of the methods implemented (see Section 5.4.1), detail the settings, and analyse the scores reached.

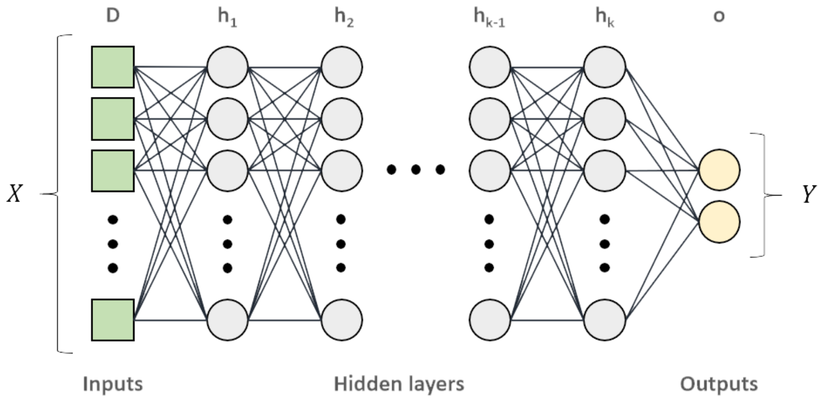
5.4.1 Methods

Precision and Recall are typically on opposite sides of the optimisation, so maximising both is often difficult. Depending on the priority given to each one, there are a variety of classification methods based on differing representations and mechanisms. In this section, we have compared alternative types of algorithms with the aim of finding out the differences and similarities in the inference.

For this purpose, we have evaluated unsupervised approaches which rely on high Recall but relatively low Precision, supervised approaches that assume label independence and are characterised by high Precision but relatively low Recall, and supervised approaches exploiting label dependence to achieve high Precision while preserving Recall. In turn, we have distinguished conventional methods from extreme algorithms among the supervised approaches, with the latter being grouped into three families. Therefore, we can finally distinguish up to 6 types of algorithms: Unsupervised approaches, standard approaches based on label independency, standard approaches based on label dependency, XMTC approaches based on label independency, XMTC approaches based on label embeddings, and XMTC approaches based on PLT. For each type, we have selected some representative approaches. Although others could have been considered, this comparative study aims to provide general conclusions in terms of the type of approach. Table 5.5 shows a summary of the representations and foundations of the selected approaches, which are detailed below:

- **Unsupervised approaches**:
  - **SIM**. The proposal has been explained in Section 4.4. The computational complexity for indexing would be $\mathcal{O}(NLD)$, where N is the number of instances, L is the number of codes, and D is the vocabulary size. The prediction based on similarity reduces the complexity to $\mathcal{O}(D)$.
- **Standard approaches based on label independency**:
  - **B-SVM**. This method has been described in Section 5.3.2. In summary, documents have been represented as TF-BNS vectors and a different SVM model has been trained per code. As these features are label-specific, the representation of each document varies depending on the code to be modelled. During prediction, the new EHR is transformed into a different representation per code to feed the model that produces the assignment probability. The training computational complexity for OvR classifiers is $\mathcal{O}(LN^2D^2)$, where N, L, and D are the number of instances, codes, and features respectively. In turn, the prediction complexity is $\mathcal{O}(LKD)$, with K being the number of support vectors.
  - **B-MLP**. The approach outline is the same as the B-SVM approach but using Multi-Layer Perceptron (MLP) as models. Hence, a MLP has been trained per code to infer the assignment probability to the EHRs. Since this is a one-to-many cardinality multi-label problem (multiple codes per document), the set of codes finally associated to a EHR are the result of the aggregation of the output of all classifiers. A k-hidden-layer Fully Connected Neural Network (FCNN) with h neurons each has been used for each code, as shown in Figure 5.12. Equation 5.8 describes the o outputs of the network with k layers as a function of the weights W and outputs of the preceding layers, where σ is the activation function, x are the inputs, and b are the bias values. For a binary classification, a Binary Cross Entropy (BCE) function (Equation 5.9) is usually applied to the output to calculate the probability of each label, the presence or absence of the code in this case.

Figure 5.12: *Binary FCNN architecture for each code. The size of each layer is shown at the top.*



$$\hat{y} = a_o^k = \left[\sigma\left(\sum_m W_{nm}^L\left[\cdots\left[\sigma\left(\sum_j W_{pj}^2\left[\sigma\left(\sum_i W_{ji}^1 x_i + b_j^1\right)\right] + b_p^2\right)\right]\cdots\right]_m + b_n^k\right)\right]_o \tag{5.8}$$

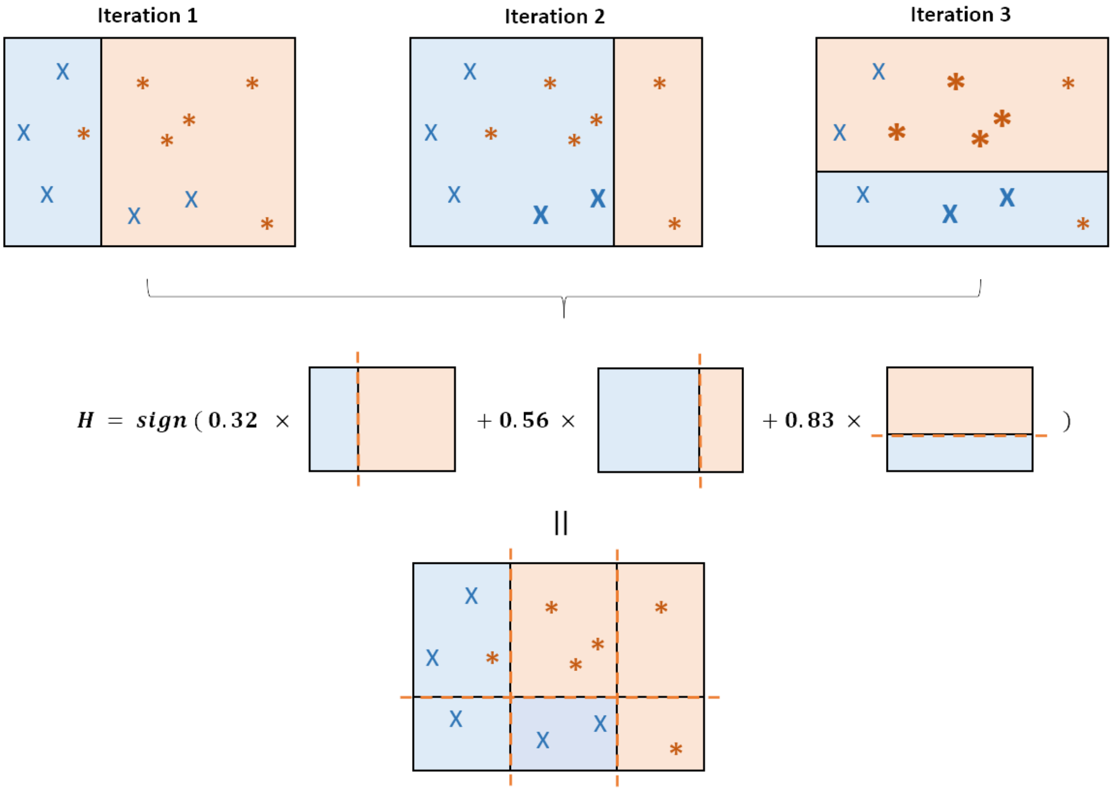
$$CE(\hat{y}, y) = -y \cdot log(\hat{y} + (1 - y) \cdot log(1 - \hat{y})) \tag{5.9}$$

It implies a computational complexity $\mathcal{O}(LNDh^k)$ for training, where L is the number of codes, N is the number of instances, D is the number of features, and k is the number of layers containing h neurons. Prediction complexity is reduced to $\mathcal{O}(LDh^k)$.

- **B-AdaBoost**. The same scheme as for the approach B-SVM has been used. Instead of SVM models, an AdaBoost model has been trained for each code, which is based on iterative training focusing on errors. The main idea is to train multiple sequential classifiers per code, which are weighted according to the accuracy achieved, as described in Equation 5.10, where E would be the error rate across all instances. Figure 5.13 shows an example of the way the algorithm works.

  At each stage, the previous trained classifier is used to identify misclassified instances and weight them as described in Equation 5.11 for the next training, where $D_t(i)$ is the weight of the instance i, α_t is the weight of the classifier t, y_i is the ground truth, $h_t(x_i)$ is the prediction of the previous classifier, and Z_t is the sum of all the weights for normalisation. After several iterations, all the classifiers corresponding to the appropriate code are ensembled according to Equation 5.12.

Figure 5.13: *Example of learning the AdaBoost algorithm.*



$$\alpha_t = 0.5 * ln\big((1-E)/E\big) \tag{5.10}$$

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \tag{5.11}$$

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{5.12}$$

This method involves the computational complexity $\mathcal{O}(KNlog(N)D)$ for training and $\mathcal{O}(LKlog(N))$ for predicting, where L is the number of codes, K is the number of trees, N is the number of instances, and D is the number of features.

- **B-LSTM**. Binary classifiers based on a LSTM are ensembled using a OvR strategy. An overview of the classifiers is shown in Figure 5.14, with the pre-processed EHRs transformed into word sequences. The embedding layer uses the weight matrix $E^{(0)}$ to convert the word vector $\vec{v}_j$ of the instance j into in a sequence of vectors (Equation 5.13), which are further processed by the LSTM units. The maximum values of the hidden states from all the stages are projected onto a dense layer, resulting in a softmax function that yields the relevance probabilities of the input documents to a code.

$$y^{(1)} = E^{(0)} \cdot \vec{v}_j \tag{5.13}$$

Equations 5.14, 5.15, and 5.16 described the mathematical behaviour of a LSTM unit, where [$W^{(i)}$, $W^{(f)}$, $W^{(o)}$, $W^{(c)}$, $U^{(i)}$, $U^{(f)}$, $U^{(o)}$, $U^{(c)}$, $b^{(i)}$, $b^{(f)}$, $b^{(o)}$, $b^{(c)}$] are weight matrices. x_t is the word vector within $y^{(1)}$ in the timestep t, σ is the sigmoid function, and h_t is the current exposed hidden state. Figure 5.15 shows the architecture of a LSTM unit.

A step can be expressed in three parts:

- ⋆ The operation of the gates in Equations 5.14, where f_t is the forget gate's activation vector, i_t is the input gate's activation vector, and o_t is the output gate's activation vector.

$$\begin{aligned} i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \\ f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(t)}) \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) \end{aligned} \tag{5.14}$$

Figure 5.14: *Overview of an LSTM-based neural network for binary classification.*

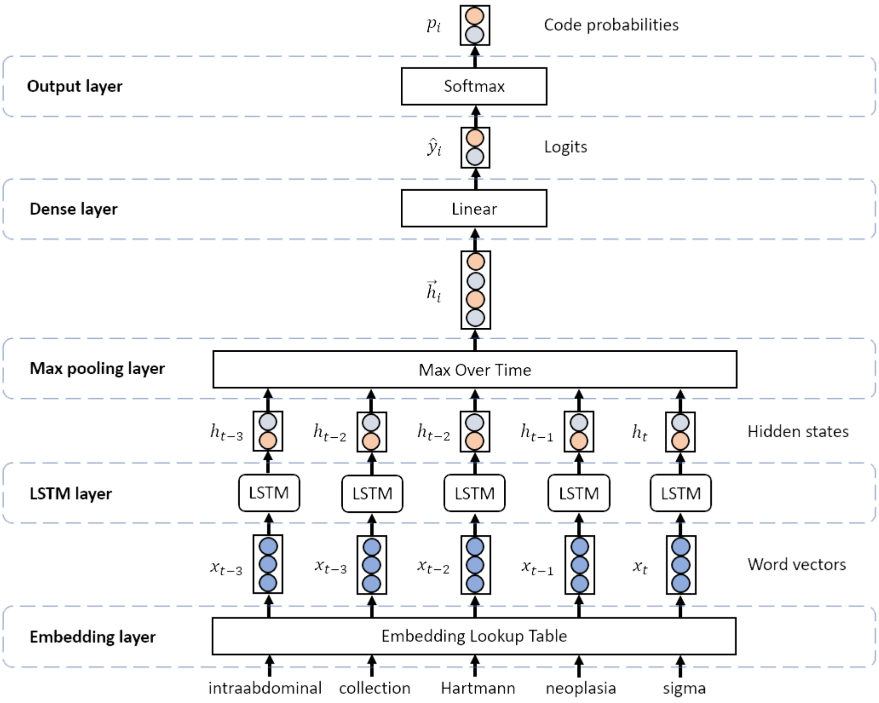
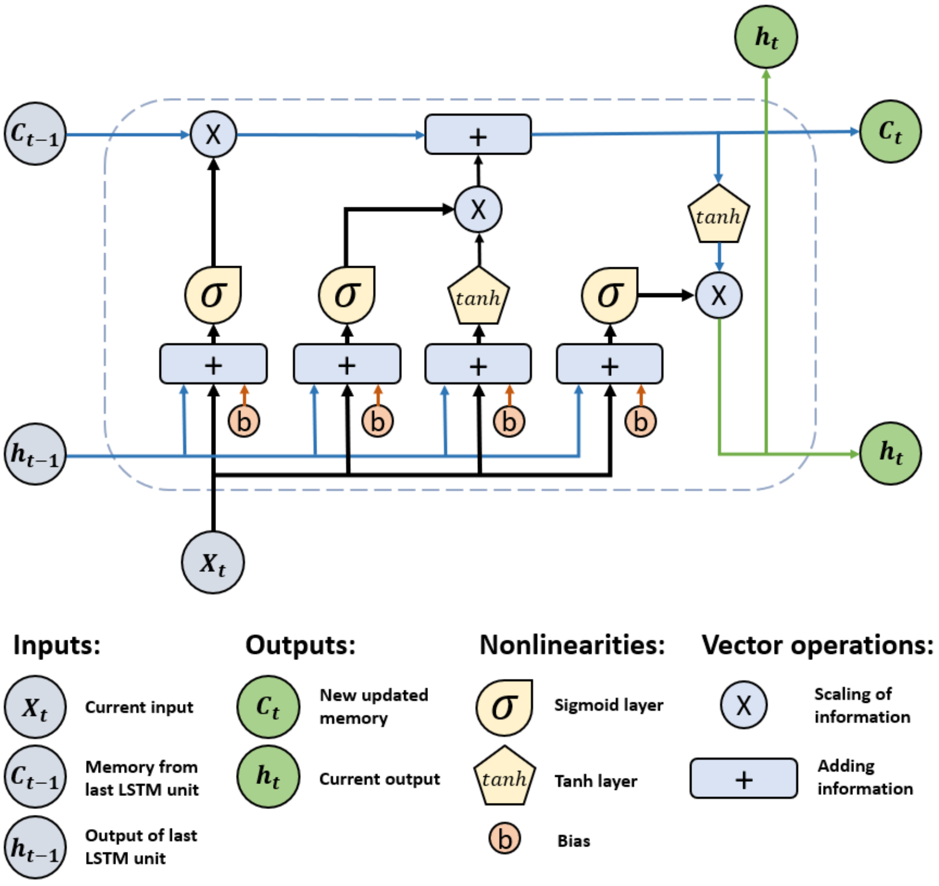


Figure 5.15: *The structure of the LSTM neural network reproduced from Le et al. (2019).*



- ⋆ The update of the current state of the cell in Equation 5.15, with $\tilde{c}_t$ being the cell input activation vector.

$$\tilde{c}_t = tanh(W^{(c)}x_t + U^{(c)}h_{t-1} + b^{(c)}) \tag{5.15}$$

- ⋆ The estimation of the next hidden state in Equation 5.16, where c_t is the cell state vector and $\odot$ is the element-wise multiplication.

$$\begin{aligned} c_t = i_t \odot \tilde{c}_t + f_t \odot +\tilde{c}_{t-1} \\ h_t = o_t \odot tanh(c_t) \end{aligned} \tag{5.16}$$

The output of the LSTM layer $y^{(3)}$ is composed of the hidden states from all steps, which is then collapsed into two variables $y^{(4)}$ across a dense layer (see Equation 5.17). $y^{(4)}$ values are associated with the presence or absence of the code c. Finally, a probability for each case p_i is computed by applying a softmax function, where L is the number of codes.

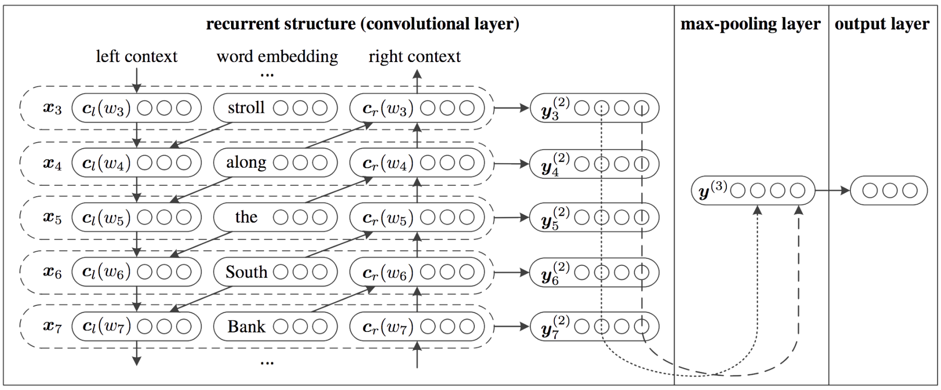
$$
\begin{aligned}
y^{(3)} &= \max_{t=1}^{T} h_t \\
y^{(4)} &= \sigma(W^{(4)} y^{(3)} + b^{(4)}) \\
p_i &= \frac{e^{y_i^{(4)}}}{\sum_{k=1}^{L} e^{y_k^{(4)}}}
\end{aligned}
\tag{5.17}
$$

All parameters are learnt with BCE loss and Adam optimizer. The computational complexity for training is $\mathcal{O}(BTDL)$, where B and T are the batch and document sizes, D is the embedding dimension, and L is the number of codes. Instead, the prediction cost is $\mathcal{O}(TDL)$.

– **RCNN**. RCNN is founded on both recurrent and convolutional architectures. The model is focused on enriching word representations ($w_t \in \vec{v}_j$) with left ($c_l(w_t)$) and right ($c_r(w_t)$) context information (Equation 5.18). The idea is to jointly learn the matrix of representations E together with the matrix pairs ($W^{(l)}$, $W^{(r)}$) and ($W^{(sl)}$, $W^{(sr)}$), with the former including the transformations for moving from the previous context, $c_l(w_{t-1})$ or $c_r(w_{t+1})$, to the next, and the latest providing the operations for incorporating the last word vector, $e(w_{t-1})$ or $e(w_{t+1})$, into the current context. Both terms are subjected to an activation function σ.

$$
\begin{aligned}
e(w_t) &= E \cdot \vec{v}_{j,t} \\
c_l(w_t) &= \sigma\left(W^{(l)} c_l(w_{t-1}) + W^{(sl)} e(w_{t-1})\right) \\
c_r(w_t) &= \sigma\left(W^{(r)} c_r(w_{t+1}) + W^{(sr)} e(w_{t+1})\right)
\end{aligned}
\tag{5.18}
$$

Figure 5.16: *Architecture of a RCNN reproduced from Lai et al. (2015b).*


recurrent structure (convolutional layer)
max-pooling layer
output layer
left context
word embedding
right context
...
x_3 $c_l(w_3)$ stroll $c_r(w_3)$ $y_3^{(2)}$
x_4 $c_l(w_4)$ along $c_r(w_4)$ $y_4^{(2)}$
x_5 $c_l(w_5)$ the $c_r(w_5)$ $y_5^{(2)}$
x_6 $c_l(w_6)$ South $c_r(w_6)$ $y_6^{(2)}$
x_7 $c_l(w_7)$ Bank $c_r(w_7)$ $y_7^{(2)}$
...
$y^{(3)}$

The RCNN scheme is described in Figure 5.16. First, a sequential architecture is applied to the input x_t, which is the result of the concatenation of the corresponding word vector and the context vectors (5.19).

$$x_t = c_l(w_t) \oplus e(w_t) \oplus c_r(w_t) \tag{5.19}$$

A linear followed by a non-linear transformation (activation function $tanh$) is applied to each representation x_t to produce the output of the second layer $y_t^{(2)}$ (Equation 5.20). Then, a convolutional architecture is adopted by using a max-pooling mechanism over the document length T to yield the output the third layer $y^{(3)}$. The last one is a full-connected layer for adapting the network output to the number of codes L. A softmax function is applied in order to generate the probabilities for each code i.

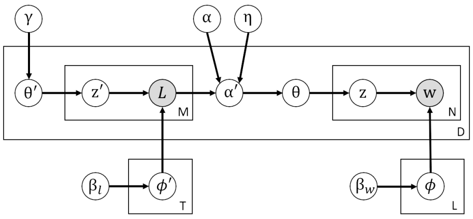
$$\begin{aligned} y_t^{(2)} &= tanh(W^{(2)}x_t + b^{(2)}) \\ y^{(3)} &= \max_{t=1}^{T} y_t^{(2)} \\ y^{(4)} &= W^{(4)}y^{(3)} + b^{(4)} \\ p_i &= \frac{e^{y_i^{(4)}}}{\sum_{k=1}^{L} e^{y_k^{(4)}}} \end{aligned} \tag{5.20}$$

All parameters are learnt with BCE loss and Adam optimizer. The computational complexity for training is $\mathcal{O}(BTD)$, where B is the batch size, T is the maximum document size, and D is the embedding dimension, while the complexity for predicting is $\mathcal{O}(TD)$.

- **Standard approaches based on label dependency**:
  - **KNN**. This method has been described in Section 4.3.2.The complexity for indexing and predicting is the same as **SIM**.
  - **D-LDA**. In contrast to all previous methods, this algorithm is generative, so better performance is expected for minority codes. This proposal is based on Monte Carlo Markov Chain (MCMC) methods for modeling the words in documents as a mixture of labels, so that the label probabilities are learnt at word level rather than document level while label dependencies are jointly captured. The idea is to build a Markov chain by estimating the marginal probability $P(w|d)$ of the word w in a document d as $\sum_l P(w|l)P(l|d)$, i.e., the sum of the joint probabilities of the word w given the label l and the label l in the document d. Hence, each label l is modeled as a multinomial

distribution ϕ_l over words (word-label distribution) and each document d is modeled as a multinomial distribution θ_d over the observed labels (label-document distribution). Figure 5.17 reproduces an overview of the architecture with the variables described below.

Figure 5.17: *Architecture of the model Dependency-LDA reproduced from Rubin et al. (2012).*



As for the word-label distribution, the latent indicators $z_i^{(d)}$ for all words are learnt as a modified form of the collapsed Gibbs sampler in Equation 5.21, where $\hat{\phi}_{w,l}$ is the estimated probability of the word w given the label l and $\hat{\theta}_{l,d}$ is the estimated probability of the label l given the document d. In turn, $\mathcal{W}$ and $\mathcal{L}$ are the set of observed words and labels, N_{wl} is the number of times the word w is assigned to the label l in the training data set, and N_{ld} is the number of times the label l is assigned to another word in the document d. N_l is the frequency of the label l in the training data set, $N_{\mathcal{L}}$ is the total number of assigned labels, α and η are scaling parameters for controling the probability of minority labels, and $\beta_{\mathcal{W}}$ is a parameter for promoting the likelihood of minority words.

$$\begin{aligned} P(z_i^{(d)} = l | w_i^{(d)} = w, \mathcal{W}, \mathcal{L}, \alpha_l, \beta_{\mathcal{W}}) \propto \hat{\phi}_{w,l} * \left(N_{ld} + \alpha_l\right) \\ \hat{\phi}_{w,l} = \frac{N_{wl} + \beta_{\mathcal{W}}}{\sum_{w'=1}^{\mathcal{W}} \left(N_{w'l} + \beta_{\mathcal{W}}\right)} \\ \hat{\theta}_{l,d} = \frac{N_{ld} + \alpha_l}{\sum_{l'=1}^{\mathcal{L}} \left(N_{l'd} + \alpha_i\right)} \\ \alpha_l = \eta * \frac{N_l}{N_{\mathcal{L}}} + \alpha \end{aligned} \tag{5.21}$$

Regarding the label-document distribution, labels for each document are sampled from a set of abstract topics. Similarly, the latent topic indicators $z_i'^{(d)}$ are computed as described in Equation 5.22, where $\hat{\phi}'_{l,t}$ is the probability of the label l given the topic t and $\hat{\theta}'_{d,t}$ is the probability of the topic t given the document d. $\mathcal{L}$ and $\mathcal{T}$ are the set of observed labels and defined topics, N_{lt} is the number of times the label l is assigned to the topic t in the training data set, and N_{dt} is the number of times the topic t is assigned to another label in the document d. γ and β_L are the offset for the minimum probability values of rare topics and labels.

$$\begin{aligned} P(z_i^{\prime(d)} = t | l_i^{(d)} = l, \mathcal{L}, \mathcal{T}, \gamma, \beta_L) &\propto \hat{\phi}'_{l,t} * \left(N_{dt} + \gamma\right) \\ \hat{\phi}'_{l,t} &= \frac{N_{lt} + \beta_L}{\sum_{l'=1}^{\mathcal{W}} \left(N_{l't} + \beta_L\right)} \\ \hat{\theta}'_{d,t} &= \frac{N_{dt} + \gamma}{\sum_{d'=1}^{\mathcal{L}} \left(N_{d't} + \gamma\right)} \end{aligned} \tag{5.22}$$

In prediction, $z^{(d)}$, $l^{(d)}$, and $z'^{(d)}$ vectors are sequentially updated for the test documents in the chain as described in Equation 5.23 by freezing $\hat{\phi}_{w,l}$, $\phi'_{l,t}$, $\theta'_{t,d}$, and $\hat{\phi}'_{l,t}$. Finally, such vectors are used to estimate the document distribution $\hat{\theta}_d$ over all labels (see Equation 5.21).

$$\begin{aligned} P(z_i^{(d)} = l | w_i^{(d)} = w, \mathcal{W}, \alpha_l, \hat{\phi}_{w,l}) &\propto \hat{\phi}_{w,l} * \left(N_{ld} + \alpha_l\right) \\ P(l_i^{(d)} = l | \theta'_d, \phi') &\propto \sum_{t=1}^{T} \phi'_{l,t} \cdot \theta'_{t,d} \\ P(z_i^{\prime(d)} = t | l_i^{(d)} = l, \gamma, \hat{\phi}'_{l,t}) &\propto \hat{\phi}'_{l,t} * \left(N_{dt} + \gamma\right) \end{aligned} \tag{5.23}$$

As for the complexity, the model requires $\mathcal{O}(\frac{N_{\mathcal{W}} N_{\mathcal{L}}}{N} + T N_{\mathcal{L}})$ for each training iteration, where $N_{\mathcal{W}}$ is the number of words in the data set, $N_{\mathcal{L}}$ is the number of assigned labels, T is the number of topics, and N is the number of documents. Instead, the prediction complexity is $\mathcal{O}(N_{\mathcal{W}}(L + T))$, with L being the number of unique labels.

- **XMTC approaches based on label independency**:
  - **B-XGBoost**. This method follows the same OvR strategy: a XGBoost classifier per code is trained. XGBoost is based on sequential decision trees specially trained to correct previous errors. Unlike other algorithms based on gradient boosting, Taylor expansions are used to optimise the loss function rather than negative gradients. The main idea is to follow an iterative process in which greedy trees are built to fit the pseudo-residuals, i.e., the difference between the current approach and the known correct target vector, and used to generate new ones, shortening the distance to the ground true values.

    Equation 5.24 describes the objective function that requires minimisation, where n is the number of instances, l is the loss function for decision trees,

x_i and y_i are the feature and label vectors for the instance i, $\hat{y}_i^{(t-1)}$ are the predictions made by the previous learner, and f_t is the function to be learnt by the decision tree. This first term implies the loss function for estimating the pseudo residuals from the predicted value. Instead, the second term involve a regularization mechanims to penalize building complex tree, where γ represents the penalty to encourage pruning, T indicates the number of leaves, λ is the regularization term, and w are the weights assigned to the leaves.

$$\min_{f_t} \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad \text{where} \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{5.24}$$

l requires to be transformed into an Euclidean function in order to use traditional optimization techniques. In this way, a Taylor approximation is applied yielding the simplified objective to minimize at step t in Equation 5.25. Finally, a scoring function $\tilde{\mathcal{L}}^{(t)}(q)$ is reached for the learner q at iteration t in Equation 5.26, which is used to measure the loss gain.

$$\begin{aligned} \min_{f_t} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \\ \text{where} \quad g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), \quad h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \end{aligned} \tag{5.25}$$

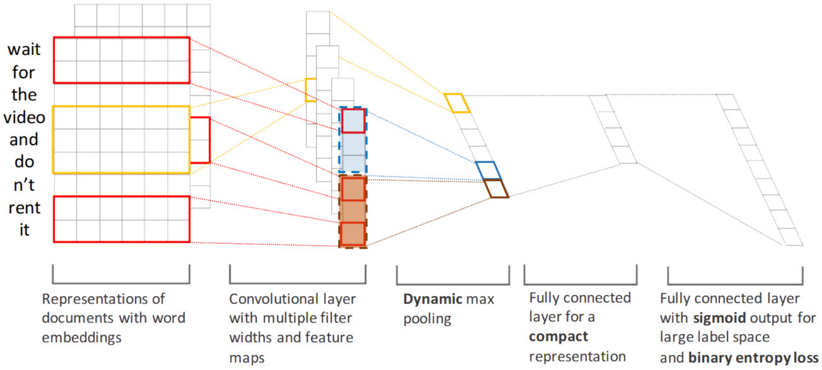
$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{5.26}$$

During the tree building, all residuals on one leaf are splitted into two new nodes only if the sum of the scores of both nodes exceeds the parent's score. The exact greedy tree is designed to iterate over all feature values in order to find the split that results in the maximum loss reduction. Afterwards, all branches with gain less than the threshold γ are pruned. The new learner is used to produce the predictions used in the next iteration. λ value increases with t to ensure tree diversity. The final code probability is achieved by summing the score of all the tree learners.

The computational complexity for training is $\mathcal{O}(LKd||X||_0 log(N))$, where L is the number of codes, K is the number of trees, d is the maximum depth of trees, $||X||_0$ is the number of non-missing entries in the feature matrix, and N is the number of instances. In turn, the predicting cost is $\mathcal{O}(LKdlog(N))$.

- **XML-CNN**. This proposal is one of the first neural networks designed for the Extreme Multi-label Text Classification. The foundations of the convolutional architecture (shown in Figure 5.18) lie in a dynamic max pooling for capturing fine-grained features from different sections of the EHR, a dense layer for reducing parameters, and a BCE loss over sigmoid output.

Figure 5.18: *Architecture of the model XML-CNN reproduced from Liu et al. (2017).*


wait
for
the
video
and
do
n't
rent
it
Representations of documents with word embeddings
Convolutional layer with multiple filter widths and feature maps
Dynamic max pooling
Fully connected layer for a **compact** representation
Fully connected layer with **sigmoid** output for large label space and **binary entropy loss**

Given an instance with T words, $c^{(2)}_{j,t}$ are the features for the word t within the current instance j after applying the convolution filter v_q to the text region from the t-th word to the k-th word (Equation 5.27). x are the word representations, σ is the sigmoid function, and h_q is the size of the filter q. Multiple filters are used for capturing different fine-grained sequential features $c^{(2)}_j = [c^{(2)}_{j,1}, ..., c^{(2)}_{j,T}]$

$$c^{(2)}_{j,t} = \sigma(v_q^T x_{t:k+h_q-1}) \tag{5.27}$$

Instead a max-over-time pooling for generating a single feature for each filter, p features are captured by dividing $c^{(2)}_{j,t}$ into chunks for dealing with long documents as shown in Equation 5.28.

$$P(c_{j,1:T}) = [\max\{c_{j,1:\frac{T}{p}}\}, ..., \max\{c_{j,T-\frac{T}{p}+1:T}\}] \tag{5.28}$$

All pooled features are proyected into a dense layer for reducing parameters, and then a fully-connected layer is used for adapting the output to the number of codes (Equation 5.29), with $[W^{(3)}, W^{(4)}, b^{(3)}, b^{(4)}]$ being the weight matrices to be learnt.

$$
\begin{aligned}
y^{(2)} &= [P(c_{1,1:T}), ..., P(c_{j,1:T})] \\
y^{(3)} &= W^{(3)} y^{(2)} + b^{(3)} \\
y^{(4)} &= W^{(4)} y^{(3)} + b^{(4)}
\end{aligned}
\tag{5.29}
$$

Finally, a binary cross-entropy loss over sigmoid activation is defined as the objective function in Equation 5.30, where Θ represents the model parameters, N is the number of instances, L is the number of codes, σ is the sigmoid function, $y_{j,i}$ is the ground truth value for the instance j and the code i, and $\hat{y}_{j,i}$ is the predicted value for the same instance and code.

$$
\begin{aligned}
&\min_{\Theta} -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{L} \left[y_{j,i} log\big(\sigma(\hat{y}_{j,i})\big) + (1 - y_{j,i}) log\big(1 - \sigma(\hat{y}_{j,i})\big) \right] \\
&\qquad\qquad\qquad\qquad\qquad \text{where} \quad \sigma(x) = \frac{1}{1 + e^{-x}}
\end{aligned}
\tag{5.30}
$$

The complexity of the proposal **XML-CNN** for training and predicting is $\mathcal{O}(h(pq + L))$, where h is the number of words in the considered text region, q is the number of convolutional filters, and p is the number of features extracted from the filters.

- **XMTC approaches based on label embeddings**:
  - **SLEEC**. A TF-IDF vector representation has been used for each document. The method SLEEC is based on learning label embeddings for transforming the task into a sublinear one. For this purpose, clustering is performed on the set of instances to build different label vector spaces and parallelise the process. For each of these clusters a non-convex and non-differentiable objective function, which is shown in Equation 5.31, is proposed.

$$
\min_{V_c} \sum_{i \in \mathcal{I}_c} \sum_{j \in \mathcal{N}_{Y_c}^{(i)}} \left| \left| y_i^{\top} y_j - x_i^{\top} V_c^{\top} V_c x_j \right| \right|^2 + \lambda \sum_{i \in \mathcal{I}_c} |V_c x_i|_1 + \mu ||V_c||_F^2 \tag{5.31}
$$

The first term defines the sum of squared errors between the inner product of label vectors y_i and y_j and the inner product of embedding vectors x_i and x_j, where V_c is the projection matrix, $\mathcal{N}_{Y_c}^{(i)}$ is the set of nearest neighbors for the instance i in a cluster $\mathcal{I}_c$ with the partition c, x_i and y_i are the feature and label vectors for the current instance, and x_j and y_j are the feature and label vectors for the neighbor j.

In turn, the second and third terms involve L_1- and L_2-regularization mechanisms, with the former preserving the sparsity in the embedding space and the last one preventing the overfitting within the projection matrix V_c. Hence, λ and μ are regularization parameters.

At prediction, a new EHR is placed in the most similar cluster and the associated label vector is calculated. The closest labels within such vectorial space are assigned. The computational complexity for training and predicting is $\mathcal{O}(N\hat{L}^2 + N\hat{L}\bar{n})$, where N is the number of instances, $\hat{L}$ is the label-embedding dimensionaliy, and $\bar{n}$ is the average number of neighbors.

- **AnnexML**. The overall strategy of the proposal is based on the SLEEC method but introducing the KNN algorithm into the clustering and changing the objective function to a ranking-based one. Given the same variables i, c, $\mathcal{I}_c$, $\mathcal{N}_{Y_c}^{(i)}$, x_i, y_i, x_j, and y_j as in Equation 5.31, the new objective function is defined in Equation 5.32, where γ is a scaling parameter and k is the corresponding instance from the set of randomly selected instances $S_c^- \subseteq \mathcal{I}_c$ in the partition c.

$$\min_{V_c} \sum_{i \in \mathcal{I}_c} \sum_{j \in \mathcal{N}_{Y_c}^{(i)}} log\left(1 + \sum_{k \in S_c^-} e^{-\gamma \Delta_{ijk}}\right)$$
$$\text{where} \quad \Delta_{ijk} = \frac{x_i^\top V_c^\top V_c x_j}{||V_c x_i||\,||V_c x_j||} - \frac{x_i^\top V_c^\top V_c x_k}{||V_c x_i||\,||V_c x_k||} \tag{5.32}$$

AnnexML requires a computational complexity similar to the one of *SLEEC*.

- **XMTC approaches based on PLT**:
  - **FastXML**. It is one of the most representative tree construction algorithms for extreme classification. FastXML is based on ensembles of weak and fast Probability Label Trees or learners generated by recursive partitioning of the feature space, originally composed of TF-IDF values. The objective function of each learner is described in Equation 5.33, where the first term is the L_1 regularization of the linear separator to be learnt W for forcing a sparse linear separation. The second term is a logarithmic loss where x_i is the feature vector of the instance i, C_δ is a parameter to weight the term, , and δ_i indicates if the instance i has been assigned to the positive cluster (value $+1$) or negative cluster (value -1). The third and fourth terms maximize the $nDCG@L$ score of the predicted label rankings for the positive (r^+) and negative (r^-) clusters, with y_i being the label vector of the instance i, y_{r_l} being the label in the position l within the ranking r, and C_r being the parameter to weight such terms.

$$\begin{aligned}
\min \quad & ||W||_1 + \sum_i C_\delta(\delta_i) log(1 + e^{-\delta_i W^\top x_i}) \\
& -C_r \sum_i \frac{1}{2}(1+\delta_i)\mathcal{L}_{nDCG_{@L}}(r^+, y_i) \\
& -C_r \sum_i \frac{1}{2}(1-\delta_i)\mathcal{L}_{nDCG_{@L}}(r^-, y_i) \qquad (5.33) \\
\text{where} \quad & \mathcal{L}_{nDCG_{@L}}(r, y) = \frac{\sum_{l=1}^{L} \frac{y_{r_l}}{log(1+l)}}{\sum_{l=1}^{min(L, 1^\top y)} \frac{1}{log(1+l)}} \\
w \in R^D, \quad & \delta \in \{-1, +1\}^L, \quad r^+, r^- \in \Pi(1, L)
\end{aligned}$$

The final predictions are estimated with the chain rule by multiplying the probabilities of all preceding nodes. As for the computational complexity, it is roughly proportional to $\mathcal{O}(KDlog(L))$, where K is the number of trees, D is the number of word in the vocabulary, and L is the number of labels.

- **Parabel**. Another PLT-based proposal is Parabel, which uses a k-means clustering over the label representation based on the normalised sum of the features of the label-relevant instances, typically TF-IDF values. The idea is to implement an ensemble of trees, each of which is designed with the global objective function described in Equations 5.34 for partitioning the label space within a node with L labels into two balanced groups. V_l is the label representation, $y_{il} \in 0, 1$ indicates the presence or absence of the label l, and $x_i \in R^D$ is the feature vector for the instance i. In turn, α_l indicates if the label l has been assigned to the positive cluster (value $+1$) with mean μ_+ or negative cluster (value -1) with mean μ_-.

$$\max_{\mu_\pm \in R^D,\, \alpha \in -1, +1^L} \frac{1}{L} \sum_{l=1}^{L} \left(\frac{1+\alpha_l}{2} \mu_+^\top V_l + \frac{1-\alpha_l}{2} \mu_-^\top V_l \right)$$

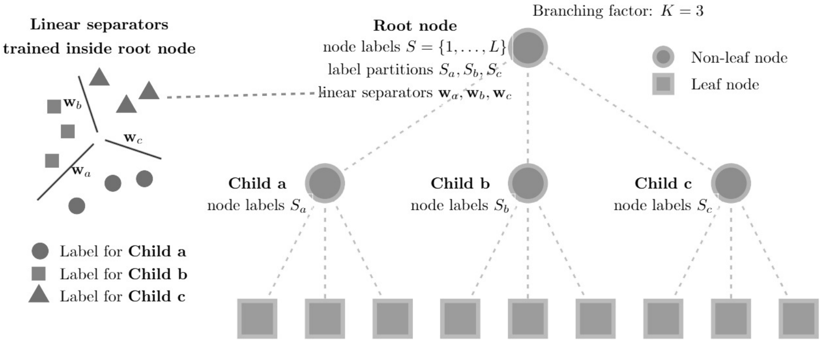
$$\text{where} \quad V_l = \frac{V_l'}{||V_l'||_2}, \quad V_l' = \sum_{i=1}^{N} y_{il} x_i, \quad ||\mu_\pm||_2 = 1, \quad -1 \le \sum_{l=1}^{L} \alpha_l \le 1 \qquad (5.34)$$

The computational complexity of such a proposal is $\mathcal{O}((L/N + M/q + log(L))KNqDlog(L))$ for training, where L is the number of labels, N is the number of instances, D is the vocabulary size, K is the number of trees, M is the maximum number of labels per leaf node, and q is the number of partitions per node. Inference is performed by combining all trees, with a complexity of $\mathcal{O}(TPDqlogL + KPDM)$, where P is the number of the most probable paths at each level.

- **Bonsai**. This method is an extension of Parabel. Label representations are similar to those from Equation 5.34, but x_i represents the concatenation of the feature vector and label co-occurrences vector. Figure 5.19 shows a schematic of the tree constructed by the proposed method. Label partitioning is also performed via k-means clustering in Equation 5.35, where K is the number of clusters, c_k is the center within the cluster k, and V_l is the vector representation of the label l, member of the cluster. Deep trees lead to poorer performance, so that shallow trees are generated with large number K of unbalance leaves in this case. As for the computational complexity, the authors claim that it is 3 times higher than that of **Parabel**.

$$\min_{c_1,\ldots,c_K \in R^{\nu}} \left[\sum_{k=1}^{K} \sum_{l \in c_k} 1 - V_l^{\top} \cdot c_k \right] \tag{5.35}$$

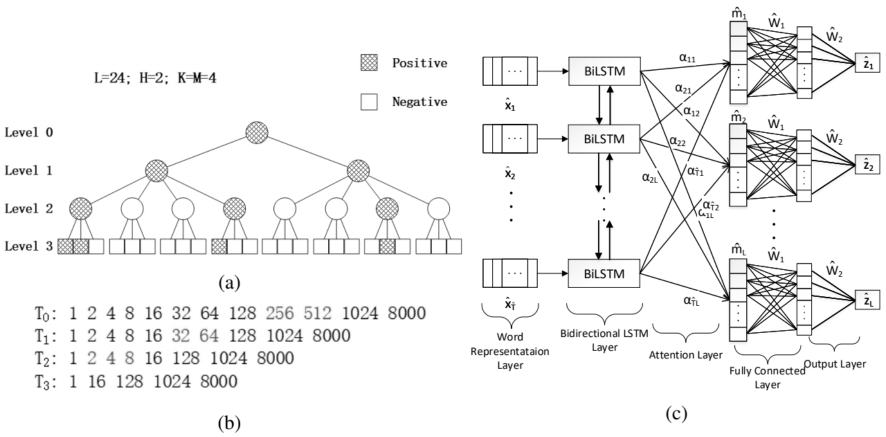
Figure 5.19: *Overview del modelo Bonsai reproduced from Khandagale, Xiao, and Babbar (2020).*



- **AttentionXML**. The proposal is a deep learning method combined with a PLT structure. AttentionXML uses the same label partitioning as Parabel (Equation 5.34) for producing deep PLTs, and then, the trees are compressed into a shallow one by pruning intermediate nodes.

  A multi-label neural network is trained per level but only on the top C label candidates per instance generated by the preceding classifier. The candidates are ranked by the scores prioritising the labels within the node, so that the use of such a sample of labels acts as a negative sampling. Each neural network consists of five layers: a word representation layer, bidirectional LSTM layer, multi-label attention layer, fully-connected layer, and output layer. Figure 5.20 illustrates the architecture comprising these layers.

Figure 5.20: *Outline of the model AttentionXML, reproduced from You et al. (2018).*



Equation 5.13 details the embedding layer with the weight matrix $E^{(0)}$, while Equations 5.14, 5.15, and 5.16 describe the behaviour of an LSTM layer. A bidirectional LSTM comprises forward and backward LSTM units, with the output being the concatenation of the output of both components (Equation 5.36).

$$\hat{h}_t = \vec{h}_t \oplus \overleftarrow{h}_t \tag{5.36}$$

The attention mechanism learns the matrices [$W^{(j)}$, $b^{(j)}$] for each label, which weighs the hidden state $\hat{h}_t$ associated with each word x_t. The hidden document feature m_j in Equation 5.37 represents the extracted relevant information of the whole document and is the the weighted sum of all hidden states $\hat{h}_t$, where α_{ij} is the normalised attention relevance per word x_t.

$$\begin{aligned} m_j &= \sum_{i=1}^{T} \alpha_{ij} \hat{h}_i \\ \alpha_{ij} &= \frac{e^{W^{(j)} \hat{h}_i + b^{(j)}}}{\sum_{t=1}^{T} e^{W^{(j)} \hat{h}_t + b^{(j)}}} \end{aligned} \tag{5.37}$$

Finally, the document feature m_j is fed into a fully-connected layer in Equation 5.38 for reducing the parameters and adapt to the output size L

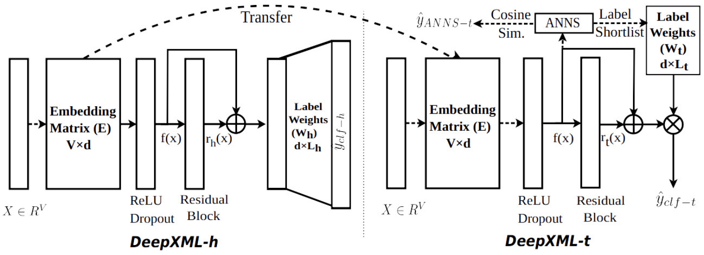
(the number of codes). $[W^{(4)}$, $b^{(4)}]$ are matrices to be learnt. A probability p_i per code i is estimated using the softmax function.

$$
\begin{aligned}
y^{(4)} &= \sigma(W^{(4)} m_j + b^{(4)}) \\
p_i &= \frac{e^{y_i^{(4)}}}{\sum_{k=1}^{L} e^{y_k^{(4)}}}
\end{aligned}
\tag{5.38}
$$

The computational cost is proportional to $\mathcal{O}(B\frac{L}{q^H}NK)$ for training and $\mathcal{O}(B\frac{L}{q^H}(N+K))$ for inference, where B is the batch size, L is the number of labels, N is the number of instances, K is the number of trees, H is the depth of the trees, and q is the number of partitions.

- **DeepXML**. This method divides the learning of the classification task into two connected training stages, one for head labels or another one for tail labels. Figure 5.21 shows an overview of the proposal. Both training are based on the same architecture, transferring part of the parameters learnt for frequent labels to minority ones.

Figure 5.21: *DeepXML, reproduced from Dahiya et al. (2019).*



A first neural network is trained on subsets of head labels (those frequent). The subsets are constituted by performing k-means clustering on the documents. The model is composed of three layers: an embedding layer, residual layer, and fully-connected layer. The embedding representations are randomly initialized and learnt during training. Regarding the residual block, it is a sequence of a linear transformation, batch normalization, ReLU and Dropout functions. The final outputs $\hat{y}_{clf-h}$ are calculated by the full-connected layer as described in Equation 5.39, where x_t is the token t, e_t is the embedding for the token t, T is the total number of tokens, and $[\mathcal{E}$, $W^{(1)}$, $b^{(1)}$, $W^{(2)}$, $b^{(2)}]$ are parameter matrices.

$$
\begin{aligned}
y^{(0)} = ReLU(\sum_{t=1}^{T} x_t \cdot e_t) \quad \text{where} \quad x_t \in X^{(0)}, \quad e_t \in \mathcal{E} \\
y^{(1)} = y^{(0)} + ReLU(W^{(1)}y^{(0)} + b^{(1)}) \\
\hat{y}_{clf-h} = W^{(2)}y^{(1)} + b^{(2)}
\end{aligned}
\tag{5.39}
$$

Two Approximate Nearest Neighbour Search (ANNS) are additionally trained over the label centroids of the head label subsets and over the tail labels. The search is subsequently used both to generate label candidates $\hat{y}_{anns-h}$, or $\hat{y}_{anns-t}$, and to identifying the closest documents for performing a negative sampling generating a shortlist of examples.

A second neural network based on the same layers is trained on tail labels (those infrequent), which exploits the already learned, freezed word embeddings from the first model and fine tunes the residual layer. The transfer learning is conducted by assuming that there is not enought training data in tail for learning quality vectors from scratch.

The final predictions is a combination of all outputs, as described in Equation 5.40, where $\hat{y}_{anns-h}$ and $\hat{y}_{anns-t}$ are the outputs of the ANNS for head and tail labels respectively, $\hat{y}_{clf-h}$ and $\hat{y}_{clf-t}$ are the outputs of both neural networks, and σ is the sigmoid function, and $\beta \in [0, 1]$ is a scaling parameter.

$$
\hat{y} = (1 - \beta) * \sigma([\hat{y}_{anns-h}; \hat{y}_{anns-t}]) + \beta * \sigma([\hat{y}_{clf-h}; \hat{y}_{clf-t}]) \tag{5.40}
$$

The complexity of **DeepXML** can be measured as $\mathcal{O}(NDlogL)$ for training and $\mathcal{O}(D^2 + DlogL)$ for testing, with N being the number of instances, D being the vocabulary size, and L being the number of labels.

- **DECAF**. Similar to Astec, DECAF is designed around three components: a text embedding block, classifier with label-specific parameters, and negative sampler focused on high recall. Text embeddings are jointly learnt for documents and label text in Equation 5.41, where E is the embedding matrix, r is the token vector, R is a residual block, α and β are scaling constants, $\odot$ is the component-wise multiplication, and σ is the sigmoid function. Hence, $\hat{x}_i$ are the document embeddings and $\hat{z}_l^1$ are the label-text embeddings.

$$\begin{aligned} \mathcal{E}(r) = \sigma(\alpha) \odot \hat{r}^{(0)} + \sigma(\beta) \odot (R \cdot ReLU(\hat{r}^{(0)})) \\ \hat{r}^{(0)} = Er \\ \hat{x}_i = ReLU(\mathcal{E}_D(x_i)) \\ \hat{z}_l^1 = \mathcal{E}_L(z_l) \end{aligned} \tag{5.41}$$

The objective function is based on One-vs-All classifiers, with w_l in Equation 5.42 being the output of the classifier for the label l. α_L and β_L are parameter matrices shared across all labels while $\hat{z}_l^2$ is a specific embedding learnt for each label.

$$w_l = \sigma(\alpha_L) \odot \hat{z}_l^1 + \sigma(\beta_L) \odot \hat{z}_l^2 \tag{5.42}$$

Regarding the negative sampler, a label-clustering learner is trained for generating balanced subsets of labels and producing a shortlist with all positive document and the closest negative ones for each instance. in total, the complexity of the whole training would be $\mathcal{O}(NDlog(L))$, with N being the number of instances, D being the vocabulary size, and L being the number of labels.

The final output scores are a combination of the values from classifiers and clustering, with a cost of $\mathcal{O}(Dlog(L))$.

5.4.2 Experimentation

We have implemented the described methods on the long EHRs from the sample of the HUFA corpus described in Section5.3.3. This sample comprises 6,438 different codes in the training data set, with a maximum document length of 105,599 words, and a vocabulary of 49,095, so that the implemented model with the longest training and prediction times (**B-LSTM**) will have to process on the order of 10^{13} parameters ($\mathcal{O}(TDL)$) to predict the codes of each EHR during the experimentation. The other methods handle fewer parameters and shorter processing times. The configurations and results achieved in each case are detailed below.

Experimental settings

Although multiple settings have been explored by varying the document representations and hyperparameters of each model, only the best solution per method maximising Precision at K is presented as the purpose of this analysis is to compare the performance between the proposals applied to the ICD coding for large EHRs. The best settings of each model used in the experimentation are described below.

- **Unsupervised approaches**:
  - **SIM**. The same configuration as described in Section 4.4.1 is used.
- **Standard approaches based on label independency**:
  - **B-SVM**. The same parameters as those described in Section 5.3.3 have been used.
  - **B-MLP**. The number of layers L has been set to 4 and the number of neurons per layer h has been set to 80. A ReLU activation function $\big(\sigma(a_n^L) = max(0, a_n^L)\big)$, Adam optimisation algorithm (Kingma and Ba, 2014), and a binary cross-entropy loss function have been applied with 20 iterations. The optimisation involves an adaptive learning rate ranging from 0.01 to 0.03, momentum of 0.9, β_1 of 0.9, β_2 of 0.999, and ϵ of $1e-8$.
  - **B-AdaBoost**. No external weight has been used in the objective function to reduce the relevance of classifiers after each iteration. A total number of 50 trees have been ensembled.
  - **B-LSTM**. The network is fed with batches of 16 EHRs converted into word sequences. Pre-trained word embeddings are not used, but are learned during training. The embedding dimensionality is set to 300 in the first layer. A single LSTM layer with a size of 50 hidden states and a dropout mechanism of 0.3 has been used. Finally, the output size is set to 2 for getting the probabilities of the presence or absence of each code. The same Adam optimisation as described in the setting **B-MLP** is applied over 20 iterations.
  - **RCNN**. The documents are also transformed into word sequences, producing batches of size 16. Pre-trained word embeddings are not used, but are learned during training. A bidirectional LSTM layer with 300 hidden states and a dropout of 0.3 has been defined. Finally, an output size of 6,438 (the number of labels) is fixed. All parameters are learnt with 20 iterations, the BCE loss and the Adam optimizer described in the setting **B-MLP**.
- **Standard approaches based on label dependency**:
  - **KNN**. The same setting as described in Section 4.3.3 is used.
  - **D-LDA**. The input are the documents in a BoW format. The following values have been set for the parameters with which to model the label probabilities: $\alpha = 50$, $\beta_{\mathcal{W}} = 0.01$, $\eta = 1$, $\gamma = 50$, and $\beta_{\mathcal{L}} = 0.01$. In addition, the number of topics, iterations, and chains have been set to 300, 500, and 5 respectively.
- **XMTC approaches based on label independency**:

  - **B-XGBoost**. The regularization parameters λ and γ are set to 1 and 0.001 respectively. In turn, the maximum depth of trees is fixed in 6 levels.
  - **XML-CNN**. As in other proposals, the EHRs are converted into word sequences grouped in batches of 16 for the embedding layer, which has been set to a dimension of 300. A single layer of convolutional filters with sizes 2, 4, and 8 (h_q) has been defined. In turn, the training is conducted by a number of 128 (p) for the features captured by the filter and a dropout rate of 0.5.

- **XMTC approaches based on label embeddings**:
  - **SLEEC**. 20 learners have been trained over 150 clusters and a label dimensionality of 100. A number of 15 nearest neighbors has been used. Finally, both regularization parameters λ and μ have been set to 1.
  - **AnnexML**. 15 learners have been trained on 50 clusters producing a label dimensionality of 50. In turn, the scaling parameter γ has been set to 10.

- **XMTC approaches based on PLT**:
  - **FastXML**. 50 trees have been trained on TF-IDF vectors with the hyperparameters $C_\delta = 1$ and $C_r = 1$.
  - **Parabel**. This setting uses the averaged TF-IDF vectors for representing the 6,438 labels, which are splitted into binary groups recursively. A total of 15 trees have been used.
  - **Bonsai**. Similarly, the labels are represented by the associated TF-IDF vectors combined with sparse label co-ocurrence vectors. 15 trees of 3 depth levels are trained by generating a maximum of 100 clusters per node (K).
  - **AttentionXML**. A total of 15 trees with 2 levels of depth have been trained, clustering 256 leaves with 25 codes each. Clustering is performed with the TF-IDF vectors. For each tree, two models are trained, one on the 256 intermediate candidates, and the other on the 10 most probable nodes for each instance. As for the network, the embedding layer is defined with a dimension of $49,095 \times 300$ (vocabulary size × embedding dimensionality) and fed with a batch of 16 documents. A single bi-LSTM layer with 512 units and a dropout of 0.5 has been used, followed by two fully-connected layers of sizes 512 and 256. All parameters are learnt with 10 iterations, the BCE loss and the Adam optimizer described in the setting **B-MLP**.
  - **DeepXML**. The data set is organised in batches of 64 EHRs. The embedding dimensionality is set to 300 and the 30 most frequent codes have been

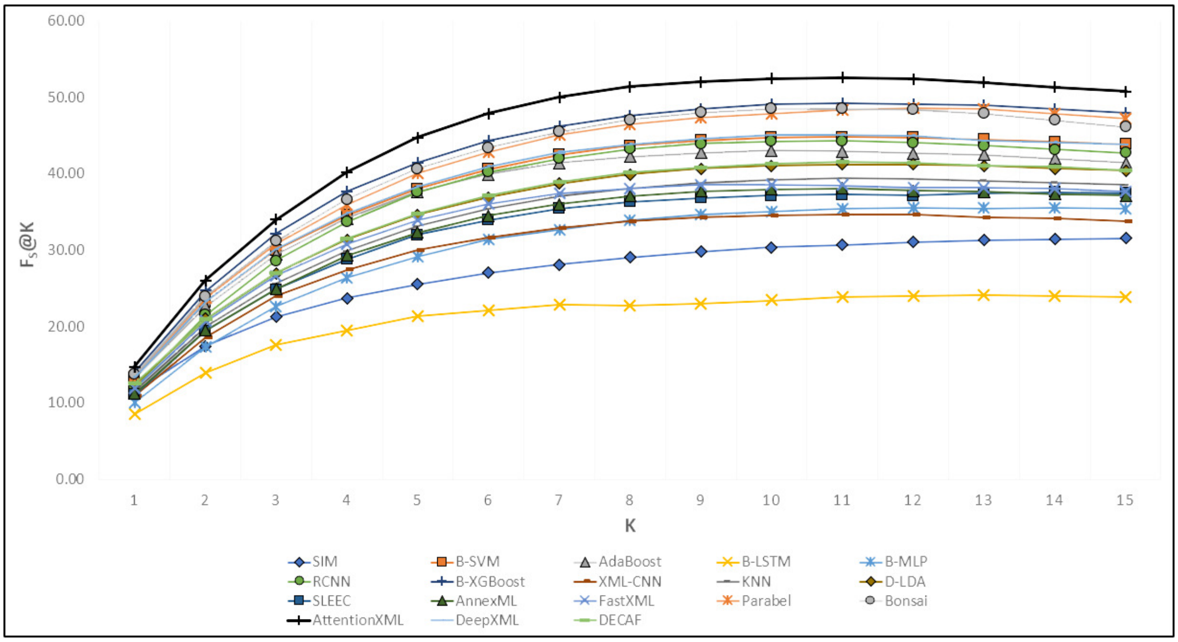
selected as head labels (for the surrogate task). Clustering of the instances is performed on the TF-IDF vectors. Finally, the scaling parameter β is set to 0.6 for combining the surrogate and extreme outputs. All parameters are learnt with BCE loss and the Adam optimizer described in the setting **B-MLP**.

- **DECAF**. Batches of 16 documents have been used, setting the embedding dimension to 512, and the scaling constants $alpha$ and $beta$ to 0.55 and 1.5 respectively. The classifer comprises 512 hidden states with a dropout of 0.5 and 0.2 for the surrogate and extreme tasks. Clustering has been performed using Parabel on the TF-IDF vectors, with 3 trees, for the negative sampler. A logistic loss and the Adam optimizer described in the setting **B-MLP** have been used for training all parameters in 20 iterations each network.

Results

It is worth noting that extreme methods are typically evaluated with ranking metrics; this is one of the reasons for using such a type of metric in all the evaluations during this research. In this line, Figure 5.22 shows an overview of the evolution of F_S scores for the implemented methods as a function of the variation of K values. As one can notice, the best performances are reached at around $K = 10$, which is in line with the average number of codes per EHR.

Figure 5.22: *F_S score of methods at different values of K.*



All metrics described in Section 3.4 have been calculated for each of the settings. The overall results of the methods based on label independence are shown in Table 5.6, while the scores of the methods based on co-dependencies are gathered in Table 5.7. In addition, the relative estimated processing times of each method for training and prediction on the same machine have been included, with the unit being the prediction computational time of **FastXML** (the fastest method).

With respect to the former, the ML OvR methods (**B-SVM**, **B-AdaBoost**, **B-MLP**, **B-LSTM**, and **B-XGBoost**) achieve significantly higher scores, with an average improvement of 32% for micro-averaged F. **XGBoost** achieves better micro-averaged, $nDCG$, and training time values, followed by **B-SVM** and **AdaBoost**. If one pay attention to the representation, the BoW methods achieve better performance than those based on vector representations, such as **B-MLP** (micro-averaged F of 34.55)

		Label independence							
		Unsupervised	**Standard**					**XMTC**	
		SIM	**B-SVM**	**B-AdaBoost**	**B-MLP**	**B-LSTM**	**RCNN**	**B-XGBoost**	**XML-CNN**
Micro	P	31.49	38.11	36.75	35.29	15.08	23.99	**42.27**	24.99
	R	24.66	34.7	34.88	33.84	14.46	23.17	**40.02**	23.97
	F	27.66	36.32	35.79	34.55	14.76	23.57	**41.12**	24.47
	P_S	39.25	46.9	45.37	43.96	23.95	35.65	**50.38**	35.26
	R_S	30.89	42.87	43.13	42.15	22.97	34.57	**47.81**	33.82
	F_S	34.57	44.8	44.22	43.04	23.45	35.10	**49.06**	34.52
	PSP	31.05	32.62	36.66	31.69	11.84	20.83	**41.11**	22.44
	PSR	28.71	25.69	25.12	24.05	6.53	17.73	**29.74**	14.41
	PSF	29.84	28.74	29.81	27.35	8.42	19.15	**34.51**	17.55
	PSP_S	38.50	42.22	46.56	40.88	20.41	33.18	**50.13**	32.93
	PSR_S	35.89	33.95	32.72	31.70	11.68	28.71	**37.22**	21.92
	PSF_S	37.15	37.64	38.43	35.71	14.85	30.78	**42.72**	26.32
Macro	P	**38.71**	9.26	9.61	5.23	0.14	7.46	10.48	1.50
	R	**43.27**	9.90	7.41	5.78	0.51	7.98	9.95	1.99
	F	**40.86**	9.57	8.37	5.49	0.22	7.71	10.21	1.71
	P_S	**41.04**	13.51	13.52	7.40	0.22	13.18	13.87	2.70
	R_S	**45.47**	13.87	10.05	7.69	0.61	13.21	12.54	2.88
	F_S	**43.14**	13.69	11.53	7.54	0.33	13.20	13.17	2.78
	PSP	**39.56**	7.21	7.78	3.72	0.06	6.46	8.57	0.91
	PSR	**45.18**	7.98	5.74	4.08	0.19	6.87	7.96	1.15
	PSF	**42.19**	7.58	6.61	3.89	0.09	6.66	8.25	1.02
	PSP_S	**41.94**	10.8	11.09	5.40	0.10	11.52	11.50	1.70
	PSR_S	**47.59**	11.43	7.94	5.55	0.28	11.42	10.20	1.72
	PSF_S	**44.59**	11.11	9.26	5.47	0.15	11.47	10.81	1.71
Order	$nDCG$	66.67	76.73	74.51	76.26	51.19	61.66	**80.80**	65.76
	$nDCG_S$	69.40	82.86	81.44	82.56	66.45	72.78	**85.04**	75.85
	$PSnDCG$	66.71	70.64	71.09	69.33	48.79	60.11	**73.48**	61.38
	$PSnDCG_S$	69.39	76.38	**78.66**	75.06	61.85	71.03	77.38	70.37
Complexity	Training time	113,000	104,000	**43,900**	271,000	891,000	138,000	45,500	648,000
	Test time	457	1,180	691	636	4,890	**50**	681	2,680

Table 5.6: *Results of CIE-10-ES predictions for each method assuming label independency.*

compared to **B-LSTM** (micro-averaged F of 14.76). Indeed, simpler algorithms such as linear or decision tree based methods (**B-SVM**, **B-AdaBoost**, and **B-XGBoost**) outperform deep learning models (**B-MLP**, **B-LSTM**, **B-RCNN**, and **XML-CNN**), which is probably due to the greater reliance on large data volumes of deep learning algorithms combined with the relative scarcity of examples for minority (tail) codes. More complex representations or algorithms involve a larger number of parameters that need to be learned (factor that increases computational complexity), especially considering the length of the EHRs, which means slower convergence in learning, perhaps not enough for less represented codes. Such fact is reinforced by the significant differences in the training times (65,000 versus 487,00) and macro-averaged scores for exact matching (9.38 versus 3.78 for macro-averaged F). In contrast, it should be noted that the **RCNN** model is apparently able to capture low-level patterns reaching the best partial matches in the minority codes, with the highest macro-averaged scores P_S, R_S, and F_S (13.20 on average). In addition, **IR** does not produce remarkable micro-averaged scores but has by far the highest macro-averaged scores, about 30% F_S above the second one (**B-XGBoost**). As it is not a data-driven method, it has considerably more coverage and gets more matches on tail labels.

As for the methods based on label co-dependencies, PLT-based methods outperform the other proposals with an increase of 27% and 14% for micro-averaged F and F_S respectively, and 61% and 57% for their macro-averaged counterparts. In particular, **AttentionXML** achieves the highest micro-averaged (44.62 in F) and $nDCG$ (83.80) scores, followed by **Bonsai** with which the values drop by 12% on exact matchings and 8% on partial matchings. Comparing PLT-based methods, although the simpler BoW methods such as **Parabel** and **Bonsai** are generally faster to train (41 on average) and outperform vector-based methods such as **DeepXML** and **DECAF** in micro-averaged scores, **AttentionXML** is able to effectively use word embeddings while considerably reducing the number of internal parameters by focusing only on the relevant information via the attention layer. Even though **DeepXML** and **DECAF** are specially designed for short texts, the former reaches the highest macro-averaged values by transferring learning from the head codes to the tail ones (11.82 in F). Label embedding-based methods perform worst on all metrics, while the non-data-driven method **SIM** and the generative model **D-LDA** does not stand out either.

For a better understanding of the behaviour of the algorithms on predicting head and tail codes, the F_S results from Tables 5.6 and 5.7 have been broken down into 8 groups of codes with different frequencies, all with the same total number of instances in the training data set. The distribution is shown in Figures 5.23 and 5.24.

Regarding methods based on independent labels, **B-XGBoost** achieves the top results for both head and tail codes, with **B-MLP** achieving the highest values in the first group and **IR** reaching the best score in the last group. In general, **IR** provides more or less the same performance at different frequencies. **B-SVM** is also a solid

		Label dependence									
		Standard		XMTC							
				Label embeddings		PLT					
		KNN	D-LDA	SLEEC	AnnexML	FastXML	Parabel	Bonsai	AttentionXML	DeepXML	DECAF
Micro	P	28.90	32.20	27.98	28.65	30.23	40.74	41.53	**46.35**	37.99	33.24
	R	25.44	30.65	26.28	27.13	28.65	37.46	38.18	**43.01**	33.74	29.81
	F	27.06	31.40	27.10	27.87	29.42	39.03	39.78	**44.62**	35.74	31.43
	P_S	37.73	42.07	38.30	39.12	39.64	49.87	50.54	**54.47**	47.73	43.57
	R_S	35.60	40.12	36.10	37.09	37.62	46.02	46.60	**50.67**	42.67	39.30
	F_S	36.63	41.07	37.17	38.08	38.60	47.87	48.49	**52.50**	45.06	41.33
	PSP	26.46	32.24	27.83	27.76	30.02	38.93	39.49	**41.05**	33.93	30.36
	PSR	18.73	21.21	17.50	17.94	18.33	27.58	28.25	**31.13**	25.72	22.45
	PSF	21.93	25.59	21.49	21.79	22.76	32.29	32.94	**35.41**	29.26	25.81
	PSP_S	34.00	42.82	38.49	38.56	39.52	48.45	48.90	49.34	43.65	40.90
	PSR_S	26.29	29.22	25.25	25.92	25.15	35.55	36.17	**38.64**	34.29	31.32
	PSF_S	29.65	34.74	30.50	31.00	30.74	41.01	41.58	**43.34**	38.41	35.47
Macro	P	6.15	6.49	4.64	3.47	3.41	10.55	10.76	8.54	**10.93**	9.27
	R	7.78	6.71	6.21	5.71	5.13	11.41	11.64	10.88	**12.87**	11.21
	F	6.87	6.60	5.31	4.32	4.09	10.96	11.18	9.57	**11.82**	10.15
	P_S	8.20	9.03	6.53	5.06	4.45	13.77	14.07	10.90	**14.81**	13.34
	R_S	9.27	8.47	7.54	6.76	5.99	13.93	14.20	12.69	**16.09**	14.67
	F_S	8.70	8.74	6.99	5.79	5.11	13.85	14.13	11.73	**15.42**	13.97
	PSP	5.18	5.18	3.82	2.56	2.69	8.59	8.80	6.67	**9.12**	7.73
	PSR	6.32	5.37	5.20	4.40	4.08	9.47	9.65	8.61	**11.10**	9.65
	PSF	5.69	5.27	4.41	3.24	3.24	9.01	9.20	7.52	**10.01**	8.59
	PSP_S	6.94	7.23	5.35	3.76	3.50	11.28	11.59	8.62	**12.40**	11.19
	PSR_S	8.16	6.81	6.29	5.20	4.78	11.60	11.82	10.13	**13.89**	12.67
	PSF_S	7.50	7.02	5.78	4.36	4.04	11.44	11.70	9.31	**13.10**	11.88
Order	$nDCG$	61.77	72.19	68.31	68.37	72.55	78.55	79.40	**83.80**	76.95	71.45
	$nDCG_S$	68.79	80.75	76.65	76.21	78.50	83.25	83.92	**87.39**	82.43	78.38
	$PSnDCG$	58.48	68.70	63.43	63.25	66.32	70.39	71.44	**73.34**	70.30	67.53
	$PSnDCG_S$	64.89	**77.20**	70.36	69.47	71.09	74.21	75.21	76.50	74.85	73.61
Complexity	Training time	27	36,800	10,600	1,760	**18.1**	40	66	60,400	11,000	88,000
	Test time	9.57	197	4.57	2.14	**1**	3.86	6.71	85.4	1.29	17.1

Table 5.7: *Results of CIE-10-ES predictions for each method assuming label dependency.*

option, especially for groups 3 to 7. As for the methods including label dependencies, **AttentionXML** outperforms all methods for head labels, while **DeepXML**, **DECAF**, **Bonsai**, and **Parabel** achieve the highest values for tail labels, in this order. All other methods involving dependencies behave in a similar way.

Overall, **AttentionXML** performs the best in the first 4 groups, **Bonsai** (or **Parabel**) and **B-XGBoost** achieve significant results in the intermediate groups, differing in that **Bonsai** involves lower computational complexity, and **IR** and **DeepXML** (or **DECAF**) more successfully recover codes from the last group, noting that **IR** is able to predict zero-shot as well as few-shot codes.

Figure 5.23: *Results of methods from Table 5.6 in micro-averaged F_S score broken down into groups of codes ordered from highest to lowest frequency. All 8 groups have the same impact on the training data set (i.e., they have similar numbers of instances).*

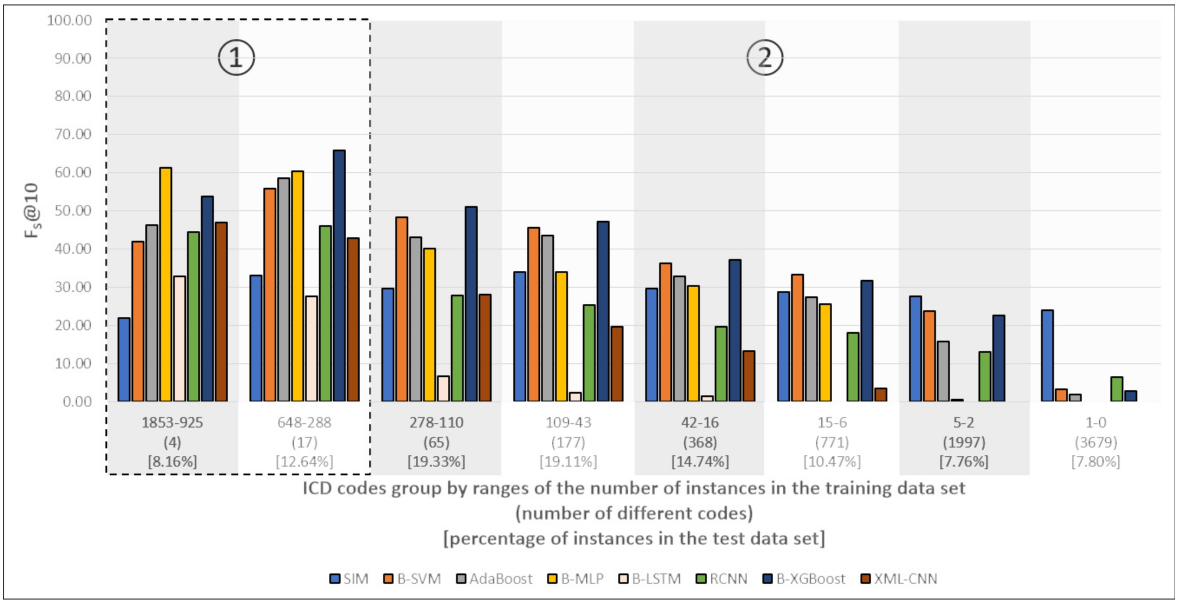
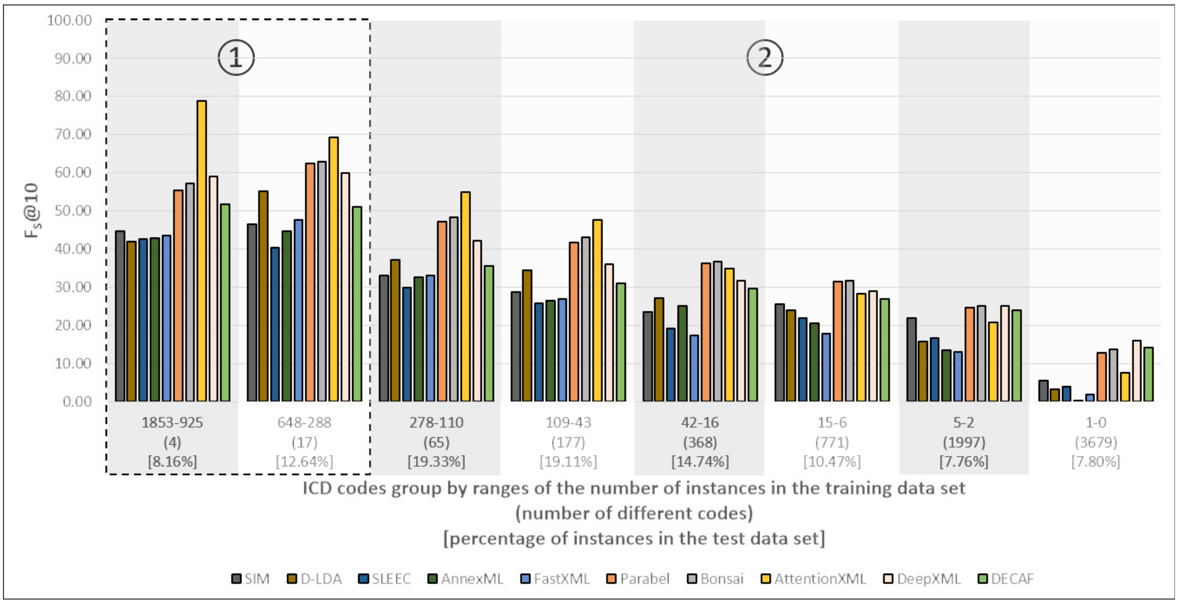


Figure 5.24: *Results of methods from Table 5.7 in micro-averaged F_S score broken down into groups of codes ordered from highest to lowest frequency. All 8 groups have the same impact on the training data set (i.e., they have similar numbers of instances).*



5.5 Discussion and concluding remarks

Data augmentation techniques are scarce in the SOTA of the clinical domain. For this reason, we have explored multiple data augmentation techniques adapted to the domain to generate synthetic data for under-represented codes, and thus reduce the imbalance of the data distribution. In this line, the application of lexical substitutions based on structured knowledge and MT techniques have been proposed. Surprisingly, there are hardly any proposals that approach coding from an extreme perspective even though the task fits perfectly into this domain. As far as we know, our publication (Almagro et al., 2020) describes the first work approaching icd coding with XMTC methods. This was followed only by the two works proposed by Chalkidis et al. (2020) and Zhang, Liu, and Razavian (2020). For that reason, we have conducted a comparative study of different families of XMTC algorithms and other conventional types to contextualise the performance of each one and identify the possible differences. The performance of the proposed data augmentation and XMTC methods is discussed below.

Data augmentation performance The extension of the training data set has been explored by means of in-domain permutations based on SNOMED CT and non-domain modifications using the MT Marian model. The impact of imbalance on inference has been effectively reduced using SNOMED CT terminology, which succeeds in improving the representativeness of minority codes by increasing the lexical diversity of the clinical vocabulary. Instead, the paraphrases introduced by the MT models have a negative impact on learning, at least for minority codes. Consequently, data augmentation using lexical substitution based on clinical ontologies is one of the techniques that increases the predictive ability of supervised methods for under-represented codes without harming the overall performance of the system. This answers part of the Research Question "*Which techniques can increase the predictive capacity of ICD-10 codes with fewer instances while improving overall system performance? How and how much can the computational complexity of the task be reduced?*" (RQ 3).
Whereas data augmentation can significantly improve underrepresented codes, they have been explored in a OvR approach given the volume of data involved and the label interdependencies that prevent increasing the presence of one code without increasing the presence of others. The increase in data has a very significant negative effect on scalability, but it could be solved with the use of XMTC algorithms. This combination is explored in Chapter 7 along with other techniques.

XMTC performance As discussed in Section 5.4.2, methods based on different mechanisms show non-overlaping behaviours depending on the code frequency. None of them outperforms in the whole spectrum, which could indicate the possibility of fusing some methods to fill performance gaps by producing a piecewise inference function (a proposal is described in Chapter 7).

Hence, proposals such as the unsupervised methods enriched with structured data outperforms all methods on few-shot codes, for which examples is lacking. Alternatively, tail codes are best inferred by **DeepXML** and **DECAF** methods, which apply transfer learning techniques to improve convergence during training. Following this line, the next chapter (Chapter 6) aims to improve the prediction of not-so-specific codes by exploring different transfer learning techniques in the ICD coding domain. Furthermore, PLT-based approaches using BoW representations exhibit better performance for medium-frequency codes. Finally, the best models both assuming independent labels and exploiting co-dependencies for bulk (frequent) codes are **XGBoost** and **AttentionXML**, which rely on capturing specific features per label, either using independent training or attention layers.

In general, the best results have been achieved by PLT-based methods, which rely on unsupervised techniques such as k-means clustering to contribute information about the inherent structure of the data to learning. Such information seems crucial to promote less frequent codes. In addition, it also seems essential to include label focus mechanisms in training to facilitate the identification of code-specific patterns to improve the modelling, which seems to be decisive in long text tasks to discriminate relevant information among a large amount of data. So in response to Research Question 3, in addition to data augmentation methods, label-specific features and XMTC algorithms (especially those PLT-based) improve the inference of minority codes while increasing overall performance.

As for the computation time, algorithms based on label independence take on average 13 and 43 times longer to train and predict compared to those algorithms based on label dependence. Hence, we can state that the latter handle coding better in terms of efficiency. In particular, PLT-based algorithms stand out for their reduced prediction time, which is more suitable for a possible real-world application. Therefore, going back to Research Question 3, we claim that approaching coding as the classification of subsets in a PLT-based algorithm reduces computational complexity, both in terms of model volume and inference time. If we focus on the latter parameter (decisive in the domain), reductions of more than 600 relative units are achieved.

CHAPTER

6

EXPLOITING TRANSFER LEARNING

Content

This chapter presents different transfer learning approaches with the aim of improving generalisation in learning code classification. After a brief review of the types of transfer learning in NLP and the most widespread methods, instance-based, feature-based, and parameter-based transfer techniques are explored.

The following objectives are planned:

- Explore cross-lingual methods for enriching ICD codes with datasets in other languages.
- Generate contextualised and non-contextualised word vectors for the clinical domain.
- Evaluate the application of in-domain and general embeddings in ICD coding.
- Explore the use of language models for ICD coding.
- Include hierarchical information via parameter-based transfer learning.

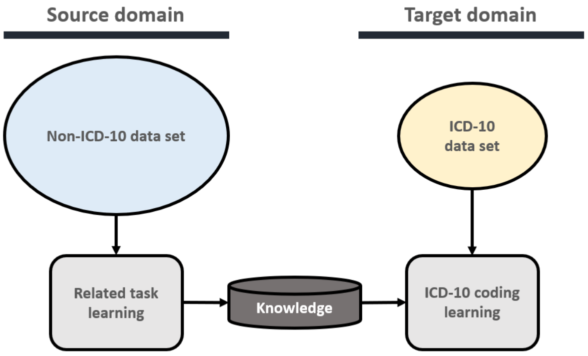
6.1 Introduction

ICD codes are more general than the entities described in the EHRs as the ICD purpose is to group multiple concepts for statistical analysis, as mentioned in Section 1.2.2. In fact, final codes at different hierarchical levels coexist in the standard, which implies varying degrees of abstraction. The lower specificity of codes with fewer characters is usually associated with a higher lexical diversity, as those codes encompass a larger number of clinical events. Consequently, more examples are needed for appropriate generalisation during learning unless prior knowledge of semantics is introduced.

In turn, learning complex tasks from scratch requires large amounts of labeled data and entails high computational cost, but huge data resources are not always available and computational costs may exceed hardware capabilities. In addition to the fact that collecting data in the clinical domain is not easy, the inherent biases of ICD-10 coding cause an extreme imbalance, resulting in a poor sample of instances for many codes. Besides, individual characterisation of the vast number of ICD-10 diagnoses and procedures usually leads to unwieldy memory space and execution time. Although such issues have been tackled in Chapters 4 and 5, can also be addressed somehow introducing more general knowledge in learning ICD-10 coding.

Learning from scratch can be avoided by acquiring general knowledge from other related tasks. This process, which is illustrated in Figure 6.1, is called transfer learning. First, traditional ML approaches are applied to solve a more generic task (source task) for which more data is available such as clinical language modelling; then, the generated patterns are used to improve ICD-10 coding learning (target task). The main idea of transfer learning is to use the knowledge based on the common

Figure 6.1: *Transfer learning outline for ICD coding.*



elements with the previously-learned tasks as a starting point to focus only on the new elements to be learned for the target task, which accelerates the convergence. For example, clinical language models capture domain-specific syntactic structures, medical expressions, and particular clinical vocabulary, so that they can be used to enrich the representation of documents. This is expected to enhance and speed up generalisation, thus reducing the number of examples of the target task needed for learning.

This chapter focuses on transfer learning techniques applied to the Spanish ICD-10 coding with the objective of accelerating generalisation during learning. Thus, the chapter aims to identify the best methods to enhance inference for answering the Research Question "*Which transfer learning methods are easily applicable to ICD-10 coding and which ones are most effective in improving inference?*" (**RQ 4**). Although there are different ways of transferring knowledge from one task to another, as will be discussed in Section 6.2, we have focused on the mechanisms that best fit the task: instance-based, feature-based and parameter-based methods.

Instance-based transfer Instance methods rely on the use of instances of very similar source tasks to augment the training data using some minor transformation. In line with this idea, we have exploit the information shared in the data sets from the multilingual *Causes of Death* corpus in order to investigate the feasibility of extending the training data with others collections with different marginal probability distributions. The main idea is to transform the source instances with MT techniques and weight them during training.

Feature-based transfer Those methods consists of using source tasks to identify or generate appropriate feature representations that minimise divergence and errors in the target task. We have explored the transfer of information learned in general-purpose traditional and contextualised word embeddings on the HUFA corpus. Given the scarcity of Spanish word embeddings in the clinical domain, we have also explored the generation from Spanish EHRs and scientific literature.

Parameter-based transfer Such methods are based on the assumption that related task models share parameters or prior distribution of hyperparameters, e.g., neural networks can be pre-initialised with the internal states of other models. We have explored the pre-initialisation based on LM and the exploitation of inter-level information by reusing the internal parameters of models applied to the classification of non-final codes, such as chapters or groups.

6.2 Related Works

This section reviews the transfer learning methods proposed in the SOTA, with a particular focus on the clinical domain. In this way, we intend to identify research gaps on which to elaborate proposals, in addition to providing a context for the related methods described in Chapter 2.

6.2.1 Introduction

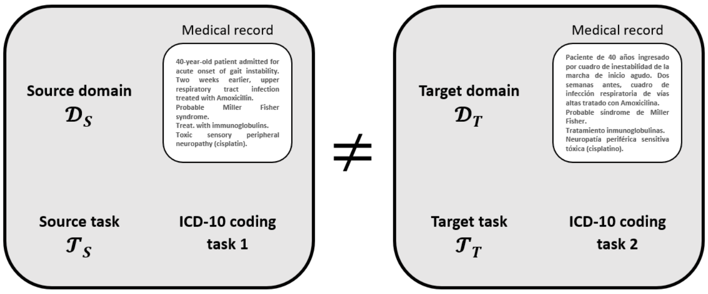
Machine learning methods have traditionally assumed the same distribution of features for the training data and those to be predicted, so any differences typically result in degraded performance (Shimodaira, 2000). Transfer learning is motivated by the intention to generalise learning beyond specific tasks, and even domains, in order to tackle tasks with less available data. The idea is to pass on processed information from other, typically more general, task (source task) to the learning of a new one (target task) so that common patterns are reused. Hence, the more similar the source and target tasks are, the less new elements need to be learnt. For example, a NER task for all medical entities exploit the same language as other tasks including the classification of medical reports based on specialisations, so both share common low-level features such as specific clinical syntactic and morphological patterns. Therefore, the learning of the first task could theoretically be used to solve the other.

The following is a formal definition based on the notations proposed by Pan and Yang (2009) and Weiss, Khoshgoftaar, and Wang (2016) for the transfer learning of a classification task. A domain $\mathcal{D}$ is defined as $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ is the feature space, $P(X)$ is the marginal probability distribution, and $X = \{x_1, ..., x_n\} \in \mathcal{X}$ is the set of feature vectors representing the instances. Therefore, two domains are different when the feature spaces are not equal $(\mathcal{X}_1 \neq \mathcal{X}_2)$ or the marginal probability distributions differ $(P(X_1) \neq P(X_2))$. In turn, a task $\mathcal{T}$ within the domain $\mathcal{D}$ is defined as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, where $\mathcal{Y}$ is the label space and $f(\cdot)$ is the predictive function. Such a function directly depends on the conditional probability distribution

$P(Y|X)$ as it is inferred from the label and feature vector pairs $\{x_i, y_i\}$, where X is the set of feature vectors, Y is the set of label vectors, x_i is the feature vector representing an instance i, and y_i is the label vector associated with the same instance i. Similarly, two tasks are different if they do not share the same label space $(\mathcal{Y}_1 \neq \mathcal{Y}_2)$ or the same conditional probability distribution $(P(Y_1|X_2) \neq P(Y_1|X_2))$.

For the purpose of this thesis, the target task is the ICD-10 coding of medical records. If we assume for example a traditional BoW approach based on word frequencies, $\mathcal{X}$ would be the space comprising the word frequency vectors for all the stated clinical words, while $P(X)$ would be constituted by the set of individual probabilities for each word frequency vector. In turn, $\mathcal{Y}$ would be the space containing the entire set of potential codes and $f(\cdot)$ would be the coding fuction. Furthermore, x_i and y_i would be the word frequency vector representing a document i and its associated code vector respectively. Finally, $P(Y|X)$ would composed of the set of probabilities for each code vector given a word frequency vector.

Figure 6.2: *Example of transfer learning for ICD-10 coding.*


Source domain
$\mathcal{D}_S$
Medical record
40-year-old patient admitted for acute onset of gait instability. Two weeks earlier, upper respiratory tract infection treated with Amoxicillin.
Probable Miller Fisher syndrome.
Treat. with immunoglobulins.
Toxic sensory peripheral neuropathy (cisplatin).
Source task
$\mathcal{T}_S$
ICD-10 coding task 1
≠
Target domain
$\mathcal{D}_T$
Medical record
Paciente de 40 años ingresado por cuadro de inestabilidad de la marcha de inicio agudo. Dos semanas antes, cuadro de infección respiratoria de vías altas tratado con Amoxicilina.
Probable síndrome de Miller Fisher.
Tratamiento inmunoglobulinas.
Neuropatía periférica sensitiva tóxica (cisplatino).
Target task
$\mathcal{T}_T$
ICD-10 coding task 2

Once the concepts to be handled have been defined and explained, we can formally define transfer learning as the process of exploiting information from the source task $\mathcal{T}_S$ and domain $\mathcal{D}_S$ to improve the predictive function $f_T(\cdot)$ of the target task $\mathcal{T}_T$ and domain $\mathcal{D}_T$, while satisfying $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$. Figure 6.2 has been used to illustrate the transfer process in two ICD-10 coding tasks. The multiple possible scenarios are presented below:

- $\mathcal{X}_S \neq \mathcal{X}_T$: source and target feature spaces are different. It would be the case if the source task is the ICD-10 coding of medical records in a different language. Both languages may share some kind of information related to grammar and syntax that can be exploited.

- $P(X_S) \neq P(X_T)$: marginal probability distributions are not equal. An example would be to code records from a particular health centre and leverage information from another coding task based on scientific literature. It is the same language $(\mathcal{X}_S = \mathcal{X}_T)$ but the documents are different as they have separate audiences, which implies differences in word frequency, i.e., frequency feature biases. Although generic words such as "*patient*" are expected to appear in both tasks with the same frequency, domain-specific words such as "*HTN*"[1] will differ in their distribution, leading to different feature vectors.

- $\mathcal{Y}_S \neq \mathcal{Y}_T$: different label spaces. An example of non-matching label spaces would be a source coding task based on the ICD-O-3, which is similar to the target task but uses different codes.
- $P(Y_S|X_S) \neq P(Y_T|X_T)$: the conditional probability distributions of labels are not the same. A case of different conditional probability distributions would be to exploit knowledge from coding veterinary records. Despite sharing the same language $(\mathcal{X}_S = \mathcal{X}_T)$, the same codes $(\mathcal{Y}_S = \mathcal{Y}_T)$, and maybe even approximately the same frequency of words $\left(P(X_S) = P(X_T)\right)$, individual words may have different meanings. An example would be "*parrot beak*", which can refer to the animal's body part in a veterinary record or to a specific disorder in the case of humans. In this way, the same document representations would lead to different codes. The same applies to try to capitalise on learning between corpora annotated with different ICD criteria. As discussed in Section 3.2, the same document would result in different code sets in each collection due to discrepancies between annotation guidelines.

Introducing general knowledge into ML models involves multiple benefits according to Olivas et al. (2009). As illustrated by the authors in Figure 6.3, transfer learning reaches higher start, slope, and asymptote for the learning convergence of the target task, which means better initial performances, rates of improvement and final performances respectively. Such faster convergence is associated with the lower data requirements and lower computational cost mentioned above. Moreover, transfer learning implies usability as the same model generated from vast volumes of data can be reused for multiple purposes. But the property of most interest in this case is undoubtedly the ability to achieve better generalisation in biased data, which is expected to improve the performance for supervised ICD-10 coding models.

Despite all these potential advantages, transfer learning does not always improve performances. A negative transfer can be produced by introducing more noise than improvement if source and target domains are not well-related. Therefore, it is essential to identify the task conditions in order to properly choose which transferable information is useful for the target task, the best way to transfer it, and how to avoid transferring information that degrades the performance of the approach. Traditionally, three types of transfer learning scenarios were established according to the availability of labelled data for both the source and target tasks, as Pan and Yang (2009) stated:

- Inductive transfer learning: labelled data is available in the target task.

[1] HTN is the acronym for High Blood Pressure

Figure 6.3: *Overview of the benefits of applying transfer learning in the target task.*

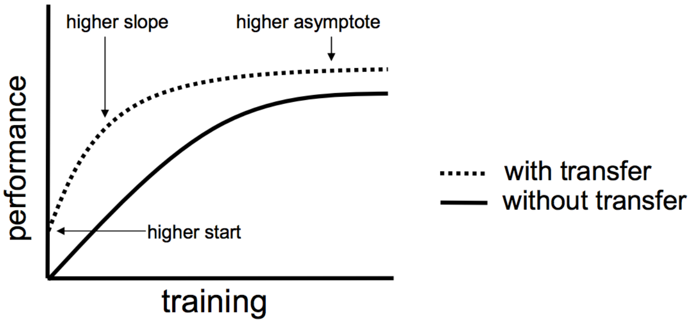
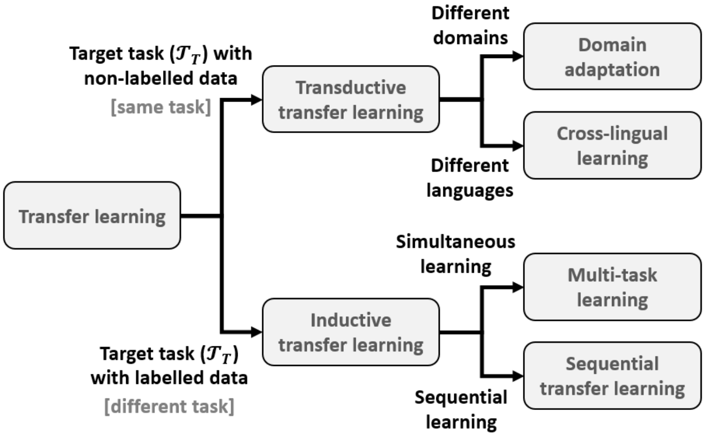


Figure 6.4: *Types of transfer learning methods in NLP. Adaptation of the taxonomy proposed by Ruder (2019).*


Transfer learning
Target task ($\mathcal{T}_T$) with non-labelled data
[same task]
Transductive transfer learning
Different domains
Domain adaptation
Cross-lingual learning
Different languages
Target task ($\mathcal{T}_T$) with labelled data
[different task]
Inductive transfer learning
Simultaneous learning
Multi-task learning
Sequential transfer learning
Sequential learning

- Transductive transfer learning: target tasks without labelled data that leverage source tasks with labelled data.

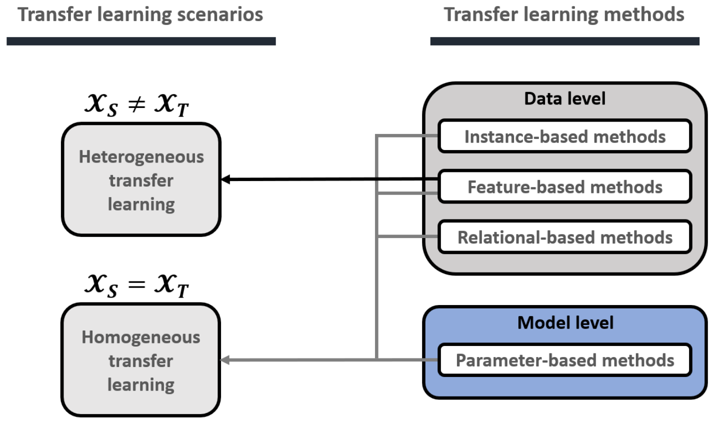
- Unsupervised transfer learning: source and target tasks both without labelled data.

In the case of transfer learning in the NLP area, most of the proposed techniques are transductive or inductive. Figure 6.4 offers an approximate outline of the possible alternatives: domain adaptation, cross-lingual learning, multi-task learning and sequential transfer learning. Domain adaptation methods try to correct the marginal probability distribution differences, cross-lingual learning is focused on dealing with different languages as feature spaces are not equal, multi-task learning attempts to infer more than one goal at a time with the idea of exploiting common information across tasks, and sequential transfer learning exploits knowledge with a sequence of steps.

Labelled data are available for the ICD-10 coding task presented in this thesis, so inductive transfer learning methods have been applied. These methods attempt to capture the set of general patterns that characterise the domain via source tasks and incorporate them somehow in the target task. The acquisition of these patterns is typically unsupervised such as the exploitation of information from language models. We have focused on the relatively recently alternative scenarios outlined by Weiss, Khoshgoftaar, and Wang (2016) in a new, more flexible categorisation: homogeneous and heterogeneous transfer learning. It is a homogeneous scenario if source and target domains share the same feature space $(\mathcal{X}_S = \mathcal{X}_T)$; otherwise, it is a heterogeneous scenario $(\mathcal{X}_S \neq \mathcal{X}_T)$. The goal in homogeneous scenarios is to overcome gaps between data distributions, either by attempting to correct for differences between marginal probability distributions $(P_S(X) \neq P_T(X))$ or discrepancies between conditional probability distributions $(P(Y_S|X_S) \neq P(Y_T|X_T))$. For this purpose, alternative transfer mechanisms can be applied depending on what type of information is intended to be transferred: **instance-based**, **feature-based**, and **relational-based** methods operate at the data level, while **parameter-based** methods transfer the knowledge at the model level. In the case of the heterogeneous scenario, the feature space is different, even non-overlapping, so it first requieres feature-based methods to bridge the gap between feature spaces and then address a new homogeneous scenario. Figure 6.5 provides an overview of the transfer learning methods.

The following subsections provide further details on the transfer learning methods explored in the NLP context. An additional discussion section has been included to provide a brief review of ICD coding proposals based on transfer learning, which constitutes the basis for our proposals described in the following sections.

Figure 6.5: *Overview of the transfer learning scenarios and methods.*


Transfer learning scenarios
Transfer learning methods
$\mathcal{X}_S \neq \mathcal{X}_T$
Heterogeneous transfer learning
$\mathcal{X}_S = \mathcal{X}_T$
Homogeneous transfer learning
Data level
Instance-based methods
Feature-based methods
Relational-based methods
Model level
Parameter-based methods

6.2.2 Instance-based transfer

Sometimes instances of two related tasks are very similar and can be used almost together. Instance-based transfer relies on leveraging the availability of comparable instances within a related task to exploit the information directly or by means of minor prior knowledge-based transformations.

The most popular technique is the weighting of the source domain samples in an attempt to correct for differences in marginal distribution. Weights are typically based on similarity values computed by a binary classifier, which is trained to separate source and target instances. For example, Asgarian et al. (2018) assign two-factor composite soft weights to source samples for two tasks, facial expression recognition and injury prediction. The first factor is a measure of similarity to the target task produced by a regressor, which is trained to distinguish the origin of the instances. The second factor is proportional to the confidence of a predictive model, which is trained on both source and target instances to quantify the effectiveness of the source samples. Another technique is explored by Yao and Doretto (2010), who propose a modification of the Adaptive Boosting algorithm to reduce the negative transfer by iteratively reducing the weight of source instances that do not contribute information to the target task. We have published a proposal that uses MT methods and weights to balance the impact of instances from collections in other languages (Almagro et al., 2019).

Alternatively, multi-instance transfer learning is another common technique, which capitalises on the presence of incomplete instances whose information is at group level (Foulds and Frank, 2010). In particular, a model is trained for classifying sets of instances, called bags, with the aim of subsequently inferring individual-specific labels within groups. For example, Kotzias et al. (2014) propose a classifier of review ratings at document level, which uses an objective function based on label propagation in order to subsequently infer ratings at sentence level. Similarly, Lutz, Pröllochs, and Neumann (2018) explore a Sentiment Analysis sentence-level model for financial news by transferring information from document level to sentence level. The same authors propose a loss function with two terms, one that penalises assigning different labels to similar instances, and other that penalises incorrect associations of group labels.

6.2.3 Feature-based transfer

Semantic representation has historically been explored through two typically opposing paradigms: compositional and distributional principles. The compositional perspective deals with words as discrete symbols interacting by means of rules. Chomsky (2014) proposes methods based on grammars defining the general features of the language. However, the distributive perspective, which exploits low-level linguistic patterns in large volumes of text such as lexical structures and word semantic meanings, has become dominant. Thus, distributional semantic representations based on the prior knowledge embedded in text collections have emerged, in line with the distributional hypothesis stated by Harris (1954): words with similar contexts tend to be similar in meaning. In this way, the semantics of words have often been characterized by contextual distributions, produced by processing all the contexts within large corpora in which the linguistic units are found.

Two types of distributional semantic models can be distinguished (Baroni, Dinu, and Kruszewski, 2014): count-based and predict-based models. The first ones are based on word-context matrices comprising global co-occurrence counts, e.g., Hyperspace Analogue to Language (HAL) (Lund and Burgess, 1996) and Latent Semantic Analysis (LSA) (Landauer, 2007). In contrast, predict-based models are designed to predict words given the context, which is intimately tied to language models. As a result, semantic representations as widespread as GloVe (Pennington, Socher, and Manning, 2014) and word2vec (Mikolov et al., 2013) have been generated. Several studies, such as the one proposed by Baroni, Dinu, and Kruszewski (2014), have shown that embeddings derived from predict-based models reach higher performance in downstream tasks than count-based ones. In fact, such distributional representations have been an ongoing element in the SOTA proposals of recent years. For this reason, we have focused on semantic representations derived from predict-based rather than count-based models.

An overview of the proposed source models for the production of distributional representations is presented below. We also give a brief introduction to the collections available for embedding generation, focusing on Spanish corpora that may be relevant to our problem as the corresponding main experimentation is conducted on the HUFA corpus.

Distributional semantic representations

Distributional semantic representations, commonly known as embeddings, are the most widespread features in most tasks over the last few years, such as entailment (Bowman, Potts, and Manning, 2014), Sentiment Analysis (Socher et al., 2013), summarization (Nallapati et al., 2016), and Question Answering (Seo et al., 2016). These are dense, fixed-length vectors based on linguistic unit co-occurrence statistics. Such vectors originated from early neural language models (Bengio et al., 2003) at the word level, which projected raw word vectors onto embedding layers to reduce dimensionality. Embeddings began to be used as independent features in NLP tasks, as Turian, Ratinov, and Bengio (2010) proposed, becaming popular with works such as the one published by Mikolov et al. (2013), Pennington, Socher, and Manning (2014), and Levy and Goldberg (2014). Mikolov et al. (2013) showed the utility and interpretability of such representations encoding latent syntactic (e.g., grammatical concordance) and semantic features while being subject to explainable mathematical operations, such as subtractions and additions.

The most commonly used embeddings are at word level, as they relatively effectively encapsulate individual and complete meanings while being easy to separate. Such representations are typically generated in context-word association tasks with static vocabularies, where context is interpreted as a set of nearby words. Quality depends directly on both the model and the availability of examples, traditionally comprising one meaning per word. Nevertheless, it should be noted that the non-uniformity of textual collections implies that not all words will be represented equally, so different authors have explored techniques for improving the poor representations of rare words and dealing with the Out-of-Vocabulary (OOV) words. Besides, other levels of linguistic aggregation are sometimes required to include interactions between words, such as multi-term concepts and sentences. Another challenge arises from the strong similarity between antonyms, as opposite words often share the same contexts and are associated with similar vectors, in the same orientation. Finally, the ambiguity of language tipically entails polysemy, which is being ignored by using all contexts to constitute a single meaning. Recent embedding representations tackle this challenge with the use of language models for dynamic vector generation (contextual word embeddings). The three aspects have been detailed below.

Complementing features One way to improve representations while dealing with OOV tokens is to use sub-word information. Such linguistic units are smaller than words, but less fine-grained than characters, and retain some semantic identity like the greco-latin suffixes. Thereby, sub-word units help to increase the information on rare words, and OOV words can be represented by composition.

For this purpose, different data-driven tokenisation methods have been proposed to split words. The BPE technique is explored by Sennrich, Haddow, and Birch (2015b). BPE uses an iterative process to select the most frequent sub-words which constitutes the independent tokens. Other proposals, such as WordPiece (Schuster and Nakajima, 2012) and Unigram Language Model (Kudo, 2018), focus on building a character LM to select the units that maximise the overall likelihood. Alternatively, Bojanowski et al. (2016) explore representing words as the composition of all their possible n-grams and propose a version of the word2vec model adapted for generating in- and out-of-vocabulary word vectors called FastText. Similar proposals have been suggested for handling multiple word senses (Athiwaratkun, Wilson, and Anandkumar, 2018).

Another way to enhance vectors is the addition of the high quality semantic information of words stored in semantic lexicons. These knowledge graphs, such as PPDB (Ganitkevitch, Van Durme, and Callison-Burch, 2013), WordNet (Miller, 1995), FrameNet (Baker, Fillmore, and Lowe, 1998), and Conceptnet (Speer, Chin, and Havasi, 2017), can reduce the tendency of traditional embeddings to mix the specific information on semantic similarity and conceptual association (Hill, Reichart, and Korhonen, 2015). In turn, the use of synonymy, hypernymy, hyponymy, and other relations between words can avoid the alignment of synonyms and antonyms.

The information transfer from semantic lexicons to embeddings can be conducted during training or as a post-processing. Structured knowledge can be integrated in the training by adding a regulariser component that aims to close semantically related vectors in the lexicon (Bian, Gao, and Liu, 2014; Fried and Duh, 2014; Liu et al., 2015; Xu et al., 2014; Yu and Dredze, 2014), or by increasing the co-occurrence matrix through the graph relations (Chang, Yih, and Meek, 2013; Yih, Zweig, and Platt, 2012). As for post-processing methods, Faruqui et al. (2014) propose a function called Retrofitting which minimises the distance composed of the space between the vectors and the adjacent nodes in the lexicon and the space between the vectors and their nearest distributed neighbours. Mrkšić et al. (2016) also introduce a function to maximize distances between antonyms. The method ATTRACT-REPEL proposed by Mrkšić et al. (2017) also modify the non-lexicon vectors neighbouring the target vectors. Finally, Vulić et al. (2018) extend the specialization to all vectors, whether or not included in the lexicon. For this purpose, the method Post-Specialized Word Embeddings generalises the transformations to be performed on the vectors whose words are found in the lexicon and applies them to all trained vectors.

Different aggregation levels Although the distributional hypothesis typically deals with words, the use of these representations have been proposed for different types of utterances, such as characters, chunks, and sentences. Character-level vectors are low-level features, less general than word embeddings, as they focus on capturing task-specific morphological rather than common semantic information. As Peters et al. (2018) mentioned, character embeddings are tipically accompanied by words to reach better performance, such as those approaches proposed by Lample et al. (2016) and Ma and Hovy (2016).

The idea of generating sentence is to obtain fixed-size representations from variable-length pieces of texts. For this purpose, compositional methods have been explored, with the average being a robust baseline. In this line, Wieting et al. (2015) use a supervised model based on paraphrase pairs to average word vectors, while Arora, Liang, and Ma (2017) apply a weighted average modified by Principal Component Analysis (PCA).

Other methods for directly learning high-level representations have also been explored. For example, Le and Mikolov (2014) proposed an architecture based on word2vec but adding paragraph symbols at the beginning of sentences to encode the information of the whole piece. Most of proposal are based on auto-encoders, such as the Skip-Thought vectors (Kiros et al., 2015), which are based on an unsupervised encoder-decoder architecture that tries to reconstruct the surrounding sentences.

Another authors such as Conneau et al. (2017) explore supervised models trained on sentence pairs provided by corpora such as the Stanford Natural Language Inference (SNLI) dataset. Similarly, Subramanian et al. (2018) address multiple training objective, such as MT, natural language inference, and constituency parsing, in order to provide improved multi-task sentence embeddings. Finally, Cer et al. (2018) combine Skip-Thought-style unsupervised approaches and supervised models trained on the SNLI corpus.

Sentence embeddings entail a significant loss of information, as syntactic and semantic information is compressed in such a way that the surface aspects predominate (Adi et al., 2016). Multi-level representations have been found to provide complementary information and, for this reason, embeddings at different levels are commonly concatenated with word embeddings or inputs at intermediate layers. These concatenations, also called hypercolumns, are a common technique to improve performance. An example can be found in the works proposed by Conneau et al. (2017), McCann et al. (2017), Peters et al. (2018), and Wieting and Gimpel (2017).

Contextualisation Words can have different meanings depending on the context in which they are used. Generally, the more frequent the words are, the more ambiguity they convey, and the more meanings they can harbour according to the Principle of Economical Versatility of Words enunciated by Zipf (1950). For this reason, new context-sensitive models generating the vectors dynamically have emerged. Most of source models are based on LM objectives, which predict the probability of words given the preceding, although there are also MT proposals. Anyway, the idea is to freeze source model weights to produce the common intermediate outputs that are expected to be used as task-specific model inputs, significantly improving performance.

Some of the proposals are the contextualised word embeddings described by the internal states of a character-level LM based on a LSTM architecture (Akbik, Blythe, and Vollgraf, 2018), or the Embeddings from Language Models (ELMo) representations produced by a bi-LSTM language model using both character and word embeddings (Peters et al., 2018). In this line, different versions of the Generative Pre-trained Transformer (GPT) model have been proposed by Brown et al. (2020) and Radford et al. (2018, 2019). The three versions rely on the same LM transformer architecture but varying some optimisation techniques and training collections.

All of the above models use the left-to-right or right-to-left predictive method. In contrast, Devlin et al. (2018) introduce a masked LM transformer, which aims to predict randomly deleted tokens in every sentence. Such tokens are based on sub-words instead of words. Bidirectional Encoder Representations from Transformers (BERT) has also been trained on the next sentence prediction to capture relationship between sentences. Besides, BERT-derived transformers have been proposed, such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019b), and ALBERT (Lan et al., 2019). Alternatively, the Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) architecture proposed by Clark et al. (2020) explore the replacement of tokens with alternative words using a generative model instead of only applying a mask. Thus, the objective function of the discriminator model is not to predict words but to distinguish those words that have been generated. This paradigm leads to similar performance with less training.

A MT sequence-to-sequence model with attention mechanisms is introduced by McCann et al. (2017), producing the Context Vectors (CoVe). Lample and Conneau (2019) also explore cross-lingual word embeddings produced by the XLM model, using LM and MT objectives. The experiment conducted by Hill et al. (2017) concludes that embeddings trained with MT encoders outperform those from monolingual encoders (such as model language) on Semantic Textual Similarity (STS) tasks.

All these models have outperformed previous SOTA results in many NLP tasks; however, recent studies such as the one proposed by Tenney et al. (2019) point out that language and translation models improve the modelling of syntactic information compared to non-contextual embeddings, but they hardly introduce improvements regarding the encoding of semantic information. They generate vector spaces with linguistic information but no semantic space.

Corpora

The production of word embeddings has become more popular in the last decade with the appearance of an increasing variety of methods and improvement techniques. Such methods are trained on large textual collections without the need for labelled data. Web scraper techniques together with huge online repositories have made possible the emergence of a large number of textual collections, such as the Google News corpus[2] or the Common Crawl collection[3] , where the predominant language is English.

Nevertheless, the extensive textual corpora on which these models have been trained are not as diverse as the methods themselves when dealing with non-English languages. Although various general purpose resources such as the Europarl[4] corpus (Koehn, 2005) have been proposed, they alone are not large enough. Undoubtedly, Wikipedia[5] has been the main resource adopted in the scientific community for the generation of Spanish and multilingual word embeddings as it stores large volumes of text for many languages. The 2019 Spanish version has a total of 24 million lines, 287 million tokens, and almost 2 billion characters according to the tokenisation described

[2] `news.google.com`

[3] Common Crawl is a non-profit organisation providing raw data from years of crawling websites in more than 40 languages: `https://commoncrawl.org/`

[4] Europarl comprises the proceedings of the European Parliament: `statmt.org/europarl`

[5] `https://es.wikipedia.org/`

in Section 3.3. Examples of vectors produced are the Polyglot word embeddings (Al-Rfou, Perozzi, and Skiena, 2013) or the Spanish vector representations provided by Flair (Akbik et al., 2019).

One of the most complete and diverse Spanish public corpora is the SBWC collected by Cardellino (2019) from the NLP group at the University of Chile. It comprises multiple collections such as Wikipedia, Wikisource[6], and Wikibooks[7] on date 2015-09-01, Sensem project (Alonso et al., 2007), Ancora Corpus (Taulé, Martí, and Recasens, 2008), Tibidabo Treebank and IULA Spanish LSP Treebank (Marimon et al., 2013), OPUS Project (Tiedemann, 2012), and Europarl corpus (Koehn, 2005). All the above sources involve 46 million lines, 1.4 billion tokens, and more than 7 billion characters. The author has also published vector representations generated with different algorithms such as word2vec and fastText. Those embeddings have been widely used by different scientific authors such as Doval et al. (2018), Santiso et al. (2019), and Soares et al. (2019).

A similar Spanish collection is the Spanish Unannotated Corpora (SUC). The author Cañete (2019) has included roughly the same sources but more up-to-date (dated 2019-04-20). Lines, tokens and characters are increased to 300 million, 3 billion, and more than 18 billion respectively.

Training on domain-specific texts often leads to better results despite the smaller size. However, restrictions on access to clinical texts have limited the generation of large collections on which to train in-domain word embeddings models. As far as we are aware, there is only a single relatively large public collection of clinical notes in English, called MIMIC-III (Johnson et al., 2016), whose data have been carefully deidentified, and there is no equivalent collection in Spanish. As illustrated in (Khattak et al., 2019), a popular strategy is the use of scientific literature within medicine and biology areas as the main sources for the production of the vectors. PubMed abstracts[8] is probably the most widespread biomedical textual resource for the generation of english medical word embeddings. For example, Zhang et al. (2019) combine PubMed abstracts and MIMIC-III for this task.

Given the limited amounts of Spanish content in PubMed, researchers typically focus on Scientific Electronic Library Online (SciELO) when dealing with Spanish biomedical collections. This is a virtual library with access to a collection of scientific health journals. Although it has just over 34,000 Spanish documents in 2020, it provides the full text and not just the abstract. Soares et al. (2019) have relied on SciELO and the biomedical portion of Wikipedia in order to produce the Spanish Health Embedding (SHE). These embeddings have been trained on a collection comprising 7.3 million lines, 182 million tokens, and 1 billion characters. To the best

[6] https://es.wikisource.org/

[7] https://es.wikibooks.org/

[8] PubMed is a repository of more than 30 million citations for biomedical literature: https://www.ncbi.nlm.nih.gov/pubmed/

of our knowledge, this is the unique work which releases Spanish medical vector representations. In fact, so far we are not aware of other similar publicly available representations. Most of the consulted papers use these vectors (Agirre et al., 2019; Akhtyamova et al., 2020), or general representations (Polignano et al., 2020; Rivera and Martínez, 2019).

6.2.4 Parameter-based transfer

The learned parameters constituting the prediction functions from NLP ML models store task-specific low-level language patterns. Two related tasks should share some of these low-level features, so theoretically parameters from source tasks could be exploited to transfer knowledge to target tasks. There are two main strategies in parameter-based transfer learning: multi-task and sequential transfer learning.

Multi-task learning

Multi-task learning relies on simultaneous trainings of related tasks. A joint learning is expected to generalize better as it involves an implicit data augmentation and attention mechanism by forcing to learn common representations which ignores the data-dependent noise. In addition, predicting multiple tasks at once introduces an inductive bias which acts as a regulariser preventing over-fitting and reduces the ability of the model to fit random noise, also called Rademacher complexity (Søgaard and Goldberg, 2016). Multi-task learning is particularly useful for target tasks depending on auxiliary ones (Søgaard and Goldberg, 2016) such as document classification, wich could leverage a NER task.

One of the first works in this direction was proposed by Collobert et al. (2011), who jointly trains tasks such as NER, POS Tagging, and chunking. Other authors have explored complementing the target task with MT tasks. Thereby, Luong et al. (2015) propose to learn together MT and parsing. A trend among authors has been to improve the language understanding capacity of the target task by including an additional language model target during training (Liu et al., 2018; Rei, 2017).

Multi-task learning does not necessarily improve the target task. Several studies have been conducted to identify the conditions surrounding negative transfer. For example, Bingel and Søgaard (2017) conclude that if the learning convergence is slow in the source task and fast for the target task, then the joint learning is likely to benefit the target task. In parallel, Changpinyo, Hu, and Sha (2018) explore the improvements of learning more than two tasks over only two. As counterpart, multitasking approaches imply inefficiency, as they generally require training from scratch each time and their task-specific target functions need to be custom-weighted (Chen et al., 2018).

Sequential transfer learning

Sequential transfer learning consists of transferring knowledge with a sequence of steps, tipically a first phase of pretraining and a second phase of adaptation. The most popular parameter-based transfer technique is fine-tuning: train a model on the source task, use the tuned parameters to initialise another model, and adapt or fine tune the parameters to the target task. The idea is that pre-training acts as a regularising element by improving generalisation (Erhan et al., 2010). In this line, Zoph et al. (2016) use a MT model trained on language pairs with considerable volume of resources to improve the learning of language pairs with less data. Another example are the application of emoji prediction to the Sentiment Analysis conducted by Felbo et al. (2017), or the use of a pre-trained POS Tagging model for the task of word segmentation explored by Yang, Zhang, and Dong (2017). Although the source tasks in such cases are specific, the latest trend is to choose source tasks that involve a greater understanding of the language and can therefore be used to improve more than one task.

The study conducted by Zhang and Bowman (2018) suggests that LM strongly capture syntax and improve target tasks more than other pre-training tasks such as translation or autoencoding. LMs require knowledge of syntax, semantics and actual facts, so these involve the acquisition of general knowledge, also tied to high complexity, even for humans. The parameters of LMs have been applied to many tasks, such as Sentiment Analysis (Severyn and Moschitti, 2015), Machine Translation (Ramachandran, Liu, and Le, 2016; Sennrich, Haddow, and Birch, 2015a), Question Answering (Min, Seo, and Hajishirzi, 2017), and NER (Baevski et al., 2019).

Many of the LMs used to generate contextualised embeddings have also been used as a general background for specific downstream tasks, such as ELMo (Peters et al., 2018), GPT-# (Brown et al., 2020; Radford et al., 2018, 2019), BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019b) models. Complementarily, Howard and Ruder (2018) propose the Universal Language Model Fine-Tuning (ULMFiT), which describes a series of techniques such as discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing to deal with the tendency of LMs to over-fit small data sets and degrade rapidly when adapting their parameters to classification. Numerous publications rely on the use of ULMFiT and some of previous models, especially ELMo and BERT, to address tasks in the clinical domain (Beltagy, Lo, and Cohan, 2019; Huang, Altosaar, and Ranganath, 2019; Lee et al., 2020; Peng, Yan, and Lu, 2019).

Furthermore, Phang, Févry, and Bowman (2018) explore to fine tune pre-trained models on intermediate tasks such as natural language inference before transferring to downstream tasks. In contrast, Wang et al. (2018) conclude that most intermediate tasks tend to worsen the information transferred. Furthermore, authors attempt unsuccessfully to improve the target tasks with multi-task pre-training. Other research

(Tenney et al., 2019; Wieting and Kiela, 2019; Zhang and Bowman, 2018) point out that randomly initialized, untrained neural networks achieve a high performance slightly less than many transfer learning proposals. In particular, Tenney et al. (2019) claim that such random methods exploit pre-trained word embeddings, so most of the contribution in these approaches comes from the representations.

Fine-tuning is an inefficient parameter-based method as pre-trained models tend to store a lot of parameters. Some authors have explored the combination of fine-tuning and multi-task learning. For example, Stickland and Murray (2019) propose to use the same BERT model for multiple tasks by sharing most of the parameters and only updating a few of them each time in order to reduce the task-specific parameters. In a similar way, Mulyar and McInnes (2020) explore a BERT architecture for multiple tasks in the clinical domain. Alternatively, Houlsby et al. (2019) propose to share parameter among tasks while using adapter modules which contain few trainable parameters per task. Other proposals have focused on reducing model size while preserving performance by applying knowledge distillation (Jiao et al., 2019; Sanh et al., 2019; Sun et al., 2020). Distillation is based on training a smaller model so that its output approximates the distribution of the original. But it is only useful in the inference time because additional training is required. For this reason, pruning methods which discard layers have also been explored (Gordon, Duh, and Andrews, 2020; Michel, Levy, and Neubig, 2019; Sajjad et al., 2020; Voita et al., 2019).

6.2.5 Relational-based transfer

Relational-based transfer is based on exploiting the relationship among data from a source task in order to improve a target task. One of the methods that is having the most impact recently is the zero-shot learning. It consists of transfer semantic knowledge extracted from the seen labels (source domain) via manual attributes or name representations to the classification of unseen labels (target domain) (Pourpanah et al., 2020). Hence, the inference is based on generalising relations between data point and label representations, with the semantic information bridging the gap between seen and unseen labels. For example, Pushp and Srivastava (2017) focus on learning relationship between text and weakly labels. Similarly, Zhang, Lertvittayakumjorn, and Guo (2019) explore the classification of documents with unseen labels.

6.2.6 Discussion

After outlining the SOTA in NLP on the different transfer learning methods, we consider appropriate to introduce the ICD coding proposals based on each of these methods and to contextualise our own proposals.

In the context of instance-based transfer, ICD proposals have focused on exploiting information from records in different languages. For example, Jeblee et al. (2018) and Ševa, Sänger, and Leser (2018) directly use source instances by applying multilingual word embeddings for ICD-10 code prediction. Otherwise, Ive et al. (2018) process information at character level in an attempt to leverage the similar linguistic morphology of the French and Italian languages. We have proposed the use of MT techniques to match features and weights to correct marginal probability distributions (Almagro et al., 2019).

Alternatively, most recent proposals are based on the use of word embeddings to exploit language modelling on large corpora via representation. Kalyan and Sangeetha (2020) present a table with a list of some of these proposals and the embeddings used for ICD-10 coding. Given that lexical variability is abundant in the clinical domain, sub-word information has been explored with FastText representations (Amin et al., 2019; Blanco et al., 2020; Ševa, Sänger, and Leser, 2018) for dealing with OOV words. Sub-word tokenisers have also been implemented in conjunction with BERT-style language models (Chalkidis et al., 2020; Zhang, Liu, and Razavian, 2020). In addition, some authors proposed enrich embeddings with semantic lexicons, such as Patel et al. (2017), who introduce information from ICD hierarchy, and Alawad et al. (2018), who use retrofitting techniques in conjunction with UMLS, SNOMED CT, and ICD-10 for identify ICD-O-3 codes. In view of these proposals, we have explored the generation and use of word embeddings based on character N-grams as well as the addition of structured information such as synonymy and antonymy relationships.

There is a trend towards the use of contextual word embeddings. For example, Blanco et al. (2020) compare traditional embeddings with contextual representations generated by ELMo. A comparison between XMTC models based on BoWs and networks fed with contextual embeddings generated from ELMo, BERT, and RoBERTa is presented by Chalkidis et al. (2020). The best result is obtained by the XMTC model AttentionXML, proposed by You et al. (2018). In turn, Zhang, Liu, and Razavian (2020) have applied the model AttentionXML fed by contextual embeddings for the prediction of about 2,000 frequent ICD-10 codes. Such representations have been generated from multiple BERT models pre-trained on collections of EHRs and health scientific publications. Since masked language models such as BERT require huge amounts of text, which is scarce in the Spanish clinical domain, we have trained a traditional language model based on the AWD-LSTM architecture. By using all words in the sentences, and not only the masked ones, as target words to be predicted, much less text is needed to achieve similar performances. In turn, the AWD-LSTM architecture, used in the ULMFiT paper, seems to be a good choice because of the performances achieved by other authors.

As for parameter-based transfer proposals, to the best of our knowledge, there are no multi-tasking approaches involving ICD coding. There are several proposals for fine-tuning BERT models. The non-English approaches (Amin et al., 2019; Sänger et al., 2019; Velichkov et al., 2020) have relied on general-purpose multi-lingual BERT models, while those works focused on English EHRs, such as BERT-XML proposed by Zhang, Liu, and Razavian (2020), have used BERT models pre-trained on clinical text collections. Alternatively, Silvestri et al. (2020) conducted a cross-lingual ICD-10 coding task with a BERT-style model, while Manginas, Chalkidis, and Malakasiotis (2020) explored a distinct fine-tuning process for each layer to adapt the ICD-9 hierarchical structure. All proposals have had to limit the size of the documents to the maximum size of the networks, generally to 512 and 1024. We have explored the use of general-purpose multi-lingual BERT models and the previously generated domain language model. Since EHRs tend to be long, methods to deal with length constraints have been explored.

Finally, some authors (Lu et al., 2020; Rios and Kavuluru, 2018; Song et al., 2020) propose to generalise the relationship between EHRs and ICD-9 code descriptons using relational-based transfer learning. We have not explored such techniques due to the low yields currently achieved.

6.3 Instance-based transfer learning proposals

Modern medicine tends to specialise in such a way that an increasing number of medical specialisations are offered in health systems. This implies the diversification of medical services in health centres, so that the data collected in a single hospital often only cover a certain number of topics. Additionally, diseases and pathologies vary according to environmental and socio-economic factors, with certain lifestyles contributing to the development of non-communicable diseases (Europe, 1999), e.g., more than 20 per cent of global deaths in 2016 are attributable to environmental conditions (Prüss-Üstün et al., 2016). As discussed in Section 1.2.2, all of these circumstances contribute to the collection of highly biased, strongly origin-dependent ICD data sets.

A data set with EHRs from a single health centre is unlikely to reflect information for an exhaustive number of codes. Instead, different origins often lead to variations in code distributions, keeping a degree of overlap. In theory, the greater the geographic distance and/or cultural diversity of the sources, the greater the variability of diagnoses. For this reason, it could be interesting to explore the exploitation of other data sets to fill the gaps in the main code distribution by enriching the information of under-represented codes.

We have implemented a traditional BoW approach in order to explore the feasibility and effectiveness of such an instance-based transfer proposal applied to data sets collected by health centres in different countries. The records from each source differ in language, so a cross-lingual approach is required. For this purpose, we have employed an auxiliary MT method based on DeepL[9] to transform the records from one language to another.

The proposed methods and experimentation are detailed in Sections 6.3.1 and 6.3.2 respectively.

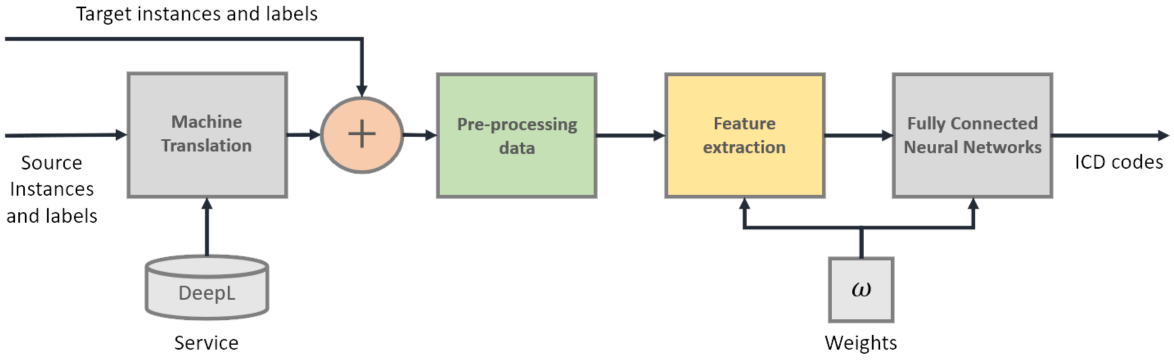
6.3.1 Method

The proposal consists of four sequential modules: Machine Translation, Pre-processing data, Feature extraction, and FCNNs. The main pipeline is shown in Figure 6.6: first, source instances are translated into target language; second, a lexical normalisation is applied to all instances; then, different features are captured for each code weighting the instances according to the origin; finally, binary neural networks are trained on these instances using the same previous weights in the loss function.

Machine Translation

Source instances have been translated into the target language by means of MT techniques based on the publicly available neural network DeepL. This MT model has been developed with convolutional networks enhanced with attention mechanisms and trained on the Linguee database (Coldewey and Lardinois, 2017). Linguee[10] is an online multilingual dictionary with access to more than 1 billion sentece pairs extracted from web indexers, with numerous websites of health care institutions included within the sources. Thus, such a MT model is expected to apply relatively high-quality automatic translations in much of the data due to the presence of medical vocabulary during training and the low verbosity of the sentences of the records.

Figure 6.6: *Pipeline of the cross-lingual instance-based method for ICD-10 coding.*



[9] https://www.deepl.com
[10] https://www.linguee.com

Despite this expected good quality, translations are subject to errors and a flattening of the lexical diversity inherent in the sentences, so that translated source instances are assumed to be of poorer quality than untranslated ones. We want to ensure that all implemented models are trained with at least one target instance in order to avoid a negative transfer by learning codes without reliable representation. For this reason, only the source instances associated with the codes found in the target subset have been used.

Pre-processing data

The tokenisation and lexical standardisation processes described in Section 3.3 have been applied. Due to linguistic variations, the lemmatiser described above has not been used in this case, but only a stemming process based on Porter's algorithm (Porter, 2001) and adapted to the target language. Furthermore, a conventional stop word list for the target language has been used and no synonyms have been introduced.

Given the volume of source instances and the application of MT techniques, an increase in the vocabulary with which to represent the target instances is expected despite some overlap. Such differences in vocabulary cause the difference in marginal probability distributions and the need to use weights as a corrective factor.

Feature extraction

The same label-specific features described in Section 5.3.2 have been used, which are based on BoW transformation, χ^2 feature selection, and TF-BNS value estimation. However, the following modifications have been made to adapt the vocabulary of the two sources:

- **BoW transformation**. Target and transformed source instances have first been transformed into BoW considering the new extended vocabulary.
- **Feature selection**. The higher the χ^2 value, the higher the probability that the word and code are not independent. Before applying Equations 5.3 and 5.4, the numbers of instances O, $O_{e_{W_i}}$, and $O_{e_{C_i}}$ are composed of the number of target instances (O_T, $O_{T_{e_{W_i}}}$, or $O_{T_{e_{C_i}}}$) and the weighted number of source instances (O_S, $O_{S_{e_{W_i}}}$, or $O_{S_{e_{C_i}}}$), as illustrated in Equations 6.1, 6.2 and 6.3. The parameters follow the same notation as in Equations 5.3 and 5.4, but with the S-index and T-index for source and target respectively. The numbers of source instances have been weighted to compensate for the higher volume than target instances, preventing the translated terms from being the only selected words. The parameter w will be adjusted to see how the weighting of some instances versus others influences the training.

$$O = w \cdot O_S + O_T \tag{6.1}$$

$$O_{e_{W_i}} = w \cdot O_{S_{e_{W_i}}} + O_{T_{e_{W_i}}} \tag{6.2}$$

$$O_{e_{C_i}} = w \cdot O_{S_{e_{C_i}}} + O_{T_{e_{C_i}}} \tag{6.3}$$

- **TF-BNS value estimation**. In order to compensate for inequalities between source and target vocabularies, we have modified Equations 5.6 and 5.7 by introducing weights depending on the origin of the instances (S-index for source and T-index for target). The parameters follow the same notation except for these indices. Thus, both rates are tuned in Equations 6.4 and 6.5, where the ratio of instances is divided into two terms, one for source instances and the other for target instances. As mentioned, the parameter w will be automatically set.

$$P_w(W_i|C_i) = w \cdot \frac{O_{S_{W_i,C_i}}}{OS_{C_i}} + \frac{O_{T_{W_i,C_i}}}{O_{T_{C_i}}} \tag{6.4}$$

$$P_w(W_i|\overline{C}_i) = w \cdot \frac{O_{S_{W_i,\overline{C}_i}}}{OS_{\overline{C}_i}} + \frac{O_{T_{W_i,\overline{C}_i}}}{O_{T_{\overline{C}_i}}} \tag{6.5}$$

Fully Connected Neural Networks

As code-specific features have been generated for each code, an OvR strategy has been implemented using a binary classifier per code. Specifically, the **B-MLP** model described in Section 5.4.1 has been used to predict the probability of each code given an instance.

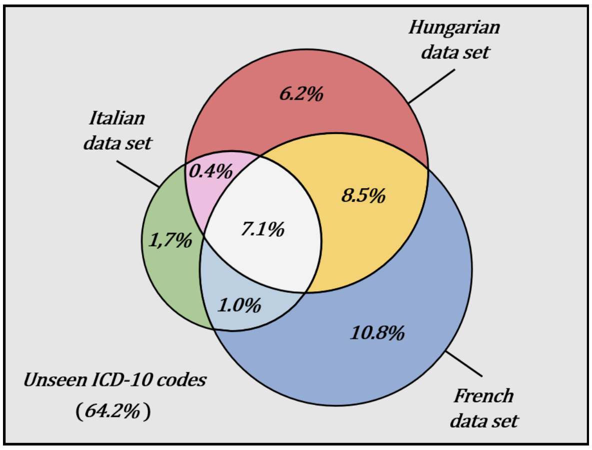
As previously discussed, source and target instances differ in the marginal probability distributions $\big(\mathcal{X}_1 \neq \mathcal{X}_2\big)$. For this reason, a binary cross-entropy loss function has been applied by assigning weights to each instance to correct for marginal probabilities. Such loss functions is shown in Equation 6.6, where w are the assigned weights, y are the binary true values of the code (1 if present and 0 if absent), and $\hat{y}$ are the predicted values in the range of 0 to 1 (the closer to 1 the higher the probability that the code is relevant to the instance).

$$CE(\hat{y}, y) = -w \cdot \big(y \cdot log(\hat{y}) + (1-y) \cdot log(1-\hat{y})\big) \tag{6.6}$$

6.3.2 Experimentation

We have evaluated the proposed method on the *Causes of Death* corpus as it is composed of three data sets in different languages and from three distant health centres: the INSERM in France, the KSH in Hungary, and the ISTAT in Italy. The different factors surrounding the health centres lead to differences in the distributions of codes as shown in Figure 6.7, which shows the code overlaps and intersections between these data sets.

Figure 6.7: *Percentage of codes included in each subset from the Causes of Death corpus, with intersections and overlaps.*



Given the limitations of the available MT techniques for clinical texts, we have focused on French and Italian, which have been empirically tested for better translation quality. In particular, the experiments have focused on the transfer of instances from the French set to the Italian one as the volume of French instances is much higher than the Italian one per code, on average 5 to 1. Hence, the lines of the French death certificates constitutes the set of source instances, while the Italian lines form the set of target instances.

The difference in the marginal probability distribution can be observed in the histogram of tokens within the Italian and French-origin instances associated with the same codes and resulting from the pre-processing methods (Figure 6.8). The MT techniques introduce a large percentage of new vocabulary –88% of the tokens from French instances are unseen– while not being able to retrieve 30% terms as ”*cerebrovascolopatia*“, ”*vascolopatia*“ and ”*iponatriemia*“; instead, the MT model suggests ”*cerebrovascolari*“, ”*vascolari*“ and ”*iponatremia*“. In total, there were 4,372 different words in the Italian vocabulary (Section 3.2), but this number is increased to 29,082 unique words by introducing the Italian-translated French certificates.

The following are the settings explored and the results of each one.

Experimental settings

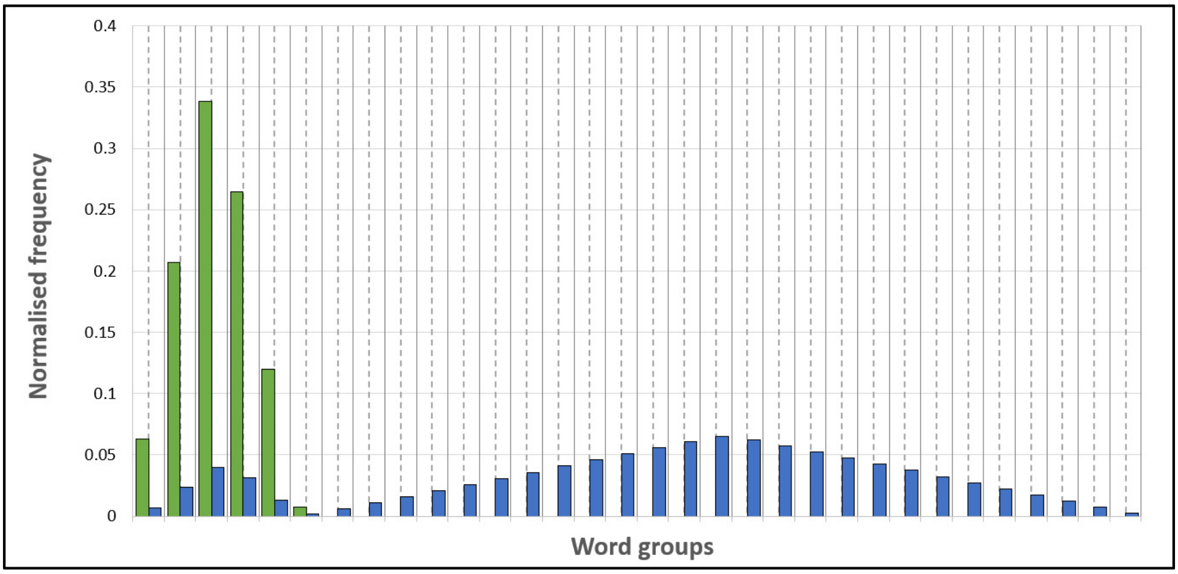
All instances of Italian origin have been unweighted, while different settings have been tested involving weights of 1.0 and 0.2 for the French instances. The idea of exploring multiple scenarios is to analyse the performance when the number of source instances exceeds the number of target instances or the impact of the source instances is reduced in proportion to the 5 times higher volume.

Although the representation of the documents varies according to the weights of each configuration, all settings reduce the feature dimensionality to the 1,000 words with the highest χ^2 values for each code and compute BNS values on such set. As far as the network is concerned, the same settings described in Section 5.4.2 for **B-MPL** have been applied: $L = 4$, $h = 80$, adaptive learning rate, Adam optimizer, and BCE loss function.

The settings are as follows:

- **No Addition Setup (NAS)**. Only target instances are used for training the models.
- **Raw Addition Setup (RAS)**. Source instances have been incorporated into the training of all models with the weights 1 and 0.2, increasing the number of negative cases for each code.

Figure 6.8: *Histogram of the tokens present in the instances with the same codes. The green values correspond to the frequencies of the Italian token groups, and the blue values are the frequencies of the French token groups, translated into Italian.*



- **High-Low Frequency Addition Setup (HLFAS)**. The corresponding source instances have been included only in the training of rare codes with the weights 1, and only in the training of frequent codes with the weights 0.2. All codes with more than or equal to 40 training examples have been considered frequent.
- **Separate Addition Setup (SAS)**. The corresponding source instances have been included in the training of the frequent and rare codes as two independent processes, without introducing negative examples in either of the two blocks.
- **Individual Addition Setup (IAS)**. Positive instances have been included independently in each model, contributing only positive examples for each code. The weights 1 and 0.2 have been reused.
- **Individual Scored Addition Setup (ISAS)**. Similar to the **IAS** setup, but using weights as a function of the code and frequency, as described in Equation 6.7, where c is the code, $f(c)$ is the training frequency, and f_{MAX} is the frequency value of the most common code.

$$w(c) = 1 - \frac{f(c)}{f_{MAX}} \tag{6.7}$$

Results from all scenarios have been evaluated with all the metrics described in Section 3.4. In addition, F-Score values have been disaggregated into groups of codes with similar frequencies to compare the behaviour of configurations in different ranges.

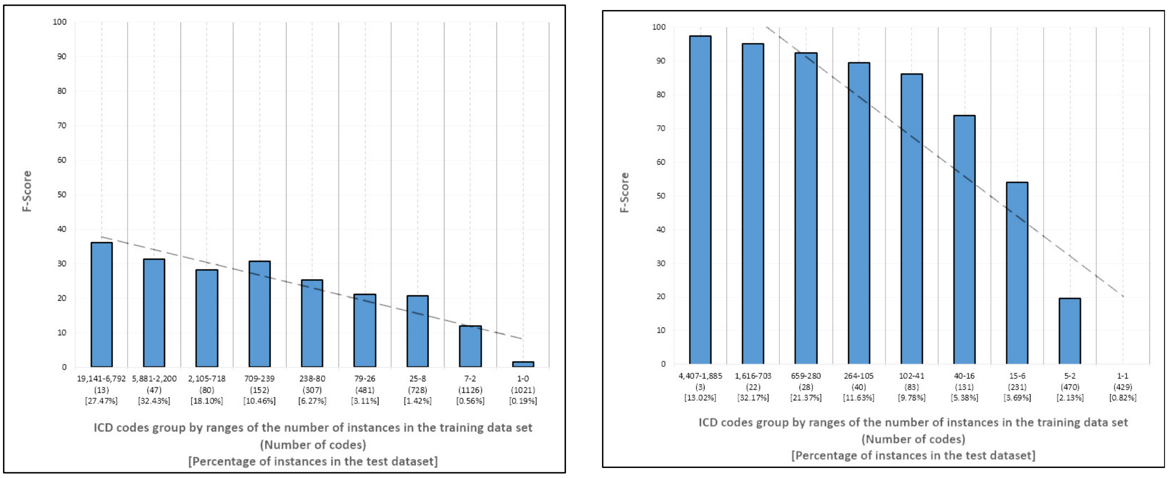
Results

The individual performance of the French and Italian subsets has been examined in more detail by applying the same pre-processing, feature extraction and training methods, excluding the translation techniques. Figure 6.9 shows the performance of both systems, French in Figure 6.9a and Italian in Figure 6.9b (both **NAS** setting), breaking down the Micro-averaged F-Score values into groups of codes with similar frequencies. As can be seen and would be expected, performance is proportional to frequency: the more instances, the better the learning converges and the higher the inference.

A larger number of examples usually results in a better characterisation of codes by the classification models. This is also evidenced by the difference between the macro and micro results for all settings presented in Table 6.1. Overall, the addition of translated labelled instances seems to influence this availability of examples by contributing to the overall improvement of the scores.

In particular, introducing the instances directly (**RAS** setting) yields a 1.4% increase in Precision, slightly lower for the hierarchical structure (P_S) and higher for

Figure 6.9: *Individual performance of the system trained on the Italian or French records in the form of Micro-average F-Score, broken down into groups of codes with similar frequencies.*



(a) *Micro-average F-Score values of the coding of French records disaggregated into groups of codes with similar frequencies.*

(b) *Micro-average F-Score values of the coding of Italian records disaggregated into groups of codes with similar frequencies.*

the propensity scored value (PSP), at the expense of Recall. The higher number of instances results in more accurate but less exhaustive models. The difference between macro-average scores —e.g. 45.19 in contrast to 31.24 F-Score— reveals that not weighting the translations (**RAS-1**) benefits a larger number of codes. As can be noticed with the F-Score values broken down by groups of codes with similar frequencies in Figure 6.10, the less frequent codes (the last four groups, on the right) improve performance by about 6 points on average —reaching a value of 63.92 out of the initial 58.31 F-Score— at the expense of frequent code performance, which is reduced from 93.49 to 93.12 F-Score. This is also reflected in the 2% increase in the PSP value. In contrast, weighting source instances to balance volume (**RAS-0.2**) leads to minor improvements for common codes —a 93.98 F-Score— but negatively affects rare codes, which fail to sufficiently strengthen their representation with a decrease of 12.92% in F-Score.

The results for **HLFAS** setting confirm these assumptions by separately enhancing common and rare codes. On the one hand, similar results to **RAS** scenario are achieved by enriching only the less frequent codes with unweighted translations. The impact of source instances on learning is intended to be increased with larger weights as less frequent target codes tend to perform worse. Similarly, improvement is also achieved by enriching only frequent codes (more than 40). For these, less transfer leading to less impact of source instances during training is desirable as frequent target

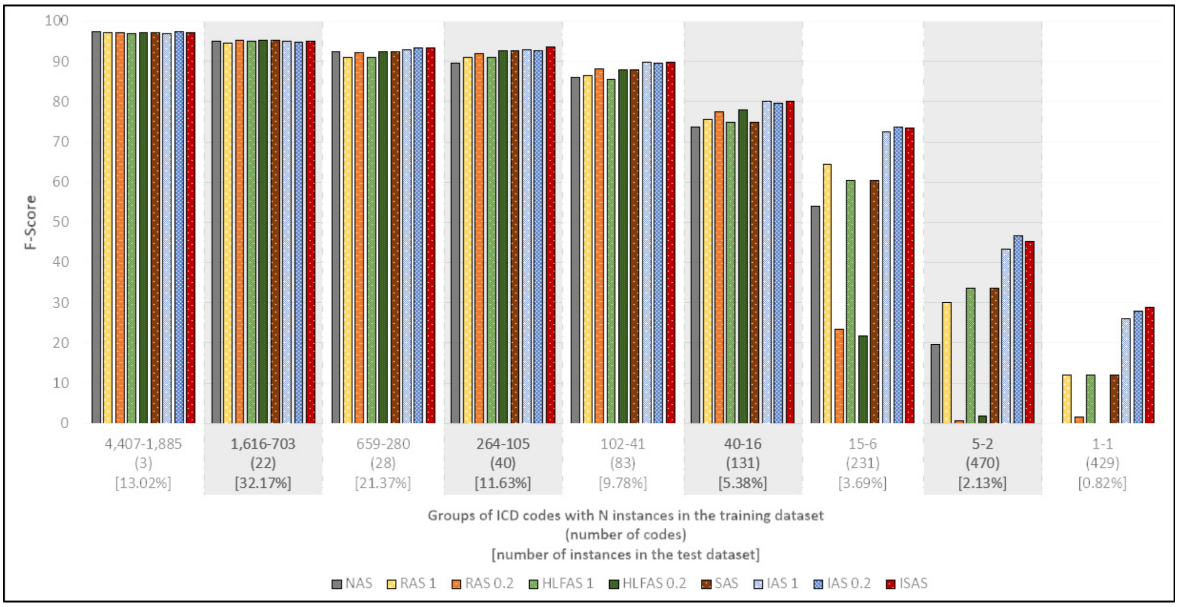
Score		Baseline	NAS	RAS		HLFAS		SAS	IAS		ISAS
				1	0.2	1	0.2		1	0.2	$w(c)$
Micro	P	16.37	92.96	94.27	**94.77**	93.90	94.22	94.03	90.88	89.33	89.76
	R	15.24	85.98	85.17	84.71	85.33	85.29	86.80	91.84	**93.59**	93.23
	F	15.78	89.33	89.49	89.46	89.40	89.54	90.27	91.36	91.41	**91.46**
	P_S	25.73	94.40	95.52	**95.91**	95.20	95.47	95.27	92.13	90.45	90.92
	R_S	23.96	87.31	86.32	85.74	86.57	86.44	87.97	93.23	**94.89**	94.57
	F_S	24.82	90.72	90.69	90.54	90.68	90.73	91.47	92.67	92.62	**92.71**
	PSP	15.14	90.84	92.73	**93.46**	92.08	92.75	92.29	87.79	86.08	86.61
	PSR	6.47	77.94	78.00	75.68	78.02	76.41	79.67	88.04	**90.21**	89.87
	PSF	9.07	83.90	84.73	83.63	84.47	83.79	85.51	87.92	88.09	**88.21**
	PSP_S	24.72	92.80	94.39	**94.88**	93.84	94.32	93.93	89.46	87.56	88.16
	PSR_S	11.08	80.22	79.82	77.28	80.00	78.20	81.56	90.35	**92.38**	92.12
	PSF_S	15.30	86.05	86.50	85.18	86.37	85.51	87.31	89.90	89.91	**90.09**
Macro	P	0.07	39.24	47.67	32.62	47.04	32.16	47.13	50.75	**51.40**	50.92
	R	0.44	37.11	42.95	29.97	42.70	30.30	43.29	52.60	**56.65**	55.82
	F	0.12	38.14	45.19	31.24	44.76	31.20	45.13	51.66	**53.93**	53.28
	P_S	0.11	40.84	49.48	33.52	48.91	33.13	49.01	52.99	**53.57**	52.97
	R_S	0.44	37.85	44.07	30.52	43.78	30.86	44.37	54.49	**58.60**	57.61
	F_S	0.18	39.29	46.62	31.95	46.21	31.95	46.58	53.74	**56.00**	55.21
	PSP	0.02	32.56	41.57	25.29	40.92	24.86	40.98	46.61	**47.55**	46.93
	PSR	0.10	30.61	37.39	23.02	37.17	23.24	37.57	48.49	**52.66**	51.79
	PSF	0.03	31.56	39.37	24.10	38.95	24.02	39.20	47.53	**50.00**	49.27
	PSP_S	0.02	34.04	43.35	26.07	42.74	25.69	42.82	48.87	**49.76**	49.00
	PSR_S	0.10	31.31	38.50	23.51	38.21	23.72	38.62	50.41	**54.67**	53.59
	PSF_S	0.04	32.62	40.78	24.72	40.35	24.67	40.61	49.63	**52.12**	51.21
Ordering	$nDCG$	44.08	91.07	90.71	90.42	90.47	91.08	92.16	95.52	**96.76**	96.41
	$nDCG_S$	64.60	92.34	91.60	91.46	91.48	91.98	93.02	96.57	**97.74**	97.41
	$PSnDCG$	43.76	90.91	90.56	90.27	90.33	90.93	91.93	95.33	**96.55**	96.20
	$PSnDCG_S$	63.63	92.18	91.43	91.30	91.33	91.82	92.78	96.37	**97.53**	97.19

Table 6.1: *Evaluation of the instance-based transfer learning using different weights.*

codes are better characterised and practically do not need additional information. Both contributions are combined in the **SAS** setting, which enhances the codes of each group without penalising the others. Hence, the last four groups (on the right) improve F-Score by 7.7%, and the rest by 0.6%. This escenario improves both the macro-average Precision and Recall on the initial setting (**NAS**).

Binary training has been used to feed all non-translated instances and only translated code-related instances to each model (**IAS** setting), thus avoiding unbalancing the codes by increasing the volume of negative instances. Weighting with 0.2 achieves the highest Recall and macro-averaged scores together with rank order metrics such as nDCG. The improvement in Recall implies an increase of 8.85% over **NAS**, 15.74% if we compare propensity scored values, which suggests that this approach is more prone

Figure 6.10: *F-Score values disaggregated into groups of codes with similar numbers of training instances, ordered from lowest to highest frequency.*



to risk-taking. In turn, macro-averaged values increase by 50% on average, which means a significant improvement in generalisation during the training of models associated with less frequent codes. In terms of order metrics, values are close to 98 out of 100, which means that nearly all wrong predictions are produced with a smaller confidence interval than the correct ones.

Although the translated instances of other codes do not affect the training of the models, fixed weights are still used, so the difference in volumes between the source and target examples of each code is not considered. **ISAS** setting is designed to balance the source and target examples of each code independently. Although the Recall decreases slightly compared to the **IAS** setting weighted at 0.2, this setting achieves a lower Precision penalisation reaching the highest F-Score: 91.46. Overviewing all the metrics, one notes that **IAS** is above in most of them. However, **ISAS** obtains slightly lower values and one might appreciate the fact that it uses a weight function instead of a constant which may be adjusted to this particular collection.

6.4 Feature-based transfer

As discussed in Section 6.2, the semantics of words can be estimated in relation to the contexts described in large textual resources. With this idea in mind, different methods have been proposed to constitute reduced dimensional vector spaces in which represent some pieces of text trying to preserve some spatial coherence between meanings.

It is common to find compound words and greco-latin prefixes and suffixes in clinical texts, in addition to typos slightly modifying the original words. Given the considerable number of lexical variations, we have decided to focus on the method FastText (Bojanowski et al., 2016) and BPE (Sennrich, Haddow, and Birch, 2015b) for transferring representations. FastText deals directly with sub-word information leading to a more flexible approach capable of representing derived forms of words learned during training. In addition to a greater coverage when dealing with OOV words, relying on character n-grams improves the characterization of rare words. Instead, we have used BPE for the contextualised vectors derived from AWD-LSTM (Merity, Keskar, and Socher, 2017), which achieves similar purposes.

Although three pre-trained Spanish word embeddings models have been published, there are no Spanish embeddings specifically generated from medical records. SBWC and SUC are trained on general purpose sources, with scarce medical vocabulary. Otherwise, SHE is trained with the medical and biological portion of Wikipedia, so it is likely that many of the meanings specific to the clinical sector are not captured. We therefore propose to exploit the large volume of records to generate clinical word vectors with which to adequately represent subsequent records.

Experimentation has been conducted on a subset of codes to avoid strongly negative effects of the imbalance while dealing with the high computational cost in deep learning. We have selected two extended document classification architectures for experimentation: Recurrent Convolutional Neural Network (RCNN) and HAN. Both models have been combined with techniques for dealing with imbalance data sets.

The generation of Spanish Clinical Embedding (SCE) based on FastText and Contextualised Spanish Clinical Embedding (CSCE) based on AWD-LSTM, the intrinsic evaluation of these vectors, the classification models, and the corresponding experimentation on a sample of ICD codes are presented below.

6.4.1 Generation

The generation of word vectors can be broken down into four sections: sources, text pre-processing, models, and outputs.

Sources

The core of the vector generation has been the set of records comprising the collections described in Section 3.2. A collection of PhD theses in medicine and related domains publicly accessible until 2019[11] has also been produced to introduce information from

[11]

the medical literature (the collection can be downloaded at the link in the footer[12]). In this line, the sources explored by Soares et al. (2019), SciELO and Wikipedia, have also been used. Besides, we have incorporated the descriptions provided by SNOMED CT, as well as the set of concepts linked by their relationships in the ontology. Finally, coverage has been intended to be increased by introducing general-purpose sources from the SBWC and SUC in extended models. Given the volume of such sources (ten times greater), these texts have been weighted by 0.1 (the inverse of the volume) during learning to reduce its impact and promote clinical data.

Text pre-processing

A customised algorithm[13] has been developed to convert pdf-format doctoral theses into free text while maximising formatting consistency. Once all the texts have been collected, all characters have been converted to lower case and the tokeniser described in Section 3.3 have been applied to identify the words. As for traditional embeddings, punctuation marks and stop words have been removed with the aim of facilitating the approach of the most meaningful words as its generation is based on unordered context windows. Instead, Language Models preserve order and assume at least whole sentences as contexts, so it is desirable to preserve all information affecting syntax. In both cases, decimal numbers or integers greater than ten have been masked.

Figure 6.11 shows the cumulative distribution of the tokens in the SCE corpora in black, the frequency at which tokens appear less than 5 times (typical cut-off frequency in embeddings generation) in blue and the intersection in red. Although more than 1.7 and 6.8 million unique tokens would constitute the vocabulary of both corpora, only 421 thousand and just over 1.2 million unique tokens exceed the 5 occurrence threshold.

Models

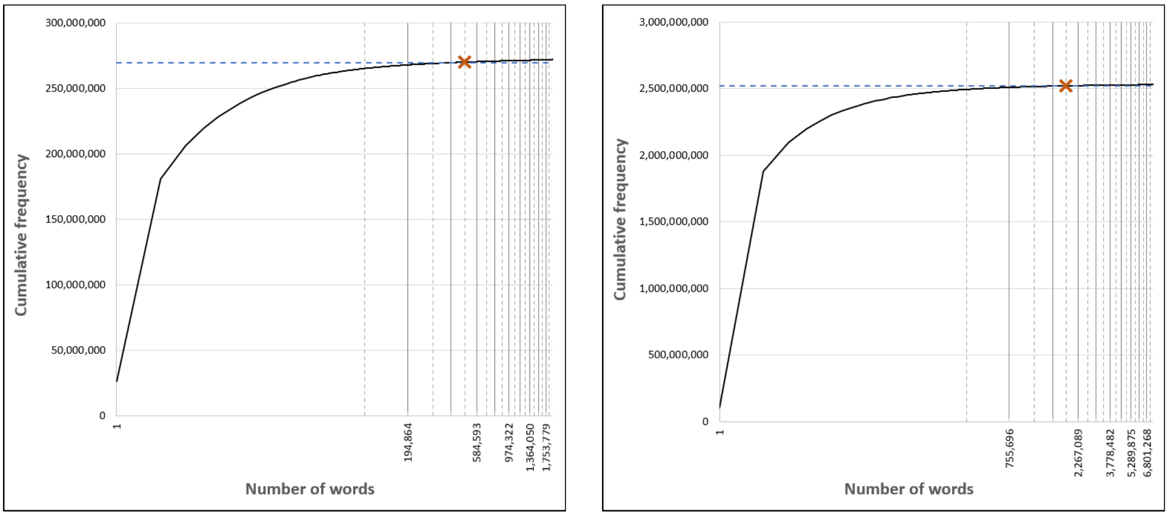
We have used two different models for training the word vectors: FastText and AWD-LSTM.

FastText This method is an extension of a skipgram model based on a shallow, fully connected neural network. The objective function consists of predicting context words from each of the words in the sentence, where the words are defined by the set of character n-grams. For example, the word ”*heart*“ would be given by the set {”*hea*“, ”*ear*“, ”*art*“, ”*hear*“, ”*eart*“, and ”*heart*“}, considering n-gram lengths from 3 to 6. The objective function is described in Equation 6.8, where T is the set of indices of words in the sequence, W_t is the target word, C_t is the set of indices of words surrounding

[12] https://zenodo.org/record/5148872#.YQQ7FI77SUk

[13]

Figure 6.11: *Cumulative distribution of tokens in corpus SCE.*



(a) *Not including tokens from other general-purpose corpora.*

(b) *Including tokens from corpora SBWC and SUC.*

the word W_t, W_c is the corresponding context word, $N_{t,c}$ is a set of negative examples, and W_n is the negative context word. In turn, $\mathcal{G}_{W_t}$ is the set of n-grams composing the word W_t, z_g is the vector for each n-gram, and v_c is the context word vector.

$$\mu = \sum_{t=1}^{T} \left[\sum_{c \in C_t} \log\left(1 + \mathrm{e}^{s(W_t, W_c)}\right) + \sum_{n \in N_{t,c}} \log\left(1 + \mathrm{e}^{-s(W_t, W_n)}\right) \right] \tag{6.8}$$

$$s(W_t, c) = \sum_{g \in \mathcal{G}_{W_t}} z_g^{\top} v_c \tag{6.9}$$

We have trained the model with all words appearing at least 5 times, the skip-gram architecture, a max length of 1 word n-gram (no multi-token composition), a range of 3 to 6 character n-grams, a vector size of 300, and 20 epochs. This setting has been applied for the in-domain texts (SCE), and for the corpus expanded with general-purpose sources (SCE-L). In addition, we have explored the pre-initialization with general embeddings as an alternative to the joint training with general-purpose sources. For this purpose, we have initialised the model with the word vectors SBWC (SCE-SBWC), SUC (SCE-SUC), and ConceptNet (SCE-CN).

AWD-LSTM It is a Language Model model based on a weight-dropped LSTM (see Equations 5.14, 5.15, and 5.16), which is improved by regularization strategies such as DropConnect[14] and the averaged SGD. Instead of words, we have used BPE to split them into subwords using a fixed vocabulary of 32,000. We have selected a 3-layer LSTM architecture with 1,152 nodes each and a 400-dimension output. Two models has been trained: the model CSCE, which is trained only on clinical or medical texts, and the model CSCE-L, which is also trained on general-purpose texts.

Training phase

SCE-L has been trained on over 2.5 billion general and in-domain tokens, 0.9% of which have been ignored. The loss function yields a value of 4.15 after the first epoch, reducing to 0.09 after 20 epochs. In total, more than 1 million different representations have been generated. In turn, SCE produces around 420 thousand vectors[15] trained on 315 million in-domain tokens, with a training time 18 times shorter. The percentage of tokens not used in training is slightly higher, around 1.1%, while the initial and final loss function values are similar.

Furthermore, three data sets have been used to visualise the structure of the generated vectors: UMNSRS-sim, UMNSRS-rel, and MayoSRS. UMNSRS-sim and UMNSRS-rel (Pakhomov et al., 2010) consist of 566 and 587 pairs of medical terms from disorder, symptom, and drug categories, which have been scored in terms of similarity and relatedness by medical residents; medical coding experts have generated 101 scores measuring the relatedness of pairs of medical terms to produce MayoSRS.

Figure 6.12 shows a two-dimensional projection of the output vectors using PCA for a set of clinical words randomly selected from the above-mentioned data set. Three distinct regions can be distinguished in the resulting space: diseases on the left, symptoms at the bottom, and drugs on the right. An English translated version is attached in the Appendix H.

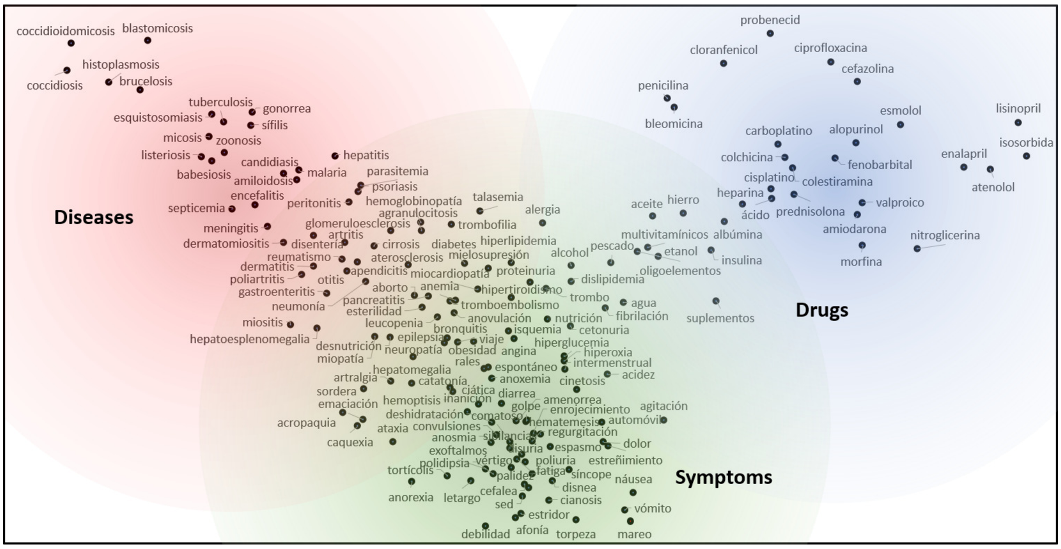
For contextualised embeddings, CSCE-L uses the large set of tokens, with a loss function ranging from 2.87 in the first epoch to 2.04 in the tenth epoch. This model achieves a perplexity of 7.7 and 99.9% word coverage with 32 thousand vocabulary sub-tokens. The clinical-only version (CSCE) has been trained only on the in-domain tokens, with a training time 3 times shorter. Figure 6.13 shows bold text paragraphs generated by the resulting Language Model (CSCE) given the previous context in light grey, which are parts of the example 3.12 and 3.13 used in Section 3.2 for the HUFA corpus. The generated text reflects a certain semantic and syntactic coherence, but is less robust in long dependencies: as new words are predicted, the initial context has less impact.

Table 6.2 shows an overview of the dimensions of the word embeddings generated and used in the experimentation, where the corpus size refers to training data and the

[14] DropConnect is a regularization method consisting of dropping random weights instead of activation functions in nodes.

[15] SCE vectors can be downloaded from `https://zenodo.org/record/5149010#.YQRa3HX7RH4`

Figure 6.12: *Two-dimensional representation of clinical words in SCE semantic space. An English translated version is attached in the Appendix H.*



word loss rate is the percentage of words ignored during training either because they were not sufficiently represented or did not pass the pre-processing filter.

	Corpus size (million tokens)	Vector number	Vector dimension	Word loss rate
SBWC	1,400	855,380	300	0.3
SUC	2,600	1,313,423	300	0.8
SHE	182	690,098	300	4.0
SCE	315	421,380	300	1.1
CSCE	315	32,000	400	0.0
SCE-L	2,581	1,182,898	300	0.9
CSCE-L	2,581	32,000	400	0.0

Table 6.2: *Estimated summary of the dimensions of the pre-trained word embeddings used in the experimentation.*

Additionally, vector re-orientation techniques have been employed on the non-contextualised embeddings SCE and SCE-L according to expert knowledge stored in semantic lexicons. In particular, we have used the method ATTRACT-REPEL with the lists of synonyms described in Section 3.3.3, using pairs of synonyms instead of groups. Equations 6.10 and 6.11 describe the two components of the optimisation function provided by the method, where B_A and B_R are mini-batches of k_1 synonym pairs and k_2 antonym pairs respectively, T_A and T_R are mini-batches of randomly selected negative examples, x_l^i and x_r^i is the word pair, t_l^i and t_r^i is the negative example pair, cos is the Cosine Similarity, τ is the hinge loss function, and δ_{att} and δ_{rep} are the attract and repel margins determining how much close or far away the vectors should be.

Figure 6.13: *English-translated example of automatic generation of clinical text based on the model CSCE. Parts of Example 3.12 and 3.13 in Section 3.2 (in grey) have been used to produce new parts (in bold). The untranslated example can be found in Appendix I.*

Anamnesis

PERSONAL HISTORY:
-PAH.
-Hansen's disease**. DM type 2. No known drug allergies.**
-Habitual drinker.
-Smoker of # cig/day for # years.

CURRENT CONDITION: patient discharged on 27/11 after admission for sigma obstructive neoplasia, who came for **an episode of metabolic acidosis due to bradycardia to continue treatment with Sintrom. Male, aged # years, attended the emergency department for fever and loss of consciousness. No chest pain or dysthermia, no chills or fever, no other accompanying symptoms.**

Physical Exploration
Afebrile (Tª 36.3ºC).Eupneic. Good general condition. Conscious and oriented in person, time and space. **Normal colour, well hydrated, nourished and perfused. Cyc: Carotid rhythmic and symmetrical. AP: mvc. No oedema or signs of DVT. Distal pulses present and symmetrical. Haemogram: Leukocytes #^{3}/l (#-#), Neutrophils %#% (#-#), Lymphocytes %#% (#-#), Monocytes %#% (#-#), Eosinophils %#% (#-#), Basophils %#% (#-#).**

Post-surgical changes consisting of a discharge colostomy and midline sutures of the abdominal wall.
In the left flank, ... **signs of metastasis with left orbit are observed. Bilateral mediastinal and axillary adenopathies compatible with inflammatory infiltrate in the pulmonary nodules (GCT). In the EEG territory, no data of signs of acute ischaemia or pathological findings are observed.**

CLINICAL JUDGMENT:
- Intestinal ileus. Intra-abdominal collection after Hartmann 17/11/2017 by neoplasia in the sigma.
- Distended gallbladder, with intrahepatic bile duct dilatation, Cholecystectomy.
- HBP with DM.
- Acute ischaemia in isotopic ergometry.

Evolution
During admission, digestive intolerance with food vomiting and **normal stools. Signs of CHF. High blood pressure on treatment with OADs. Episode of haemodynamic instability with rapid paroxysmal atrial fibrillation.**

Besides these relations, we have included the general direct synonyms and antonyms stored in ConceptNet. For further enrichment, in addition to the direct synonyms contained in SNOMED CT, we have extracted 30,238 synonym pairs involving 12,059 words by searching for lexically similar expressions but differing in a single word. In total, 1,005 antonym pairs and 645,125 synonym pairs have been used, involving 1,321 words and 597,392 words respectively.

$$Att(B_A, T_A) = \sum_{i=1}^{k_1} [\tau(\delta_{att} + cos(x_l^i, t_l^i) - cos(x_l^i, x_r^i)) + \tau(\delta_{att} + cos(x_r^i, t_r^i) - cos(x_l^i, x_r^i))] \tag{6.10}$$

$$Rep(B_R, T_R) = \sum_{i=1}^{k_2} [\tau(\delta_{rep} + cos(x_l^i, x_r^i) - cos(x_l^i, t_l^i)) + \tau(\delta_{rep} + cos(x_l^i, x_r^i) - cos(x_r^i, t_r^i))] \tag{6.11}$$

6.4.2 Intrinsic evaluation

An instrinsic evaluation of non-contextual embeddings on STS tasks has been performed using unsupervised methods. Contextual embeddings have not been used as these representations are not linearly independent and therefore each component has different relevance, i.e., they require training data to find non-linear functions. The data sets, methods, evaluation and results are described below.

Clinical STS data sets

We have used the UMNSRS-sim, UMNSRS-rel, and MayoSRS data sets proposed by Pakhomov et al. (2010) and Pakhomov et al. (2011). Those provide 566, 587, and 101 pairs of multi-term clinical concepts respectively with manually annotated semantic similarity measures. We have used a MT method based on DeepL followed by a manual review to translate these pairs into Spanish. In addition, we have included an evaluation of the reduced version of these data sets proposed by Soares et al. (2019) in the Appendix J, which include 380, 384, and 101 pairs.

STS methods

The semantic similarity between two pairs of multi-term expressions has been calculated using the Average Cosine Similarity (ACS) and Word Mover's Distance (WMD).

The ACS is described in Equation 6.12, where U and V are the weighted average of sets of word vectors of length L_1 and L_2 respectively, $U \cdot V$ is the dot product (or inner product) of the average vectors, and $\|U\|_2$ and $\|V\|_2$ are the norm (or length) of the average vectors.

$$ACS(U,V) = \frac{U \cdot V}{\|U\|_2 \|V\|_2} \tag{6.12}$$

$$U = \frac{\sum_{i=1}^{L_1} U_i}{L_1} \tag{6.13}$$

$$V = \frac{\sum_{j=1}^{L_2} V_j}{L_2} \tag{6.14}$$

WMD is a method proposed by Kusner et al. (2015) and based on the Earth Mover's Distance which aims to measure the distance between two probability distributions over the same region. This is an optimization problem described in Equation 6.15 with the constraints being detailed in the Equations 6.16, 6.17, and 6.18. L_1 and L_2 are the number of unique words in the two sequences, $W_{1,i}$ and $W_{2,i}$ are the frequency values of the ith and jth words, $d_{j,i}$ is the Euclidean distance between the ith and jth words, and T_{ij} is the amount of the ith word that travels to the jth word.

$$min \sum_{j=1}^{L_2} \sum_{i=1}^{L_1} T_{ij} d_{j,i} \tag{6.15}$$

$$\sum_{j=1}^{L_2} \sum_{i=1}^{L_1} T_{ij} = \sum_{j=0}^{L_2} W_{2,j} \tag{6.16}$$

$$\sum_{i=1}^{L_1} \sum_{j=1}^{L_2} T_{ji} = \sum_{i=0}^{L_1} W_{1,i} \tag{6.17}$$

$$T_{ij} \geq 0 \tag{6.18}$$

STS evaluation

The most widespread metric in STS is the Pearson correlation coefficient (ρ). We have used ρ to measure the correlation between the similarity measures yielded by the methods described above and the values annotated by experts. Pearson values range from 1 to -1, with the extremes being a total correlation and 0 a null correlation. The function is described in Equation 6.19, where n is the sample size and (x_i,y_i) are the pairs of similarity values.

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \tag{6.19}$$

Results

In addition to the word embedding models presented in Section 6.4.1, we have evaluated the representations proposed by other authors: ConceptNet (CN), SBWC, SUC, and SHE. Table 6.3 shows the Pearson correlation coefficients for all models.

SCE models reach the highest coefficients for both similarity methods. In particular, SCE representations trained on SUC or SBWC show the strongest correlations, with a 20% increase over SHE and 50% over general purpose embeddings using ACS, and increases of 7% and 15% using WMD. Clinical and general joint training seems to work worse than general pre-training.

It should be pointed out that retrofitting techniques do not show improvements using WMD because of the OOV words. It is further interesting to notice that the WMD method yields a zero similarity correlation for ConceptNet (NB) and retrofitted embeddings. The reason is that the Retrofitting methods normalise the vectors, just like those from ConceptNet. WMD relies on the Euclidean distance, but it disappears with all modules being of size 1.

	r based on Average Cosine Similarity				**r based on Word Mover's Distance**			
	UMNSRS-sim	UMNSRS-rel	MayoSRS	Avg.	UMNSRS-sim	UMNSRS-rel	MayoSRS	Avg.
CN	0.34	0.35	0.26	0.34	0.03	0.06	0.01	0.04
SBWC	0.36	0.36	0.14	0.34	0.36	0.37	0.34	0.36
SUC	0.44	0.42	0.12	0.40	0.42	0.42	0.31	0.41
SHE	0.47	0.47	0.33	0.46	0.44	0.44	**0.40**	0.44
SCE	**0.59**	0.51	**0.54**	**0.55**	0.46	0.46	0.38	0.45
SCE-L	0.58	**0.53**	0.43	0.54	0.40	0.40	0.37	0.40
SCE-SBWC	**0.59**	0.52	0.53	**0.55**	**0.48**	**0.48**	0.38	**0.47**
SCE-SUC	0.58	0.52	0.52	**0.55**	**0.48**	**0.48**	0.38	**0.47**
SCE-CN	0.58	0.52	0.53	**0.55**	0.46	0.47	0.39	0.46
Retrofitted SCE	0.47	0.43	0.51	0.45	0.05	0.09	0.08	0.07
Retrofitted SCE-L	0.50	0.42	0.43	0.46	0.10	0.05	0.13	0.08

Table 6.3: *Intrinsic evaluation of word embeddings models through Pearson correlation coefficients in an STS task.*

6.4.3 Classification models

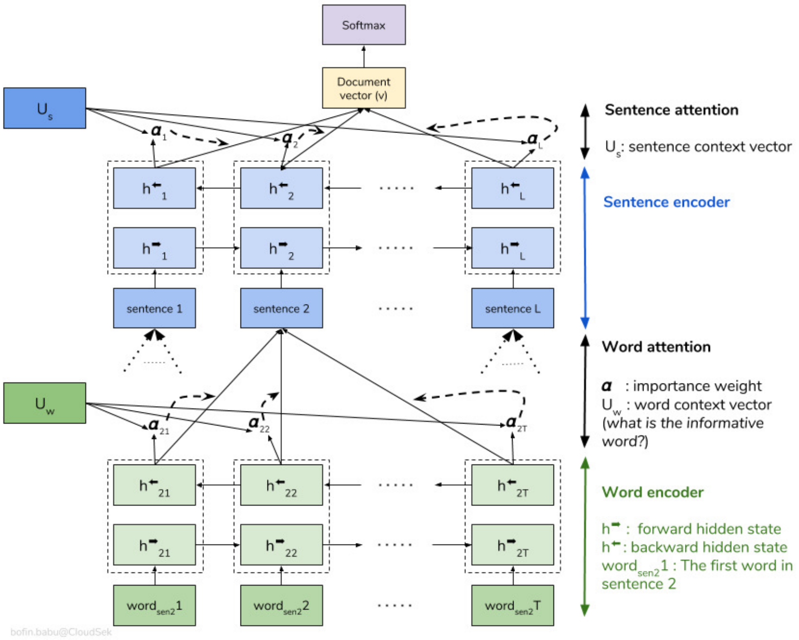
We have focused on two architectures that perform robustly for classification in long text as documents for ICD experimentation: RCNN and HAN.

RCNN combines recurrent with convolutional neural networks to contextually enrich the information of each word, thereby providing a robust architecture for long texts. It has been described in Section 5.4.

Regarding the HAN model, it combines sequential architectures at word and sentence level with attention mechanims to focus on the main contributing elements.

HAN architecture is based on an information aggregation method which uses a word encoder for capturing sequence information in a sentence, a word attention mechanism for promoting relevant words and composing sentence representations, a sentence encoder for tracing the sequence of sentences, and a sentence attention mechanism for discarding unrelevant information. A general outline is shown in Figure 6.14.

Figure 6.14: *Architecture of a Hierarchical Attention Network reproduced from Yang et al. (2016).*



The encoder layers are based on GRU (Figure 6.15) which uses gating mechanism to process the state of sequences. The current state h_t is the result of a linear interpolation between the previous state h_{t-1} and the current candidate state $\tilde{h}_t$ (Equation 6.20). The update gate (z_t) balances the amount of past and new information to be used for the current state (Equation 6.21), while the reset gate (r_t) weighs the past state's contribution to the candidate (Equation 6.23). The candidate state $\tilde{h}_t$ is computed as traditional neural networks.

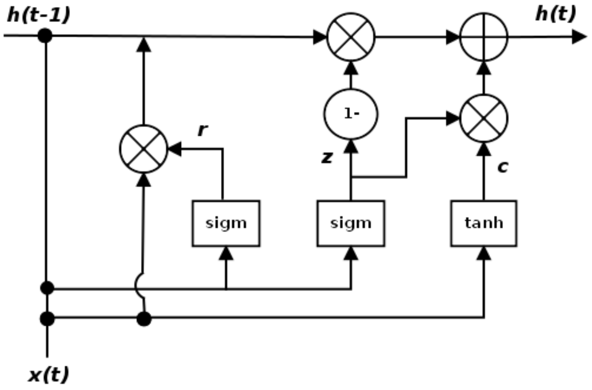
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{6.20}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{6.21}$$

$$\tilde{h}_t = tanh\,(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{6.22}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{6.23}$$

Figure 6.15: *Architecture of a Gated Recurrent Units (GRU).*



As for the attention mechanisms, one-layer FCNN is used to represent a hidden representation of h_t (u_t) (Equation 6.24). In turn, a word context vector u_a is learnt and the weights of words α_t are computed as the similarity values between u_t and u_a (Equation 6.25). Finally, the aggregate vector g (a sentence or document representation) is calculated as the weighted average of current states (Equation 6.26).

$$u_t = tanh(W_a h_t + b_a) \tag{6.24}$$

$$\alpha_t = \frac{exp(u_t^T u_a)}{\sum_t exp(u_t^T u_a)} \tag{6.25}$$

$$g = \sum_t \alpha_t h_t \tag{6.26}$$

6.4.4 Experimentation

The word embedding models described in Section 6.4.2 have been used to represent HUFA documents as sequences of word vectors. Generated contextualised clinical word embeddings in addition to general-purpose ones such as BERT have also been applied. The experimental settings and results are detailed below.

Experimental settings

Deep learning models are computationally expensive and not very robust to extreme data. The processing of each batch is expensive due to the enormous length of the documents, size of the vocabulary, and number of codes, involving a large number of parameters. Therefore, the feature-based transfer experimentation has been conducted on a sample of CIE-10-ES codes instead of the whole population. Specifically, all those with presence in more than 3% of instances (HUFA EHRs) have been selected, comprising a total of 45 CIE-10-ES codes, 37 diagnoses and 8 procedures.

Both RCNN and HAN models have been explored with a direct multi-label approach, using loss function for multiple classes. Despite the selection of the code sample, imbalance techniques for deep learning have been required to yield solid results and avoid learning more representations of the data-dominated class. For example, a random selection based on probabilities proportional to the inverse of the frequency has been used for promoting positive instances in mini-batches and improving convergence.

Besides, we have used the focal loss function proposed by Lin et al. (2017) to avoid the problem of overconfidence related to the cross-entropy loss when predicting positive labels (Kull et al., 2019). Instead, focal loss forces the model to take risk and predict with less confidence at the expense of increasing the likelihood of false positives by reducing the loss for well-classified examples and increasing the loss for hard-to-classify examples. The function is described in Equation 6.27, where y are the binary true values of codes (1 if present and 0 if absent), $\hat{y}$ are the predicted probabilities, and γ is a parameter that balances the loss (higher the value, the lower the loss for well-classified examples). Weights w also proportional to the inverse of frequency have been introduced.

$$FL(\hat{y}_t) = -w(1-\hat{y}_t)^{\gamma} log(\hat{y}_t) \tag{6.27}$$

$$\hat{y}_t = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1-\hat{y} & \text{otherwise} \end{cases} \tag{6.28}$$

Only micro- and macro-averaged F-Score values have been estimated. It does not make sense to apply propensity scored values when only dealing with frequent codes, and neither the comparison of hierarchy-sensitive scores with a limited sample of codes is very informative.

Results

Table 6.4 shows the micro- and macro-averaged F-Score values for the subset of CIE-10-ES codes on the HUFA data set using RCNN and HAN models.

As illustrated, RCNN and HAN with no pre-trained word embedding exhibits similar performance in terms of micro- and macro-averaged F-Score values, but RCNN outperforms in all cases where the model is initialised with pre-trained vectors. Perhaps the way word embeddings are trained with FastText, which handles the record as a single sequence of tokens, is more similar to that used in RCNN than in HAN, which keeps sentences and paragraphs separate during learning. It should be noted that HAN is slower to converge as it encompasses a larger number of parameters.

Some correlation seems to exist between the STS results reported in Table 6.3, with the results of the coding CIE-10-ES. The in-domain vectors SHE achieve higher micro-averaged values and slightly lower macro-averaged values than SUC, suggesting a better generalisation of SHE exclusively in some codes. All new vectors generated with the customised data collection (SCE, SCE-L, SCE-SBWC, SCE-SUC, SCE-CN, CSCE, CSCE-L) outperform the rest.

SCE reaches 72.63 and 68.40 values, which are the second highest F-Scores achieved with RCNN and represent a 3.68% increase over SHE and general purpose embeddings. Although experimentally it has been observed that combining general-purpose and in-domain text generally yields poorer domain representations, sometimes the incorporation of additional general-purpose text during vector generation introduces improvements in the predictive ability of the models by increasing lexical diversity. For example, SCE-L improves the performance of the HAN model, while SCE-L and SCE-CN achieve lower scores. SCE-SUC achieves the highest results in terms of micro-averaged F-Score, 73.48 and 69.35 for RCNN and HAN respectively.

Overall, there is not much improvement in this task when retrofitting techniques are applied (retrofitted SCE and SCE-L) as these are not able to deal with the OOV

	Micro-averaged F-Score		**Macro-averaged F-Score**	
	RCNN	HAN	RCNN	HAN
Baseline	65.32	65.63	60.61	60.42
CN	57.50	52.90	55.00	48.70
SBWC	69.15	64.30	64.35	59.55
SUC	69.70	64.35	66.15	60.80
SHE	70.05	65.15	66.00	60.75
SCE	72.63	67.65	68.42	63.80
SCE-L	72.05	67.91	67.90	**64.02**
SCE-SBWC	71.95	67.10	67.65	63.80
SCE-SUC	**73.48**	**68.40**	**69.35**	63.90
SCE-CN	72.43	67.35	68.33	63.75
Retrofitted SCE	70.40	65.48	66.10	61.46
Retrofitted SCE-L	70.21	65.41	65.89	61.33
CSCE	71.82	66.95	67.65	63.67
CSCE-L	71.34	66.91	67.19	63.69
BERT	67.98	63.24	63.81	58.56

Table 6.4: *Extrinsic evaluation of word embeddings models in ICD coding. RCNN and HAN architectures have been used, evaluating the results with micro- and macro-averaged F-Score values.*

words. Regarding ML, CSCE and CSCE-L do not offer a greater contribution than the representations provided by FastText. It perhaps confirms the assertions of Tenney et al. (2019) about the fact that the main benefits of contextual embeddings over non-contextual ones are more syntactics and morphogrammatics than semantics.

As for BERT, worse results are achieved than with general domain embeddings, both with missing clinical information. This is explained by the increase of parameters, as previously discussed. On the one hand, a representation is generated for each subword of the record (note that the internal tokeniser of the model operates with subwords). On the other hand, the resulting representations are 512-dimensional vectors instead of the conventional 300-dimensional ones, as it encodes more linguistic information.

6.5 Parameter-based transfer

The more closely related the source and target tasks are, the more likely it is that the transfer will be effective. Most of the models used for parameter transfer are language models, as discussed in Section 6.2. The idea is to exploit all the information contained in the model and not just the output layer. In contrast to the feature-based transfer which is based on producing word representations by using freezed models, fine-tuning method relies on training the parameters on the target task. In this sense, we have explored two alternative lines of research: transferring knowledge from language modelling and exploiting hierarchical structure to transfer category information. Such models are expected to transfer these low-level patterns improving the convergence of codes with fewer instances available.

6.5.1 Language Model

We have applied the fine-tuning method to clinical and general-purpose models for CIE-10-ES coding, using the language models described in Section 6.4.4 and BERT. To this end, we have combined the models pre-trained in a LM objective with a new fully-connected layer and a softmax activation function aimed at collapsing the activation functions of the last layer's nodes into the classification labels. Nevertheless, we have not introduced the results in this chapter given the low scores obtained in both models. The reason for this is the limitation of the input due to the fixed size of the models. For example, BERT only supports a maximum of 512 tokens, with each token being a subword. This size is insufficient to represent records from HUFA with more than 1,000 whole words on average.

6.5.2 Inter-levels knowledge

ICD-10 codes are not independent labels but are related to each other through the hierarchical structure of the nomenclature. So far we have used the hierarchical relationships only during evaluation; however, the training can be enriched with such additional information to improve the generalisation of the codes.

We have exploited the inter-level information from CIE-10-ES hierarchy by reusing the internal parameters of the models applied to the classification of non-final codes for the classification of final codes. The idea is to capture more common features shared by codes within the same branch when classifying chapters or groups, and use them as a basis for more specific codes with fewer examples.

The following is a description of the proposal and the conducted experimentation.

Method

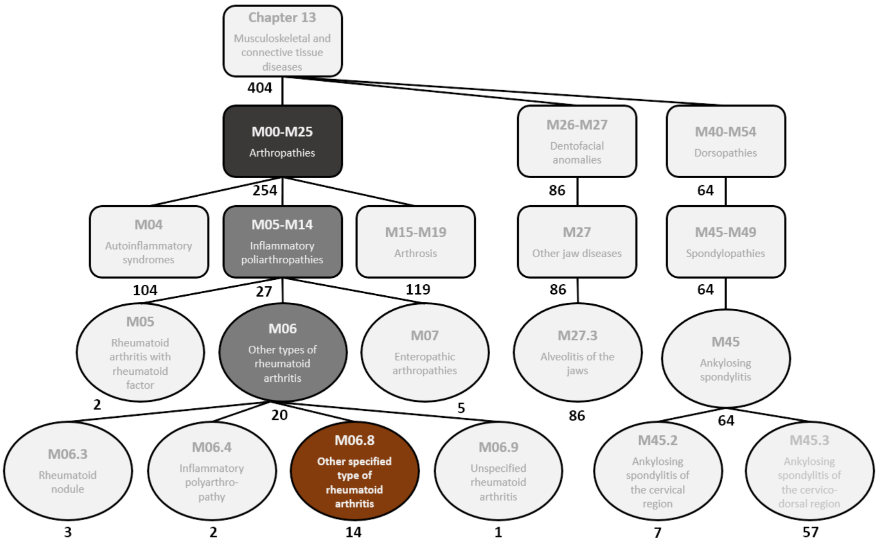
The proposal is to train rare codes using models pre-trained on top categories. The generalisation of codes with few instances is affected by both the lack of examples itself and the dominance of other instances. The aim is to balance training with top categories, which are better represented by aggregating the instances of all dependent final codes (leaf nodes on the same ICD branch).

The proposed method consists of two steps: a first training with top categories and a fine-tuning on the specific codes. More details on the selection of the top categories and the training conditions are given below.

Hierarchical route The depth of the CIE-10-ES hierarchical tree is diverse as in addition to the 3-7 character codes, there are chapters, sections, and sometimes subsections. Each division is based on specific features, e.g. Chapter 7 encompasses all diseases of the eye and its sections such as H00-H05 (Disorders of eyelid, lacrimal apparatus and orbit), H10-H11 (Disorders of conjunctiva), and H15-H22 (Disorders of sclera, cornea, iris and ciliary body) separate the codes according to the affected part of the organ.

The specificity of each chapter and section is also not proportionate. For example, Chapter 19 (Traumatic injuries, poisonings and other consequences of external causes) is more generic and comprises 9,956 codes, while Chapter 3 only deals with 247 diseases of the blood and haematopoietic organs and certain disorders affecting the immune mechanism. For this reason, instead of choosing a non-final code using a fixed distance, we have set a minimum cut-off frequency CF_i assuming that there is some correlation between the training frequency and the specificity of an CIE-10-ES branch. The idea is to pick the first top category that exceeds the training cut-off frequency for a code i, and thus avoid using too generic top categories that prevent proper code specialisation and lead to negative transfers.

Figure 6.16: *Example of selection of top categories for codes according to training frequency. The corresponding code is shown in red and the selected category in black.*


Chapter 13
Musculoskeletal and connective tissue diseases
404
M00-M25
Arthropathies
M26-M27
Dentofacial anomalies
M40-M54
Dorsopathies
254
86
64
M04
Autoinflammatory syndromes
M05-M14
Inflammatory poliarthropathies
M15-M19
Arthrosis
M27
Other jaw diseases
M45-M49
Spondylopathies
104
27
119
86
64
M05
Rheumatoid arthritis with rheumatoid factor
M06
Other types of rheumatoid arthritis
M07
Enteropathic arthropathies
M27.3
Alveolitis of the jaws
M45
Ankylosing spondylitis
2
20
5
86
64
M06.3
Rheumatoid nodule
M06.4
Inflammatory polyarthro-pathy
M06.8
Other specified type of rheumatoid arthritis
M06.9
Unspecified rheumatoid arthritis
M45.2
Ankylosing spondylitis of the cervical region
M45.3
Ankylosing spondylitis of the cervico-dorsal region
3
2
14
1
7
57

We have moved up the CIE-10-ES hierarchical tree for each under-represented final code to retrieve the nearest top categories that exceed the frequency proposed in Equation 6.29. We have used a sigmoid function to map frequencies to a range of values. $CF(f_i)$ is the cut-off frequency for code i, where f_i is the training frequency, α is the maximum cut-off frequency, β is a parameter for setting the minimum cut-off frequency, and γ is another parameter that controls the saturation rate at maximum frequency.

$$CF(f) = \frac{\alpha}{1 + e^{\beta - \gamma f_i}} \tag{6.29}$$

Empirically we have found that a useful range of frequencies would be [15,40], thus we have fitted the function so that $alpha$ would be 40, $beta$ would be 0.61, and $gamma$ would be 0.1. Figure 6.16 shows an example where section M00-M25 is selected for code M06.8. With 14 instances, the cut-off frequency would be 27.51, so neither category M06 nor section M05-M14 would pass the restrictions.

Training step The same setting proposed in Section 6.4.4 using the SCE-SUC representation and the RCNN model has been implemented, as it was the combination of vectors and model that achieved the best results.

A first step consisting of the training for the classification of the top categories associated with the final codes has been performed. In this way, the last layer will produce the probabilities for each category. The entire model is expected to capture common features, so that if a EHR contains patterns related to a respiratory disease, it will be reflected in the output that groups the codes for such diseases.

Then, the last layer is replaced by a new layer that will generate the probabilities for each final code, and a new training is conducted. The idea is to start from general features and force the network to discriminate which of these patterns are related with the final code. However, the objective is also to prevent the network from forgetting what it has learnt, so a shorter training is desirable. Two approaches have been explored in relation to prioritisation of "*remembering*" (Inclusive perspective) or "*forgetting*" (Discriminative perspective):

- Inclusive perspective. The second training is performed giving greater weight to positive instances, which have already been learned in the previous step. In this way, learning is focuses on the previous low-level features applying minor modifications for preventing the misclassification of examples related to adjacent codes.

- Discriminative perspective. Conversely, no weights are applied in the second training, so that misclassified instances produce higher loss scores and further adaptations are made to the final labels. The model is encouraged to be more confident and mute predictions.

Experimentation

The discussed proposals have been explored on the HUFA corpus and the associated CIE-10-ES codes. The experimental settings and the results achieved are described below.

Experimental settings We have randomly selected a subset of 70 infrequent CIE-10-ES codes present in both the test and training HUFA data sets. Such a subset comprises codes with training frequencies of 1, 2, 3, 5, 8, 10, and 15, with a total of 10 codes for each frequency.

We have selected non-final codes for each of the 70 final codes in order to explore the fine-tunning approach described above. In total, there are 64 top categories since 12 codes share common parents. The selected categories have an average of 174 instances, with a standard deviation of 258, which means that there are categories with frequencies very close to the cut-off frequency and others that are close to 1,000 instances due to some wide branch. Overall, the depth of the classes goes from 4.85 ± 0.8 to 3 ± 1.

Figure 6.17 shows the average number of instances and level of depth in the CIE-10-ES hierarchy for codes and categories grouped by code training frequency. As illustrated, there is a correlation between the number of instances and the depth in the tree for categories. Specifically, the categories selected for the code group with frequency 5 comprise the highest number of instances, with more than 400 on average, and the lowest level of depth, with an average of 2.7, which means that these are sections and sub-sections.

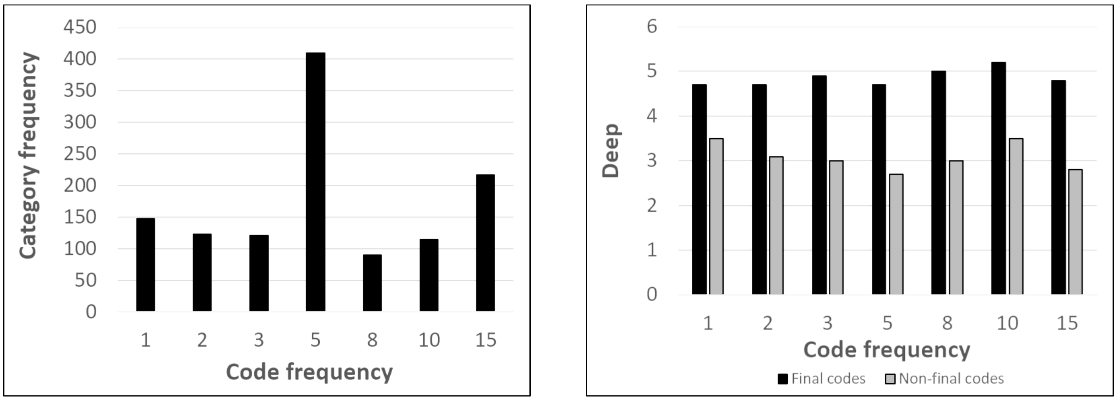
Finally, the two proposals described in the Section 6.5.2 on the sample of codes and their corresponding categories have been explored. All codes selected for experimentation are underrepresented, so it makes no sense to show propensity scored values as the codes would all be weighted with similar values. At the same time, only 5 of the 70 codes belong to the same group, so the measures based on partial matches will be practically the same. Therefore, only micro- and macro-averaged Precision, Recall, and F-Score values have been applied in the evaluation.

Results Preliminary results have been explored by directly training on the selected subset of codes, which could be considered as the baseline in this experimentation, and the proposals using hierarchical information, both of which are listed in Table 6.5.

Both approaches improve on the initial proposal, one focusing on Precision and the other on Recall. The inclusive approach almost doubles Recall in both micro- and macro-averaged, while the distriminative perspective improves micro-averaged Precision by more than 100% and macro-averaged Precision by almost 40%. The latter difference between the micro- and macro-averaged values reveals that the significant improvement is only applicable to some codes. In this line, the inclusive perspective achieves the highest macro-averaged F-Score result, with improvements more spread across the whole subset. In contrast, the discriminative perspective achieves the highest micro-averaged value, with improvements more concentrated in a few codes.

Figure 6.18 shows the micro- and macro-averaged Precision, Recall, and F-Score

Figure 6.17: *Differences in number of instances and depth in the CIE-10-ES hierarchical tree between the final codes and the selected parent categories. Codes have been grouped according to training frequency.*



(a) *Average number of instances associated with the categories of each code group*

(b) *Average depth of the codes and categories of each group*

Score		Baseline	Inclusive perspective	Discriminative perspective
Micro-average	P	8.22	8.27	**17.09**
	R	20.69	**41.72**	20.69
	F	11.76	13.80	**18.72**
Macro-average	P	9.21	11.75	**12.79**
	R	18.09	**38.94**	18.15
	F	9.40	**15.26**	13.02

Table 6.5: *Estimated summary of the dimensions of the pre-trained word embeddings used in the experimentation.*

values per training frequency. All groups except the codes with frequency 5 achieve higher values of micro- and macro-averaged Precision with the discriminative perspective. The inclusive perspective seems to follow the same trend as far as the group with 5 instances is concerned. This group contains the most abstract categories and the largest number of instances from other codes, which might indicate that the model has had difficulties in the generalisation by capturing the common features of a group that is too diverse.

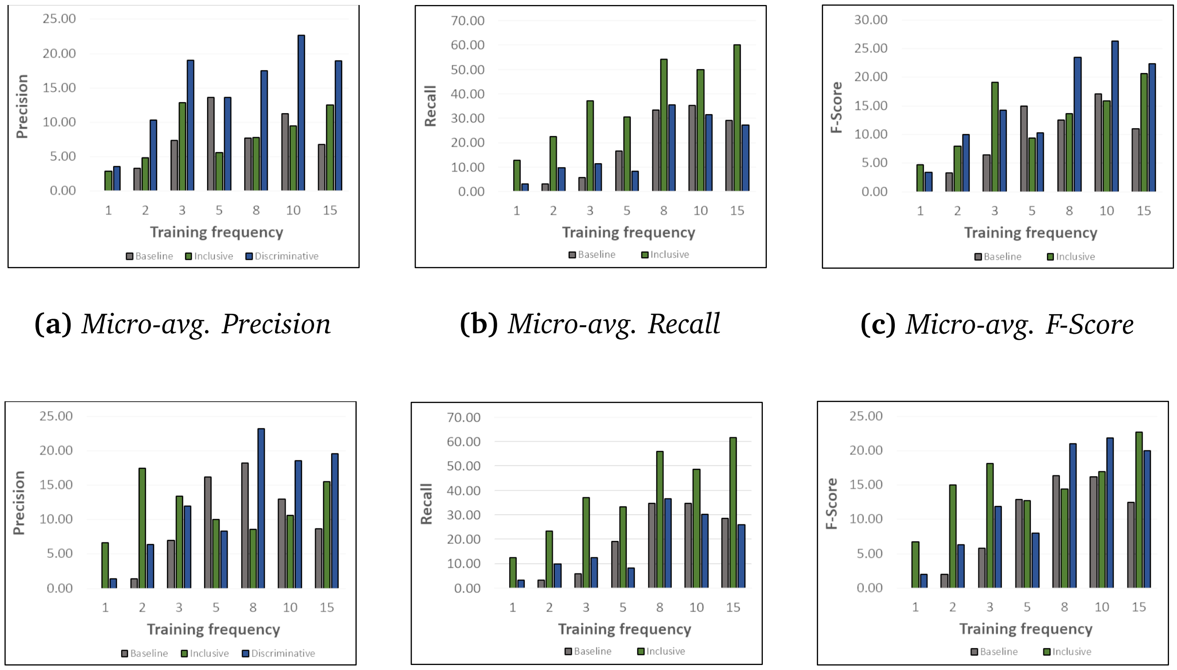
Meanwhile, the largest decrease in macro-averaged Precision is found with the inclusive perspective in the group of frequency 8, which is precisely the group with the fewest instances. There is an increase in the number of false positives, which could be due to the incapacity of the model to forget general information and specialise by reducing the relevance of new information with relatively few instances. As mentioned above, Precision values vary between the micro and macro averages, while Recall remains roughly constant. So the differences between micro- and macro-averaged F-measure are determined by the variations in Precision.

6.6 Discussion and concluding remarks

Transfer learning methods have become one of the most popular lines of research in the recent years, with papers on the application of general representations dominating the literature and papers on model reuse attracting the most attention. However, the limited access to clinical data has significantly reduced the release of such resources, especially for Spanish tasks. For this reason, we have explore the generation and application of these resources consecutively. The outcomes of the proposals are discussed below.

Transfer learning guidelines Broadly speaking, transfer learning has been found to be a recommended technique to deal with ICD data sets. Positive transfer has

Figure 6.18: *Micro- and macro-averaged Precision, Recall and F-Score values broken down by training frequency for the baseline, inclusive, and discriminative approaches. The micro-averaged values are shown above and the macro-averaged ones below.*



(a) *Micro-avg. Precision* **(b)** *Micro-avg. Recall* **(c)** *Micro-avg. F-Score*

(d) *Macro-avg. Precision* **(e)** *Macro-avg. Recall* **(f)** *Macro-avg. F-Score*

been achieved by using information from other datasets with different features and distributions (cross-lingual approaches), from huge collections of general, medical, and clinical domain text, and from the ICD hierarchy itself (fine-tuning technique). Nevertheless, performance improvements are not always achieved, e.g., some authors such as Atutxa et al. (2019) conclude that the use of external pre-trained embeddings in its proposal resulted in a performance decrease. Another example would be the parameter-based transfer experimentation in Section 6.5.1, which leads to negative transfers with results that fail to outperform non-transfer approaches.

Instance-based transfer has been explored through the application of MT techniques to multi-lingual datasets and the use of weights to balance the contribution of instances in feature extraction and learning. We have observed that independent enrichment of codes, avoiding interference in the rest, helps to achieve better generalisation, especially for under-represented codes. Models for frequent codes do not need many more examples to improve their performance, while the increased lexical diversity in models for rare codes, most of the standard and those composing the tail of the extreme distributions as discussed in Section 5.4, leads to a better ability to capture low-level features.

Transfer learning via feature representations has also been explored. For this purpose, subword-level, contextualised and non-contextualised word embeddings have been generated using medical and clinical resources. The overall result is a significant improvement in ICD coding over general-purpose word embeddings. The performance of non-contextualised embeddings in ICD coding correlates with their quality, measured as the ability to estimate similarity between multi-termed concepts. There are no major differences between contextualised and non-contextualised embeddings, which could be due to the lower ambiguity of the medical records domain.

Finally, a parameter-based learning experimentation has been conducted by exploring parameter sharing between language models and code classification models. The results have not been satisfactory mainly due to the size limitations of the models. Parameter sharing between non-final and final classification models has also been explored with the aim of exploiting the hierarchical information contained in the ICD standard. As a result, better generalisation of models associated with less represented codes has been achieved by forcing these models to learn more general feature detection, characterising superior categories.

While the cross-lingual method based on the addition of instances and the inclusion of hierarchical information aims mainly at improving the generalisation of rare codes, the use of word embeddings is transversal, favouring the representation of all codes. It remains to explore relational-based transfer by capturing the relationships between code descriptions and document representations. As for Research Question "*Which transfer learning methods are easily applicable to ICD-10 coding and which ones are most effective in improving inference?*" (RQ 4), we could conclude that representation-based transfers have led to the most significant overall improvements, followed by parameter-based transfers. Although individual experiments have been carried out, all these methods are alternatives and could be complementary. In fact, the representations of Section 6.4.4 have been used for the experiment in Section 6.5.2. In fact, analysing the overlap between knowledge transferred by the four types of methods is a future task.

CHAPTER

7

INTEGRATION OF TECHNIQUES

Content

This chapter explores the integration of the multiple techniques used in the previous experiments to deal with data sparsity, instance imbalance, and generalization issues simultaneously. For this purpose, the impact of each one on performance is assessed in accordance with the different stages of an approach: data, representation, and inference.

The following simultaneous objectives are pursued:

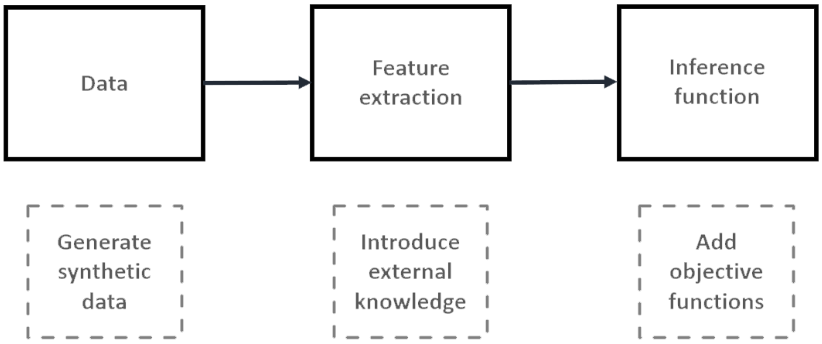
- Correct the data imbalance for supervised methods by means of data augmentation techniques and extreme algorithms.
- Use pre-trained word embeddings to improve convergence during learning by introducing external knowledge to the task.
- Increase the coverage obtained with supervised methods by supplementing predictions with unsupervised methods.

7.1 Introduction

Only a few approaches tackle more than one key aspect of ICD coding simultaneously, as discussed in Chapter 2. Such aspects can be addressed in the different elements of an approach, as illustrated in Figure 7.1, i.e., varying the training data, improving the representation, and modifying the inference function. These are generally complementary processes that can be easily integrated into a single approach. Some of the examples present in SOTA are the augmentation of data together with BERT representations (Biseda et al., 2020; Ollagnier and Williams, 2020) and models (Garcıa-Santa and Cetina, 2020). However, the joint implementation of similar techniques may require additional mechanisms, such as assembly methods. For example, approaches combining supervised and unsupervised methods are common in SOTA. Thus, some authors have implemented independent processes by using supervised methods to predict the most frequent codes and unsupervised methods to predict the least frequent codes (Pakhomov, Buntrock, and Chute, 2006; Patrick, Zhang, and Wang, 2007). Instead, other authors have explored dependent processes by generating features for supervised methods via unsupervised ones (Crammer et al., 2007; Pereira et al., 2013; Zweigenbaum and Lavergne, 2016).

In this chapter we explore a preliminary approach to combine the most promising techniques described in the previous chapters and thus answer Research Question 5: "*How could alternative ICD-10 coding approaches be combined to tackle scarcity, imbalance, and generalization constraints?*" (**RQ 5**). We have not included a section of related works of ensembles as searching for the best way to integrate all the methods into a single system is beyond the scope of this thesis and is planned for future work. Instead, in this chapter we only want to explore the possibility of effectively combining

Figure 7.1: *Outline of possible improvements in the different components of an approach.*



all the elements described below as a proof of concept. For this purpose, we use the AttentionXML model as baseline and propose a first approach consisting of:

- ***Data augmentation methods*** to reduce imbalance.
- ***Representations using pre-trained vectors*** to improve generalization by introducing external knowledge to the task.
- ***Combinations of supervised methods*** to increase robustness.
- ***Complementary uses of supervised and unsupervised methods*** to exploit learning while dealing with data sparsity.

Such data augmentation and representation techniques are described in Sections 5.3 and 6.4, the integration of unsupervised methods follows the same line as described in SOTA (see Section 2.2.1, second paragraph), and the combination of supervised approaches has been done using the widely extended technique voting. This is a first approximation, so more advanced formulas are still need to be found.

7.2 Assembly method

The setting detailed in Chapter 5 for the AttentionXML model has been used as the basis for the final proposal as it has achieved the highest micro-averaged and good macro-averaged results. Subsequently, each of the aspects discussed in the introduction have been examined: techniques focused on improving training data, representation, and code inference.

Training data We have applied the substitution-based data augmentation method described in Section 5.3.1 to increment the training data volume. In this case, we have not dealt with binary models, so the whole data set is used during learning instead of separating the synthetic data by labels.

	Relevance	
	AttentionXML	DeepXML
More than 40	1	0.2
Between 5 and 40	0.6	0.5
Less than 5	0.3	0.8

	Probability		Frequency
	AttentionXML	DeepXML	
I10	0.8	0.6	1840
I25.3	0.3	0.3	3

Table 7.1: *Example of frequency range weighted voting.*

Representation Instead of assigning random vectors, we have initialised the model with the word vectors pre-trained in the domain, which are described in Section 6.4. This is intended to achieve greater convergence during learning.

Code inference We have implemented two assembly methods to combine the predictive capability of multiple ICD-10 coding functions: voting and concatenation.
On the one hand, we aimed to exploit the variability of the performance of each approach as a function of the code frequency. In Section 5.5, we noted that every algorithm optimises code prediction in a different frequency range, so an appropriate combination exploiting the overlap between approaches could be more robust than stand-alone algorithms. For that reason, we have applied the widespread voting methods, computing the relevance of the ICD-10 codes for a document as the weighted sum of the probabilities of the predictions. Specifically, predicted code probabilities are scaled differently depending on the approach that produces it and the frequency range in which it is contained. Equation 7.1 describes the calculation of the final probability $P(c)$ of a code c, where n is the number of approaches, i represents the corresponding approach, $P_{i,c}$ is the probability corresponding to the approach i and the code c, and $\alpha_{i,c}$ is the weight assigned to the approach i for the training frequency range comprising the code c. $\alpha_{i,c}$ values have been manually assigned according to the approach performance for each frequency range.

$$P(c) = \frac{\sum_{i=0}^{n} \alpha_{i,c} \cdot P_{i,c}}{\sum_{i=0}^{n} \alpha_{i,c}} \tag{7.1}$$

Hence, the weight for AttentionXML will be higher than the one for DeepXML at high frequencies, but lower at low frequencies. For example, if we had only the AttentionXML and DeepXML models and they yielded the probabilities $P_{i,c}$ shown in the Table 7.1 for codes I10 and I25.3, the final probability for both would be $P(\mathit{I10}) = \frac{1 \cdot 0.8 + 0.2 \cdot 0.6}{1+0.2} = 0.77$ and $P(\mathit{I25.3}) = \frac{0.3 \cdot 0.3 + 0.8 \cdot 0.3}{0.3+0.8} = 0.3$.

On the other hand, we have alternated between supervised methods for the most frequent codes and unsupervised methods for infrequent codes. The idea is to complement the higher accuracy resulting from the generalisation of coding on sufficient

Approach	Less than 5 occurrences	More than 5 and less than 40 occurrences	More than 40 occurrences
AttentionXML	0.1	0.6	1.0
B-XGBoost	0.1	0.4	0.8
B-SVM	0.2	0.8	0.6
Bonsai	0.3	1.0	0.3
Parabel	0.3	1.0	0.3
DECAF	1.0	0.1	0.1
DeepXML	1.0	0.1	0.1

Table 7.2: *Weights assigned to each approach as a function of the code frequency range.*

examples with the coverage offered by the structured information created by experts. To this end, the lowest probabilities provided by voting are discarded, reducing the unconfident predictions. In this way, each document is supplemented with the most relevant codes yielded by the unsupervised method, which are placed in the last positions of the document ranking.

7.3 Experimentation

The same HUFA subset as in previous chapters (Sections 5.3.3, 5.4.2, 6.4.4, and 6.5.2) has been used to achieve comparable results. The experimental setting and results of integrating each technique into the selected baseline are detailed below.

7.3.1 Experimental setting

An ablation test combining all methods has been performed to analyse the impact of each one. To this end, the AttentionXML model and the setting described in Section 5.4.2 have been used as the baseline. Subsequently, we have progressively integrated each method in the order shown in Figure 7.1. Thus, we have used the synthetic data set produced by Lexical Substitution and described in Section 5.3.1 for data augmentation and SCE-SUC word embeddings described in Section 6.4 for pre-initialisation. As for voting, we have selected the best performing approaches in each frequency range according to Figures 5.23 and 5.24: AttentionXML, B-XGBoost, Bonsai, Parabel, DECAF, and DeepXML. Table 7.2 shows the heuristically fixed weight matrix associated with each approach and the different frequency ranges. Finally, a minimum threshold of 20% confidence has been set for alternating with unsupervised methods.

7.3.2 Results

Micro- and macro-averaged values at 10 for the AttentionXML model and each of the additional proposals are shown in Table 7.3. Each column represents the results of adding that joint method given by the previous column. While there is a consistent positive progression in practically all micro- and macro-averaged values, the matches move down in the final ranking with every additional method, as illustrated by the negative trend of the nDCG values. Overall, the micro- and macro-averaged F-Score values for the final ensemble increase by 12% and 60% respectively over the baseline. In terms of percentage improvement, the largest increases in micro-averaged Precision and Recall are produced by integrating the unsupervised and data augmentation methods respectively. In contrast, the largest increases in macro-averaged values are exclusively achieved with data augmentation techniques.

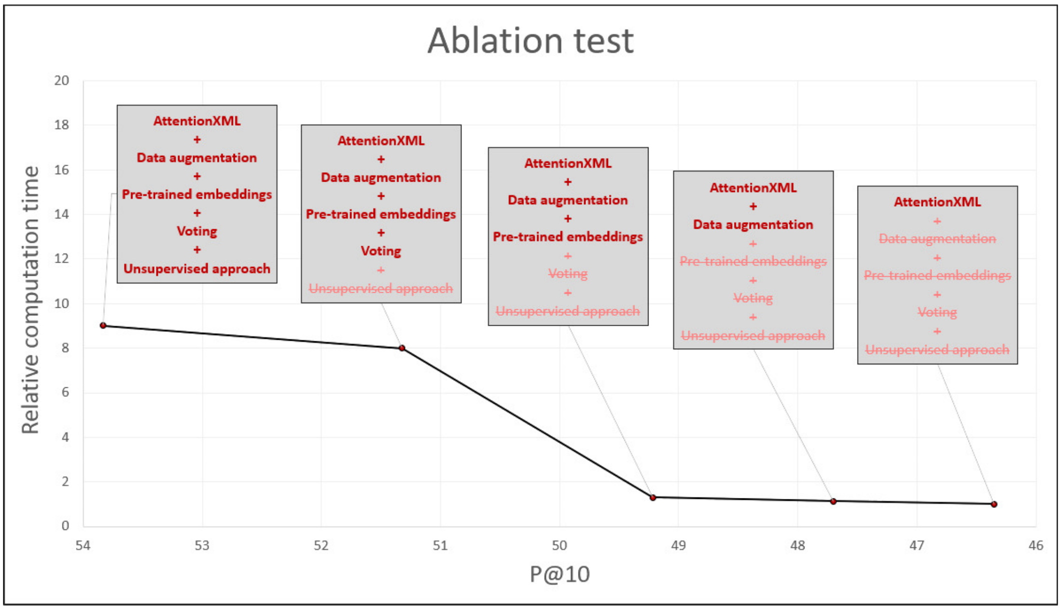
In turn, Figure 7.2 illustrates the evolution of the micro-averaged Precision at 10 values according to the removal of the methods (following the order of Figure 7.1). Each point shows the performance in terms of P@10 (X-axis) and relative inference time (Y-axis). Improving inference by including IR to reach new codes and voting to increase robustness implies a significant increase in computational time, around 8 times higher. Although the combination of different techniques achieves the largest increases on the X-axis, the use of binary methods such as XGBoost could hinder a real system with hardware limitations requiring very short response times. In any

		Baseline	+ Data augmentation	+ Pre-trained embeddings	+ Voting	+ Unsupervised method
Micro	P	46.35	47.70	48.22	51.32	**53.83**
	R	43.01	43.75	44.13	45.98	**46.74**
	F	44.62	45.64	46.08	48.50	**50.04**
	P_S	54.47	55.82	56.84	58.77	**61.32**
	R_S	50.67	51.45	52.22	52.96	**53.65**
	F_S	52.50	53.55	54.43	55.71	**57.23**
	PSP	41.05	43.97	44.53	46.32	**48.51**
	PSR	31.13	33.21	34.01	35.46	**35.98**
	PSF	35.41	37.84	38.56	40.17	**41.32**
	PSP_S	49.34	52.53	52.96	54.18	**56.80**
	PSR_S	38.64	40.99	41.55	42.73	**43.36**
	PSF_S	43.34	46.05	46.57	47.78	**49.18**
Macro	P	8.54	12.98	13.22	14.03	**14.49**
	R	10.88	14.04	14.84	15.96	**16.26**
	F	9.57	13.49	13.98	14.93	**15.33**
	P_S	10.90	16.45	16.93	17.35	**17.82**
	R_S	12.69	16.69	17.68	18.62	**19.16**
	F_S	11.73	16.57	17.30	17.96	**18.47**
	PSP	6.67	10.76	10.48	11.59	**11.83**
	PSR	8.61	11.65	11.71	13.41	**13.65**
	PSF	7.52	11.19	11.06	12.43	**12.68**
	PSP_S	8.62	13.77	13.55	14.45	**14.89**
	PSR_S	10.13	13.98	14.01	15.76	**16.02**
	PSF_S	9.31	13.87	13.77	15.08	**15.43**
Order	$nDCG$	83.80	84.07	**84.25**	77.09	75.81
	$nDCG_S$	87.39	87.77	**87.86**	82.38	81.46
	$PSnDCG$	73.34	**74.86**	73.74	70.82	67.16
	$PSnDCG_S$	76.50	**78.24**	76.91	76.86	73.40

Table 7.3: *Micro- and macro-averaged values for the ablation test.*

case, the response time can be reduced by limiting the type of proposal during the aggregation of predictions.

Figure 7.2: *Ablation test: micro-averaged Precision at 10 versus inference computation time.*



7.4 Discussion and concluding remarks

As mentioned, we have not conducted a comprehensive review of the SOTA in order to propose innovative methods of assembly. Instead, we have explored conventional ensemble of algorithms and techniques addressing each of the problems identified in the introduction of this thesis (Chapter 1). Next, we discuss the impact of each of the integrated methods and analyse the ablation test to answer the Research Question "*How could alternative ICD-10 coding approaches be combined to tackle scarcity, imbalance, and generalization constraints?*" (**RQ 5**).

Ensemble The final proposal reaches significant improvements of 12% and 60% in both micro- and macro-averaged values by reducing the negative effects associated with the data distribution and the abstract nature of codes. It is estimated that the inclusion of pre-trained vectors and the voting method mainly improve micro values, while data augmentation and unsupervised methods increase macro values. Data augmentation mainly contributes to the improvement of minority codes, which results in a significant increase in macro-averaged scores. The pre-trained vectors help the generalisation of all codes, perhaps improving a bit more the more frequent ones as they tend to use more common concepts. The voting method relies on overlapping codes, so it favours the most common ones; however, we slightly balance the improvement by including weights for the approaches by frequency ranges. Finally, the unsupervised method is used to identify underrepresented or non-represented codes, which has a direct but limited impact on the macro-averaged values. In fact, the table reflects greater improvements in micro than in macro values, which indicates that minority codes are missing in the final inference and points to the need to explore better ways of combining supervised and unsupervised methods, as discussed below. This is a first exploration aimed at validating the set of methods evaluated individually, so no further effort has been made to try to optimise the performance of the ensemble. Thus, only some of the techniques explored in the previous chapters have been applied, avoiding complex adaptations, while the integration of all the methods explored is pending as part of future work. The proposed method is far from reaching the theoretical maximum reflected in Chapter 3, so the margin for improvement is large, both in the individual methods and in the integration process. As for the incorporation of IR into the ensemble, we expected a larger increase in values. Analysing the results in Chapter 4, we find that the $nDCG$ value of IR is low compared to that of the supervised models, which means that the matches tend to be in the middle positions. The way to incorporate IR is by adding the first codes at the end of the document ranking, so the possible improvement decreases considerably. A more effective way of exploiting the coverage of an unsupervised system remains to be found.

Also, it has not been possible to improve the position of matches in the final ranking. In particular, voting methods introduce noise by displacing matching codes. It has been observed that some unmatched codes are repeated in the different approaches, leading to erroneous promotion. Another relevant aspect is the smaller improvement of partial matches with respect to exact matches, which can be seen in the 12% increase in P versus the 9% increase in P_S. The smaller gain in partial matching is produced by introducing voting, as the actual mechanism for promoting overlap between methods is based on exact matches. Therefore, it is pending to explore a voting method based on partial matching, aggregating the probabilities of each code based on similarity.

In response to Research Question 5, we have integrated 4 techniques into a single system by modifying the training data, introducing new representations, combining methods by aggregating probabilities by frequency values, and including unsupervised predictions for documents with unreliable codes. Undoubtedly, the voting and data augmentation methods are the ones that introduce the greatest improvement in micro and macro scores respectively. However, they are the most computationally expensive methods in terms of training time, which in principle is not detrimental to a real-time system. Pre-trained vectors are costly in terms of the large amount of data that needs to be collected and do not provide much improvement in this area. Finally, unsupervised methods are not providing all the relevant information they contain, so there is margin for improvement in this direction.

CHAPTER

8

CONCLUSIONS AND FUTURE WORK

Content

This chapter compiles the conclusions, pointing out contributions and lines of future work. It also lists all publications produced as a result of this research.

8.1 Discussion and concluding remarks

ICD coding is a key task in the clinical information flow to ensure the interoperability of the information in the EHRs, usually textual. Such a task requires some degree of automation, but effective modelling involves unsolved computational challenges: data scarcity, label imbalance, scalability, and limited generalisation. In this sense, commercial software for computer assistance has to support coders in complex, real-world environments, so are mainly driven by data scarcity due to strong access constraints. These approaches therefore usually rely on unsupervised methods to achieve a higher coverage. In contrast, the trend in the research community has been to explore supervised methods in more controlled environments, focusing on dealing with some of the other challenges. To the best of our knowledge, there are few approaches in the literature that do not include simplifications (in terms of the number of codes, size of records, etc.) and none that address the problem from multiple perspectives. With this background in mind, we have approached ICD-10 coding with the full complexity (without simplifications) by exploring each of these challenges individually in order to propose an ensemble that combines the best attributes of the proposed techniques.

Firstly, the scarcity of examples has been tackled by unsupervised methods, applying lexical and semantic similarity. The semantic approach performs better as it captures homonymy relationships, dealing with specificity differences. In general, the results obtained using matches between codes and textual evidences show a competitive performance in less verbose EHRs, such as death certificates, but the predictive ability drops in long EHRs, where code assignment does not exclusively use criteria based on meaning matching. While unsupervised methods are not competitive in overall predictive accuracy due to simplicity, the independence of biases and other constraints associated with data collections contributes to the considerable improvement in the prediction of minority codes. Therefore, we conclude that unsupervised methods are effective not as main processes but as complementary methods to other processes, typically supervised.

Secondly, we have addressed imbalance by means of data augmentation methods and XMTC algorithms. The application of general-purpose MT methods to change sentences and expand the number of examples (Back Translation) for underrepresented codes does not show significant improvements due to poor translation quality; appropriate in-domain MT methods are necessary to effectively expand the observed lexicon. In contrast, lexical substitution based on interchangeable expressions, such as synonyms and hyponyms, effectively introduces permutations to generate auxiliary EHRs. Joint training by weighing the impact of these new EHRs improves the prediction of rare codes. Increasing the number of examples for training the models is detrimental to the scalability of the system. Therefore, we have also explored XMTC algorithms, which employ unbalancing techniques such as subsampling and co-dependency capture, while reducing the computational cost. Among all the algorithms used, we highlight the PLT-based algorithms for the global performance, both for frequent and infrequent codes, based on the exploitation of hierarchical information. In addition, the strategy of splitting feature or label spaces leads to the greatest reduction in inference times. Nevertheless, examining the scores in more detail, we find that different types of representations and strategies maximise the inference of codes with different training frequency ranges, which can be exploited in an ensemble.

As for limited generalisation, we have explored instance-, representation-, and parameter-based transfer learning methods to leverage task-external knowledge and improve semantic abstraction. Cross-lingual approaches based on MT techniques for training with EHRs in multiple languages rely on higher lexical diversity and significantly increase Recall; however, as we have already noticed using Back Translation, MT methods not adapted to the domain frequently introduce erroneous expressions. The limitations of applying methods similar to the one proposed are given by the availability of data in multiple languages and domain-specialised translation methods. In the case of the explored instance-based transfer learning methods, we have used a collection with EHRs in multiple languages and a translator trained on medical text. In turn, the use of pre-trained representations generally contributes to achieving greater convergence during learning. Domain representations provide a better characterisation of the records, resulting in higher quality features and implying a greater increase in predictive capability; nevertheless, a limited injection of general knowledge into learning representations yields more complete representations, which contributes to better coding. In contrast, vector generation via Language Models does not yield significantly improved results. Contrary to initial assumptions, a growing number of researches point out that such models do not capture semantics better, but introduce more types of linguistic information into the representations, such as syntactic information. The possibility of using representation-based transfer learning methods is given by the availability of in-domain data.

Following the line of improving generalisation, parameter-based transfer methods that rely on the hierarchical structure of ICD-10 have been explored. Pre-training on higher categories helps models to recognise and process common features, facilitating the learning of less represented codes. The idea is that in the case of distributions where there are few positive and many negative examples, it is more effective to capture information about which specific cases are not appropriate from group information than to learn from scratch common patterns for particular positive instances.

Despite the cases where there has been a positive transfer learning, transfer does not always increase performance. In our case, the distribution of ConceptNet vectors, which are representations pre-trained on general knowledge data, complicates the adaptation of meanings to the domain. In turn, BERT representations expand the number of training parameters, increasing the size of texts by processing subwords and the length of representations by yielding vectors with more dimensions, so that the models used are unable to improve their convergence. Another example is given by the limitation of the size of the EHRs imposed by the pre-trained language models BERT and AWD-LSTM.

Finally, some of the proposed methods have been combined into an ensemble while keeping the overall computational time. For this purpose voting for increasing the overall predictive ability, data augmentation and XMTC algorithms for reducing the imbalance, XMTC algorithms also for decreasing the computational complexity, and semantic similarity based methods for dealing with data sparsity have been used. In the ablation test conducted, the contribution of each method to the increase of micro and macro values is visible, while nDCG values decrease with the unsupervised method and voting, suggesting worse positions for matches in the ranking of predicted codes. This ensemble demonstrates the possibility of applying multiple SOTA techniques to address the four main ICD-10 coding challenges.

Having explored each of the coding points described in the thesis and compared some of the corresponding approaches, we can answer the first Research Question: "*Which are the best techniques for approaching ICD-10 coding in response to the challenges posed by the task?*" (**RQ 1**). The scarcity of examples has been addressed only by unsupervised methods although there are SOTA zero-shot approaches based on relational-based transfer learning that have not been explored due to the poor published results. This challenge has also been addressed indirectly by instance- and parameter-based methods, but we have found that unsupervised methods are more effective for code inference with sparsity of examples. However, we have not found an effective way to combine such methods with supervised ones. Regarding imbalance, we have implemented data augmentation methods and XMTC approaches. The instance-based method can also be considered for the challenge. Although these methods are not mutually exclusive, the XMTC algorithms are the ones that provide the best performance. Similarly, scalability has been addressed by XMTC algorithms and representation- and parameter-based methods. Again, XMTC algorithms provide the largest contribution. Finally, generalisation challenges have been addressed by instance-, representation-, and parameter-based methods. No single comparable experiment has been performed, so there is no comparative evidence. However, at a guess, it seems likely that the parameter-based method that exploits common features within the hierarchical structure will provide the most progress. In essence, computer-assisted approaches for ICD coding can be improved by targeting the specific characteristics of the task.

8.2 Contributions

We hope that all the ideas presented in this research can lead to future new ideas for significant improvements in computer-assited ICD coding. For this reason, we highlight below all those contributions that we consider most relevant to the area, organising them by analyses, resources, and proposals.

Analyses

Firstly, we have provided an analysis of the task, identifying the main challenges for ICD coding, which we have not found in the literature, and placing the task in the XMTC research area. To this end, we have carried out an in-depth review of the SOTA by identifying the type of method used for coding. A brief overview is offered in Chapter 2. Furthermore, we have conducted a comparative analysis of unsupervised, conventional, and XMTC algorithms adapted to the coding task. These summaries are intended to be useful in describing the background of the ICD coding task for future researchers.

Resources

In terms of resources and auxiliary materials, the main data collection used in this thesis cannot be shared due to data restrictions, but lexical resources, in-domain data collection, and other tools have been released. Firstly, several domain resources have been generated for pre-processing: lemma lists and a custom lemmatiser, dictionaries of synonyms and related words, and dictionaries of pertanyms. The code for the described preprocessing can be downloaded from the link at the bottom of the page[1]. A rule-based anonymiser[2] has also been proposed. Besides, a structured digital version[3] of the coders' manuals (Alphabetic Index and Tabular List) has also been produced to extract some of the terminology not contained in the main descriptions. In terms of other data, a collection of PhD theses in medicine and other related health disciplines has also been produced[4], which has been used for the generation of clinical word embeddings[5] and language models. At last, we have proposed new evaluation metrics according to the ICD coding attributes to provide a more complete picture of the

[1] https://github.com/m-almagro-cadiz/spanish-clinical-preprocessing.git

[2] https://zenodo.org/record/5148968#.YQRIQ477SUk

[3] https://zenodo.org/record/5148885#.YQQ9Io77SUk

[4] https://zenodo.org/record/5148872#.YQQ7FI77SUk

[5] SCE and SCE-SUC word embeddings have been released at https://zenodo.org/record/5149010#.YQRa3HX7RH4

results (the code can be found in the link below[6]) All these resources and tools have been designed with the aim of facilitating research in this area.

Proposals

As for the proposals, we have explored the impact of different types of label representation on unsupervised methods in Chapter 4. Following this idea, we have implemented a new semantic similarity-based method that outperforms conventional TF-IDF-based methods by introducing homonymy relations. We have also explored two data augmentation methods for increasing lexical diversity in Chapter 5, relating the impact on performance to the quality of the resources used. The same chapter also contributes numerous implementations for ICD coding, with special focus on XMTC. Moreover, a crosslingual approach that fuses collections of data in different languages and reduces the biases associated with data sets has been described in Chapter 6. In this chapter, deep learning models based on pre-trained word embeddings for ICD-10 coding are also proposed. Besides, a sequential method for capturing hierarchical information and improving generalisation is suggested. Finally, an ensemble incorporating several of the proposal is presented, beating all the analysed scores except those measuring the position of the matches in the output rankings.

8.3 Future lines

Despite the time dedicated to this research, we have explored only a small fraction of all the potential lines. Search for new clinical resources to enrich the representation of codes and records is still pending.

One possible future line would be to use etymological dictionaries to break down Greek and Latin words into elements with independent meanings such as prefixes, lexemes, and suffixes. In this way, we would have greater specificity in texts and greater control to homogenise the information derived from the less frequent and more specialised words, thus discarding non-relevant information. For example, this would allow us to distinguish the words "*febrile*" and "*afebrile*" only by the information contained in the prefix "*a*".

Another pending line of research is the improvement of unsupervised methods. In the case of the semantic proposal, we want to combine the relations between meanings provided by SNOMED CT with the distributional semantics derived from large corpora. To do so, we would like to explore the similarity among the words not corresponding to concepts by using the generated word embeddings. In addition, we would like to estimate the similarity among the identified concepts by comparing vectors generated from the ontology SNOMED CT, such as those released by Pattisapu et al. (2020).

[6] `https://github.com/m-almagro-cadiz/hierarchical-evaluation.git`

A significant improvement is also expected when the ICD-11 is implemented, as it is designed as an ontology and provides more electronic resources. Such a version explicitly describes synonyms and narrower terms, improving the management of specificity.

Given the high performance for the AttentionXML algorithm, we would like to explore different variations. We have yet to change the TF-IDF to TF-BNS features to split the label space, implement the focal loss function, and adapt the algorithm to hierarchical learning based on pre-training on the top categories and fine tuning on the final codes. Alternatively, we would also like to further analyse relational-based transfer learning methods for zero-shot predictions and compare this performance with unsupervised methods.

Finally, the ensemble still needs to be improved. On the one hand, the integration of all the techniques in the ensemble should be explored, with a special focus on the parameter-based approach form Section 6.5.2. On the other hand, the way unsupervised and supervised methods are combined should be improved; maybe the probabilities of the codes generated by the semantic method could be used as features for the supervised methods, as proposed by Crammer et al. (2007), Pereira et al. (2013), and Zweigenbaum and Lavergne (2016). In addition, we would like to explore a voting method based on partial rather than exact matching to improve both traditional and hierarchy-based scores. To this end, we could sum the probability of all predictions for each code, weighing the values with the code similarity. Thus, the final probability for a target code would be the result of summing each inferred probability for each approach multiplied by two factors: the relevance corresponding to the frequency range for the inferred code and corresponding approach, and the similarity between the inferred code and the target code.

8.4 Publications

Journal publications

[1] **Almagro, M.**, Martinez, R., Fresno, V., and Montalvo, S. Preliminary study of the automatic annotation of hospital discharge report with ICD-10 codes. *Procesamiento del Lenguaje Natural*, 60, 45-52, 2018.

[2] **Almagro, M.**, Martínez, R., Montalvo, S., and Fresno, V. A cross-lingual approach to automatic ICD-10 coding of death certificates by exploring machine translation. *Journal of biomedical informatics*, 94, 103207, 2019.

[3] **Almagro, M.**, Unanue, R. M., Fresno, V., and Montalvo, S. ICD-10 coding of Spanish electronic discharge summaries: an extreme classification problem. *IEEE Access*, 8, 100073-100083, 2020.

Full papers in conference proceedings

[4] **Almagro, M.**, Montalvo, S., de Ilarraza, A. D., and Pérez, A. MAMTRA-MED at CLEF eHealth 2018: A Combination of Information Retrieval Techniques and Neural Networks for ICD-10 Coding of Death Certificates. In *Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, 2125, 2018.

[5] **Montalvo, S.**, Almagro, M., Martínez, R., Fresno, V., Lorenzo, S., Morales, M. C., Gonzalez, B., Alamo, J., and García-Caro, A. Graphical user interface for assistance with ICD-10 coding of hospital discharge records. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2786-2788, 2018.

[6] **Almagro, M.**, Martínez, R., Fresno, V., Montalvo, S., and Tissot, H. ICD-10 coding based on semantic distance: LSI UNED at CLEF eHealth 2020 Task 1. In *Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, 2696, 2020.

APPENDIX

A

DEATH CERTIFICATE TEMPLATE

Data supply is obligatory on basis of §. 30 of the 2016 law No CLV regarding statistics

CERTIFICATE ON THE EXAMINATION OF THE DECEASED

Register identificator							Running number							**To be filled in by the registrar**

1. The deceased's

a) Surename and given name ____________________ b) The mother's name: ____________________

c) Social Security Number: | | | | | | | | | |

2. Place of birth

3. Sex: male (1) – female (2) birth date

sex	year				month		day	

4. Address, postal code of permanent place of residence

5. Address, postal code of temporary place of residence

6. Name of the relative (arranger of the burial)

7. His/her address

8. His/her nearer indication

9. Place of death

10. Date of death year | month | day

11. Its nearer indication

12. Name of the medical attendant

13. His/her post and address of his/her place of work (consultation room, department)

14. Name of the physician examing the deceased

15. His/her post and address of his/her place of work (consultation room, department)

16. The physician □ does not find necessary a pathological examination □ finds necessary a path. examin. □ finds necessary an examination by a public authority

17. Motive of the examination (procedure) and other comments

18. Date of issue year | month | day

signature and stamp of the physician examining the deceased L. S.

19. Date of the transport to the house

20. Date authorized for the burial year | month | day

21. To be filled in by the prosector, expert

L. S. ____________________ signature

22. To be filled in by the prosector, expert in forensic medicine and by the health administrative organ, respectively, competent by the place of death who stated the cause of death

□ It can be cremated without autopsy

□ It can be cremated after autopsy □ It can be cremated after autopsy

L. S. ____________________ signature

Before filling in read the last paragraph of the Information!

23. The cause of death was stated by □ a pathologist (1) □ the medical attendant (2) □ an other physician (3)

24. a) Was an autopsy performed? □ Yes (1) □ No (2)
b) If yes, may further results be available later? □ Yes (1) □ No (2)

25. Sequence of diseases (or events) leading to death – *In ascending order!*

26. Approximate interval between onset of the disease (event) and death:

I.

Immediate cause of death a) ____________________ a)

as a consequence of:

Antecedent causes (if were any) b) ____________________ b)

as a consequence of:

c) ____________________ c)

as a consequence of:

Originating antecedent cause d) ____________________ d)

II. **Other significant conditions** contributing to death, but not directly related to the disease or condition causing it

27. Manner of death □ Natural (1) □ Accident (2) □ Suicide (3) □ Homicide (4) □ Could not be determined (5)

28. Place of injury □ Home (0) □ Residential institution (1) □ Institution (2) □ Sports area (3) □ Street, highway (4) □ Trade, service (5) □ Industrial area (6) □ Farm (7) Other: __________

29. Manner, cause and circumstances of the injury:

30. Date of injury: year | month | day

31. Pregnancy: □ Death occurred during pregnancy (1) □ Death occurred within 42 days after childbirth (2)
□ Death occurred between 42 days and 1 year after childbirth (3)

32.Date of issue: year | month | day

L. S. ____________________

signature and stamp of the physician

Comments

APPENDIX

B

DEATH CERTIFICATE EXAMPLE

Figure B.1: *Examples of Italian, French, and Hungarian death certificates. Tags shown in brackets are manually annotated ICD-10 codes, so they are not part of the text.*

Italian certificate	
leucemia linfoblastica acuta	[C91.0]
arresto cardiaco	[I46.9]
decadimento cognitivo, disfunzione, parkinson	[R41.8, R13, G20]

French certificate	
cachexie néoplastique	[C80.9]
fibrillation auriculaire avec réponse ventriculaire rapide	[I48.9, I47.2]
décompensation cardio-circulatoire et respiratoire	[I51.6, J98.8]
néoplasie pulmonaire	[C34.9]
résection sigmoïde pour néoplasie, BPCO, hypothyroïdie	[Y83.6, D48, J44.8, E03.9]

Hungarian certificate	
májcoma	[K72.9]
máj áttéti dag	[C78.7]

APPENDIX

C

CODIESP EXAMPLE

Presentamos el caso de una mujer de 29 años a la que se le realizó un **ecografía pélvica** tras una **ligadura laparoscópica**.
Se detectó una tumoración de 20 mm en la cara lateral derecha de la vejiga, bien delimitada e hipoecoica.
La paciente no presentaba síntomas miccionales, tal y como se relata en la entrevista posterior.

Se realizó una urografía intravenosa, en la que no se detectó ninguna alteración del tracto urinario superior.
El cistograma mostró un defecto de llenado superficial redondeado localizado en la pared de la vejiga derecha.
Los análisis de sangre y de orina estaban dentro de los límites normales.
Se realizó una **cistoscopia** a la paciente, que mostró la presencia de un tumor como mucosa ipsilateral ”conservada”, en el meato lateral derecho de la vejiga, inmediatamente por encima y por delante de la superficie ureteral.

Con el diagnóstico presuntivo de **leiomioma vesical**, se realizó una **resección** transuretral del tumor.
Los fragmentos resecados tenían un aspecto blanco, sólido y compacto, similar al de un adenoma prosternal, con escaso **sangrado**.
El material obtenido de la resección transuretral consistía en una proliferación de células fusiformes de citoplasma alargado, al igual que el núcleo, y ligeramente eosinófilas.
No se observaron mitosis ni atipias.
El estudio inmunohistoquímico mostró positividad para la actina específica del músculo (DAKO, clon HHF35) en las células proliferativas.

A los tres meses de la resección transuretral se realizó una **cistoscopia** de control, observándose una placa de área elevada sobre la zona de resección previa, compatible con una cistopatía calcoclear no incrustada y la posterior extirpación acidómica.

Type	**Code**	**Textual evidence**	**Character range**
Diagnosis	D30.3	*leiomioma vesical*	[873,889]
Procedure	0TJB8ZZ	*cistoscopia*	[631,641] & [1,478,1,488]
Procedure	0TTB	*resección vesical*	[883,889] ∪ [907,915]
Procedure	0UL7	*ligadura laparoscópica*	[99,120]
Diagnosis	R58	*sangrado*	[1,058,1,065]
Procedure	BW4GZZZ	*ecografía pélvica*	[72,88]

Figure C.1: *Example of a report from the CodiEsp corpus. The footer contains the annotations along with their evidence and positions in the text.*

APPENDIX

D

HUFA EXAMPLE

Figure D.1: *Example of the content of a HUFA Electronic Medical Record (part I).*

Anamnesis

ANTECEDENTES PERSONALES:
-HTA.
-Enf de Hansen, tratada con sulfonas en 2012 hasta 2016 en Dermatología de HUFA
ANTECEDENTES QUIRÚRGICOS: Hartmann (Sigmoidectomía + colostomía) por neoformación obstructiva en sigma el 17/11/2017.

TRATAMIENTO HABITUAL: atenolol 50mg 1-0-0, Higrotona 50mg 1-0-0.

ENFERMEDAD ACTUAL: Paciente dado de alta el 27/11 tras ingreso por intervención de neoplasia obstructiva de sigma, que acude por malestar abdominal, asociado a dos episodios de vómitos de contenido alimentario de varias horas de evolución. No fiebre termometrada. No cambio del hábito intestinal habitual, heces en bolsa de aspecto normal sin productos patológicos. No dolor torácico, no disnea. No síndrome miccional

Exploración Física
Afebril (Tª 36.3ºC).Eupneico. Buen estado general. Consciente y orientado en persona, tiempo y espacio. AC: rítmica sin soplos ni extratonos. AP: MVC,sin ruidos sobreañadidos. ABD: RRHH+, blando y depresible, no doloroso a la palpación profunda, no palpo masas ni visceromegalias. No signos de irritación peritoneal. PPRB negativa. Colostomía en FII, heces en bolsa de aspecto normal.
Exploraciones Complementarias

*Hemograma: LEU: 9.36 $10^3/\mu$L (3.50-11.00 Neut: 83.5 % (40.0-75.0); Hemogl: 11.5 g/dL (13.0-17.0); HTCO: 35.1 % (39.0-50.0); Plaquetas: 736 $10^3/\mu$L (130-450)

Determinaciones específicas de proteínas
PCR: 145.8 mg/L (¡ =5)
TAC ABDOMINAl: 03-12-2017

Cambios postquirúrgicos consistentes en colostomía de descarga y suturas en línea media de la pared abdominal.

En flanco izquierdo, ... se observan marcados cambios inflamatorios en la grasa mesentérica, con líquido libre con tendencia a la loculación de aproximadamente 8 x 6 x 8cm (AP x T x CC) y afectación de la fascia pararrenal anterior y del peritoneo de la pared abdominal que captan contraste. Embebidos en estos cambios inflamatorios se encuentran asas de yeyuno proximal. Hallazgos en probable relación con plastrón inflamatorio, a valorar dehiscencia de suturas como posible causa.

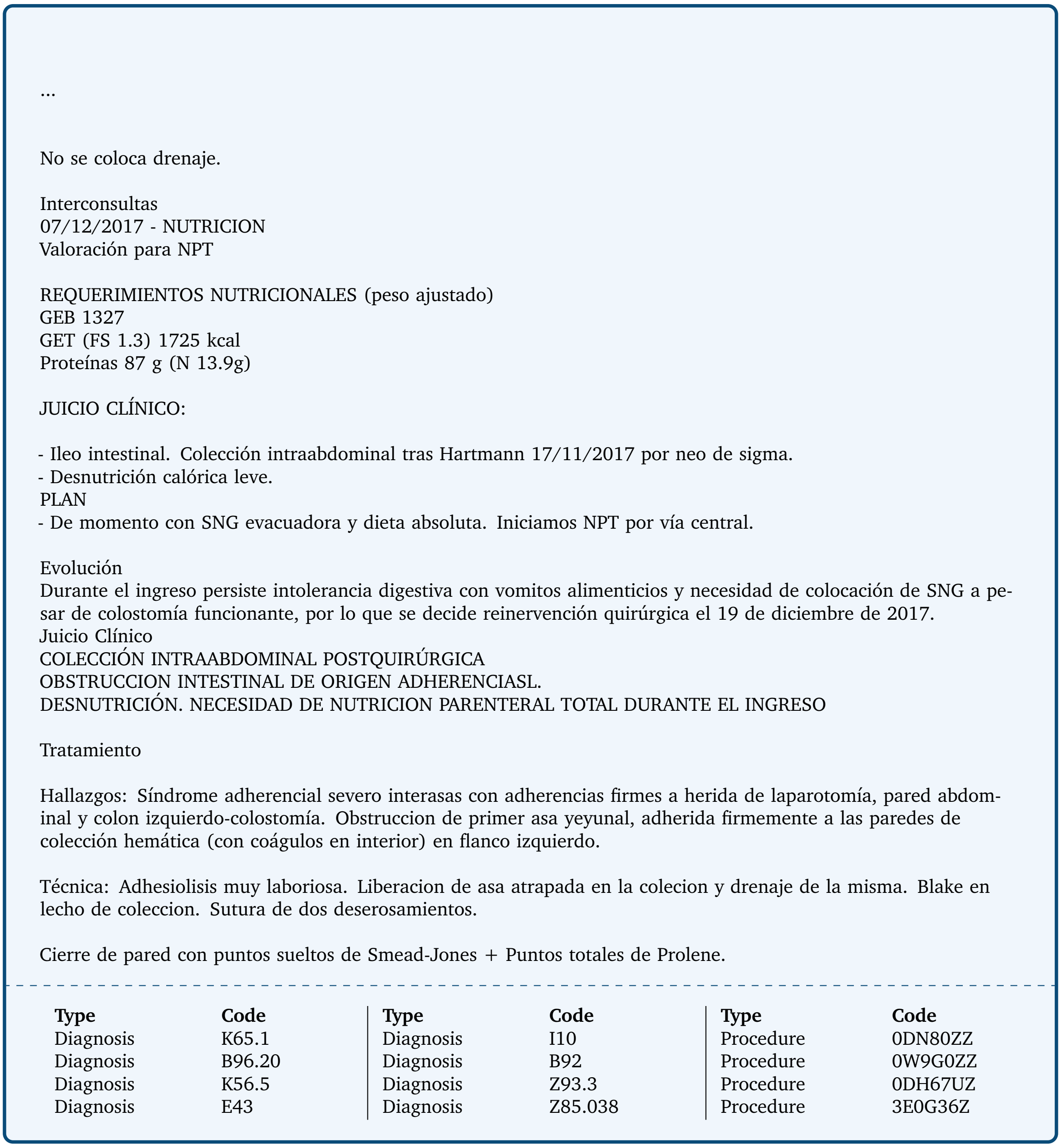
No se observan signos de obstrucción intestinal.
Resto de hallazgos abdominal (litiasis renal izquierda, pequeño quiste cortical simple en RI, hipertrofia prostática y ateromatosis calcificada aorto-iliaca) sin cambios.
En bases pulmonares se observa derrame pleural izquierdo y nódulos centrolobulillares en LID en relación con neumonitis.
Cambios degenerativos en el esqueleto axial. Anterolistesis grado I de L4 sobre L5 con espondilólisis ya conocida.

05/12/2017 - Drenaje por ecografía 04-12-2017
La colección en flanco izquierdo es ecogénica, heterogénea, sugiriendo hematoma en evolución.

Se pincha con aguja fina, saliendo escaso contenido hemático (mando muestra).

...

Figure D.2: *Example of the content of a HUFA Electronic Medical Record (part II).*



...

No se coloca drenaje.

Interconsultas
07/12/2017 - NUTRICION
Valoración para NPT

REQUERIMIENTOS NUTRICIONALES (peso ajustado)
GEB 1327
GET (FS 1.3) 1725 kcal
Proteínas 87 g (N 13.9g)

JUICIO CLÍNICO:

- Ileo intestinal. Colección intraabdominal tras Hartmann 17/11/2017 por neo de sigma.
- Desnutrición calórica leve.

PLAN
- De momento con SNG evacuadora y dieta absoluta. Iniciamos NPT por vía central.

Evolución
Durante el ingreso persiste intolerancia digestiva con vomitos alimenticios y necesidad de colocación de SNG a pesar de colostomía funcionante, por lo que se decide reinervención quirúrgica el 19 de diciembre de 2017.
Juicio Clínico
COLECCIÓN INTRAABDOMINAL POSTQUIRÚRGICA
OBSTRUCCION INTESTINAL DE ORIGEN ADHERENCIASL.
DESNUTRICIÓN. NECESIDAD DE NUTRICION PARENTERAL TOTAL DURANTE EL INGRESO

Tratamiento

Hallazgos: Síndrome adherencial severo interasas con adherencias firmes a herida de laparotomía, pared abdominal y colon izquierdo-colostomía. Obstruccion de primer asa yeyunal, adherida firmemente a las paredes de colección hemática (con coágulos en interior) en flanco izquierdo.

Técnica: Adhesiolisis muy laboriosa. Liberacion de asa atrapada en la colecion y drenaje de la misma. Blake en lecho de coleccion. Sutura de dos deserosamientos.

Cierre de pared con puntos sueltos de Smead-Jones + Puntos totales de Prolene.

Type	**Code**	**Type**	**Code**	**Type**	**Code**
Diagnosis	K65.1	Diagnosis	I10	Procedure	0DN80ZZ
Diagnosis	B96.20	Diagnosis	B92	Procedure	0W9G0ZZ
Diagnosis	K56.5	Diagnosis	Z93.3	Procedure	0DH67UZ
Diagnosis	E43	Diagnosis	Z85.038	Procedure	3E0G36Z

APPENDIX

E

HUFA TOKENISATION EXAMPLE

Figure E.1: *Untranslated example of the tokenization of two paragraphs of the report shown in Figure D.1. The original text is shown at the top, while the tokens separated by spaces are shown below.*

...

Exploración física
Afebril (Tª 36.3ºC).Eupneico. Buen estado general. Consciente y orientado en persona, tiempo y espacio. AC: rítmica sin soplos ni extratonos. AP: MVC, sin ruidos adicionales. ABD: HR+, suave y depresible, no doloroso a la palpación profunda, sin masas ni visceromegalias. No hay signos de irritación peritoneal. BRFP negativo. Colostomía en fase II, heces en bolsa de aspecto normal.
Exploraciones complementarias

*Hemograma: LEU: 9.36 $10^3/\mu$L (3.50-11.00 Neut: 83.5 % (40.0-75.0); Hemogl: 11.5 g/dL (13.0-17.0); Hematocrit: 35.1 % (39.0-50.0); Plaquetas: 736 $10^3/\mu$L (130-450)

...

Exploración física
Afebril (Tª 36.3 ºC) .
Eupneico .
Buen estado general .
Consciente y **orientado** en **persona** , **tiempo** y **espacio** .
AC : **rítmica** sin **soplos** ni **extratonos** .
AP : **MVC** , **sin ruidos adicionales** .
ABD : **HR+** , **suave** y **depresible** , **no doloroso** a la **palpación profunda** , **sin masas** ni **visceromegalias** .
No hay **signos** de **irritación peritoneal** .
BRFP negativo .
Colostomía en **fase II** , **heces** en **bolsa** de **aspecto normal** .
Exploraciones complementarias

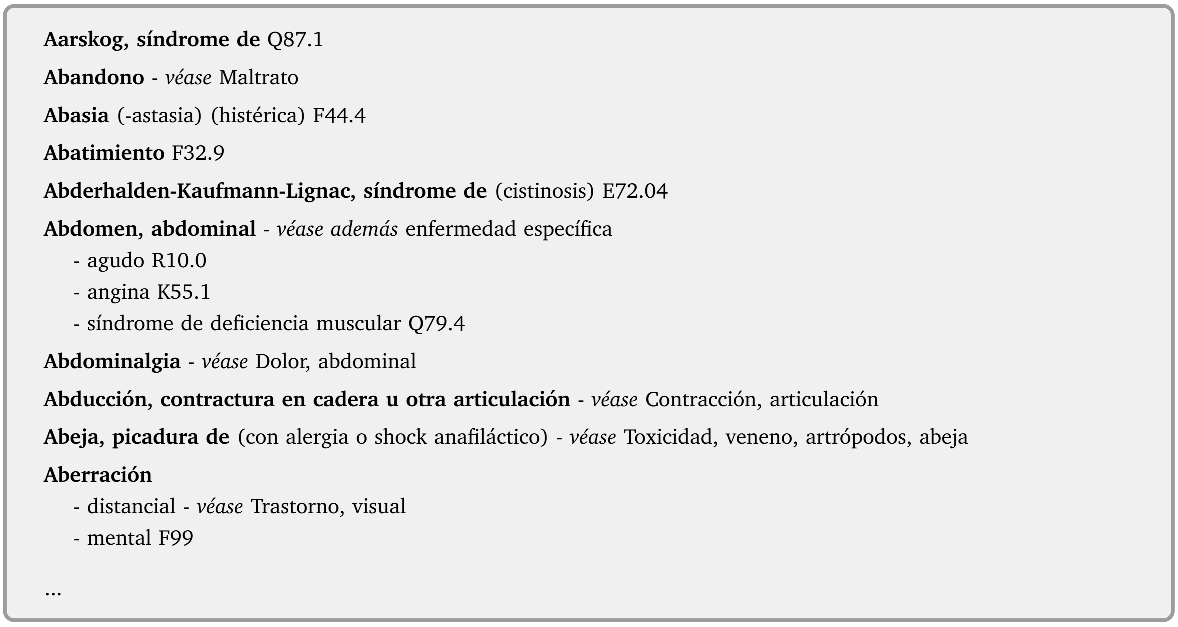
* **Hemograma** : **LEU** : 9.36 10^3 / μL (3.50 - 11.00 **Neut** : 83.5 % (40.0 - 75.0) ;
Hemogl : 11.5 g/dL (13.0 - 17.0) ;
Hematocrito : 35.1 % (39.0 - 50.0) ;
Plaquetas : 736 10^3 / μL (130 - 450)

APPENDIX

F

ALPHABETICAL INDEX EXAMPLE

Figure F.1: *Spanish example of the content of the Alphabetical Index.*



Aarskog, síndrome de Q87.1

Abandono - *véase* Maltrato

Abasia (-astasia) (histérica) F44.4

Abatimiento F32.9

Abderhalden-Kaufmann-Lignac, síndrome de (cistinosis) E72.04

Abdomen, abdominal - *véase además* enfermedad específica

- agudo R10.0
- angina K55.1
- síndrome de deficiencia muscular Q79.4

Abdominalgia - *véase* Dolor, abdominal

Abducción, contractura en cadera u otra articulación - *véase* Contracción, articulación

Abeja, picadura de (con alergia o shock anafiláctico) - *véase* Toxicidad, veneno, artrópodos, abeja

Aberración

- distancial - *véase* Trastorno, visual
- mental F99

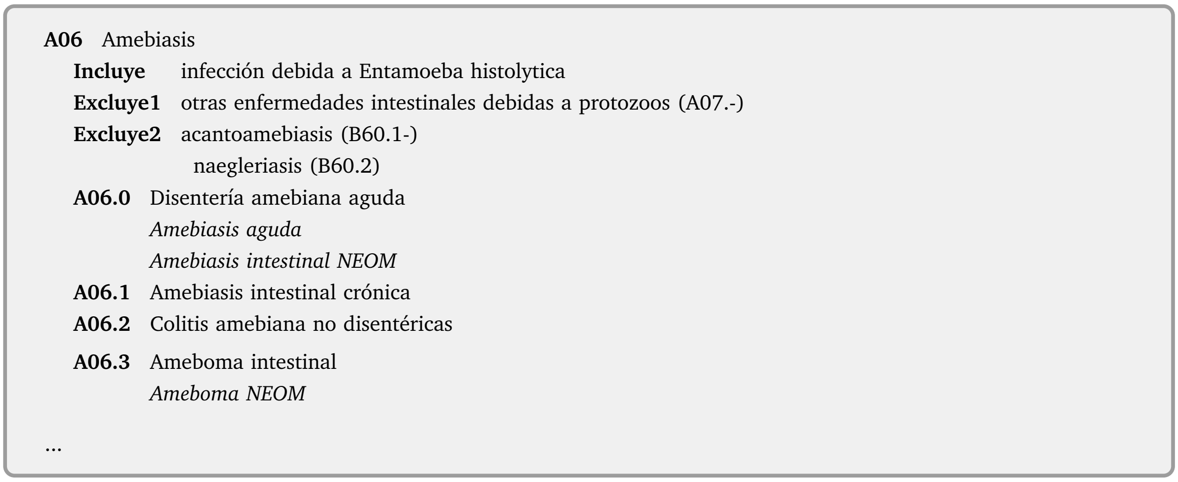
...

APPENDIX

G

TABULAR LIST EXAMPLE

Figure G.1: *Example of the content of Tabular List.*



A06 Amebiasis

Incluye infección debida a Entamoeba histolytica
Excluye1 otras enfermedades intestinales debidas a protozoos (A07.-)
Excluye2 acantoamebiasis (B60.1-)
naegleriasis (B60.2)

A06.0 Disentería amebiana aguda
Amebiasis aguda
Amebiasis intestinal NEOM

A06.1 Amebiasis intestinal crónica

A06.2 Colitis amebiana no disentéricas

A06.3 Ameboma intestinal
Ameboma NEOM

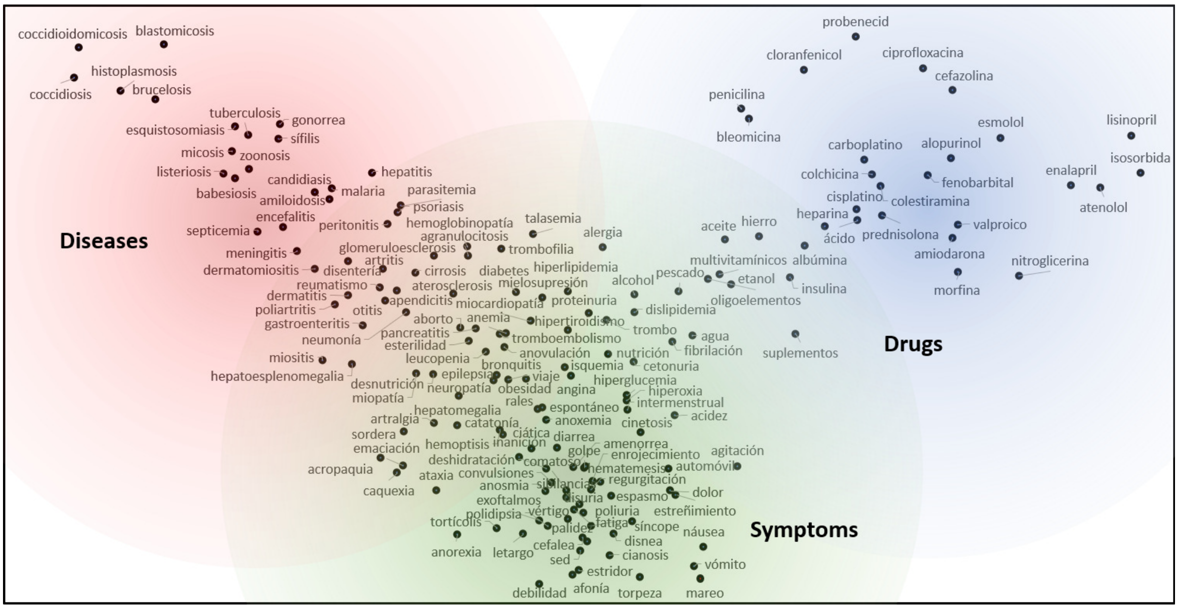
...

APPENDIX

H

CLINICAL WORD EMBEDDINGS PROJECTION

Figure H.1: *Two-dimensional representation of clinical words in SCE semantic space.*



APPENDIX

I

HUFA EXAMPLE PRODUCED BY THE AWD-LSTM MODEL

Figure I.1: *Example of automatic generation of clinical text based on the model CSCE. Parts of Example 3.12 and 3.13 in Section 3.2 (in grey) have been used to produce new parts (in bold).*

Anamnesis

ANTECEDENTES PERSONALES:
-HTA.
-Enf de Hansen**. DM tipo 2. No alergias medicamentosas conocidas.**
-Bebedor habitual.
-Fumador de # cig/día durante # años.

ENFERMEDAD ACTUAL: Paciente dado de alta el 27/11 tras ingreso por intervención de neoplasia obstructiva de sigma, que acude por **cuadro de episodio de acidosis metabólica realizado por la bradicardia para continuar tratamiento con Sintrom. Varón de # años, acude a urgencias por fiebre y pérdida de consciencia. No dolor torácico ni distérmica ni escalofríos ni febrícula, no otra sintomatología acompañante.**

Exploración Física
Afebril (Tª 36.3ºC).Eupneico. Buen estado general. Consciente y orientado en persona, tiempo y espacio. **Normocoloreado, bien hidratado, nutrido y perfundido. Cyc: Carótidas rítmicas y simétricas. AP: mvc. No edemas ni signos de TVP. Pulsos distales presentes y simétricos. Hemograma: Leucocitos #^{3}/l (#-#), Neutrófilos %#% (#-#), Linfocitos %#% (#-#), Monocitos %#% (#-#), Eosinófilos %#% (#-#), Basófilos %#% (#-#).**

Cambios postquirúrgicos consistentes en colostomía de descarga y suturas en línea media de la pared abdominal. En flanco izquierdo, ... **se observan signos de metástasis con órbita izquierda. Adenopatías mediastínicas y axilares bilaterales y compatibles con infiltrado inflamatorio en los nódulos pulmonares (TCE). En el territorio de EEG, no se observan datos de signos de isquemia aguda ni hallazgos patológicos.**

JUICIO CLÍNICO:
- Ileo intestinal. Colección intraabdominal tras Hartmann 17/11/2017 por neo de sigma.
- Vesícula biliar distendida, con dilatación de la vía biliar intrahepática, Colecistectomía.
- HTA con DM.
- Isquemia aguda en ergometría isotópica.

Evolución
Durante el ingreso persiste intolerancia digestiva con vomitos alimenticios y **deposiciones normales. Presenta signos de ICC. HTA en tto con ADO. Episodio de inestabilidad hemodinámica FA auricular paroxística rápida.**

APPENDIX

J

STS EVALUATION ON THE REDUCED DATA SETS

	r **based on Average Cosine Similarity**				r **based on Word Mover's Distance**			
	UMNSRS-sim	UMNSRS-rel	MayoSRS	Avg.	UMNSRS-sim	UMNSRS-rel	MayoSRS	Avg.
CN	0.56	0.47	0.25	0.48	0.14	0.11	0.03	0.10
SBWC	0.43	0.39	0.12	0.38	0.44	0.37	0.23	0.39
SUC	0.47	0.42	0.14	0.41	0.47	0.37	0.28	0.41
SHE	0.58	0.52	0.32	0.52	0.54	0.46	0.37	0.49
SCE	**0.66**	0.58	0.56	0.61	0.55	**0.49**	0.45	0.52
SCE-L	0.63	0.55	0.48	0.58	0.54	0.44	0.45	0.49
SCE-SBWC	**0.66**	0.57	0.55	0.61	0.56	0.48	0.44	0.51
SCE-SUC	0.65	0.57	0.55	0.61	0.56	0.48	0.44	0.51
SCE-CN	0.65	**0.58**	0.56	0.61	0.55	0.48	0.45	0.51
Retrofitted SCE	**0.66**	**0.58**	**0.59**	**0.62**	0.27	0.26	**0.49**	0.30
Retrofitted SCE-L	0.63	0.56	0.50	0.59	**0.59**	**0.49**	0.43	**0.53**

Table J.1: *Intrinsic evaluation of word embeddings models through Pearson correlation coefficients in an STS task.*

BIBLIOGRAPHY

Adi, Yossi et al. (2016). “Fine-grained analysis of sentence embeddings using auxiliary prediction tasks”. In: *arXiv preprint arXiv:1608.04207*.

Agirre, Aitor Gonzalez et al. (2019). “Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track”. In: *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pp. 1–10.

Agrawal, Rahul et al. (2013). “Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages”. In: *Proceedings of the 22nd international conference on World Wide Web*, pp. 13–24.

Ahalt, Stanley C et al. (2019). “Clinical data: sources and types, regulatory constraints, applications”. In: *Clinical and translational science* 12.4, p. 329.

Akbik, Alan, Duncan Blythe, and Roland Vollgraf (2018). “Contextual string embeddings for sequence labeling”. In: *Proceedings of the 27th international conference on computational linguistics*, pp. 1638–1649.

Akbik, Alan et al. (2019). “FLAIR: An easy-to-use framework for state-of-the-art NLP”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 54–59.

Akhtyamova, Liliya et al. (2020). “Testing Contextualized Word Embeddings to Improve NER in Spanish Clinical Case Narratives”. In: *IEEE Access* 8, pp. 164717–164726.

Al-Rfou, Rami, Bryan Perozzi, and Steven Skiena (2013). “Polyglot: Distributed word representations for multilingual nlp”. In: *arXiv preprint arXiv:1307.1662*.

Alawad, Mohammed et al. (2018). “Retrofitting word embeddings with the UMLS metathesaurus for clinical information extraction”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 2838–2846.

Albitar, Shereen, Sébastien Fournier, and Bernard Espinasse (2014). "An effective TF/IDF-based text-to-text semantic similarity measure for text classification". In: *International Conference on Web Information Systems Engineering*. Springer, pp. 105–114.

Aleksovski, Zharko and Merlijn Sevenster (2010). "Identifying breast cancer concepts in snomed-ct using large text corpus". In: *International Conference on Electronic Healthcare*. Springer, pp. 27–34.

Almagro, Mario et al. (2019). "A cross-lingual approach to automatic ICD-10 coding of death certificates by exploring machine translation". In: *Journal of biomedical informatics* 94, p. 103207.

Almagro, Mario et al. (2020). "ICD-10 coding of Spanish electronic discharge summaries: an extreme classification problem". In: *IEEE Access* 8, pp. 100073–100083.

Alonso, Laura et al. (2007). "The sensem project: Syntactico-semantic annotation of sentences in spanish". In: *Amsterdam studies in the theory and history of linguistic science series 4* 292, p. 89.

Amin, Saadullah et al. (2019). "MLT-DFKI at CLEF eHealth 2019: Multi-label Classification of ICD-10 Codes with BERT." In: *CLEF (Working Notes)*.

Anaby-Tavor, Ateret et al. (2019). "Not Enough Data? Deep Learning to the Rescue!" In: *arXiv preprint arXiv:1911.03118*.

Aramaki, E. et al. (2014). "Overview of the NTCIR-11 MedNLP-2 Task". In: *NTCIR*.

Aramaki, Eiji et al. (2016). "Overview of the NTCIR-12 MedNLPDoc Task." In: *NTCIR*.

Arifoğlu, Damla et al. (2014). "CodeMagic: semi-automatic assignment of ICD-10-AM codes to patient records". In: *Information Sciences and Systems 2014*. Springer, pp. 259–268.

Aronson, Alan R (2001). "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program." In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association, p. 17.

Aronson, Alan R et al. (2007). "From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches". In: *Biological, translational, and clinical language processing*, pp. 105–112.

Arora, Sanjeev, Yingyu Liang, and Tengyu Ma (2017). "A simple but tough-to-beat baseline for sentence embeddings". In: *5th International Conference on Learning Representations, ICLR 2017*.

Aroyehun, Segun Taofeek and Alexander Gelbukh (2018). "Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling". In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 90–97.

Asgarian, Azin et al. (2018). "A hybrid instance-based transfer learning method". In: *arXiv preprint arXiv:1812.01063*.

Athiwaratkun, Ben, Andrew Gordon Wilson, and Anima Anandkumar (2018). “Probabilistic fasttext for multi-sense word embeddings”. In: *arXiv preprint arXiv:1806.02901*.

Atutxa, Aitziber et al. (2018). “IxaMed at CLEF eHealth 2018 Task 1: ICD10 Coding with a Sequence-to-Sequence Approach.” In: *CLEF (Working Notes)*.

Atutxa, Aitziber et al. (2019). “Interpretable deep learning to map diagnostic texts to ICD-10 codes”. In: *International journal of medical informatics* 129, pp. 49–59.

Babbar, Rohit and Bernhard Schölkopf (2017). “Dismec: Distributed sparse machines for extreme multi-label classification”. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 721–729.

— (2019). “Data scarcity, robustness and extreme multi-label classification”. In: *Machine Learning* 108.8, pp. 1329–1351.

Baevski, Alexei et al. (2019). “Cloze-driven pretraining of self-attention networks”. In: *arXiv preprint arXiv:1903.07785*.

Baker, Collin F, Charles J Fillmore, and John B Lowe (1998). “The berkeley framenet project”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pp. 86–90.

Balasubramanian, Krishnakumar and Guy Lebanon (2012). “The landmark selection method for multiple output prediction”. In: *arXiv preprint arXiv:1206.6479*.

Banerjee, Ritwik et al. (2015). “Patient centered identification, attribution, and ranking of adverse drug events”. In: *2015 International Conference on Healthcare Informatics*. IEEE, pp. 18–27.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (2014). “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 238–247.

Barrows Jr, Randolph C, M Busuioc, and Carol Friedman (2000). “Limited parsing of notational text visit notes: ad-hoc vs. NLP approaches.” In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association, p. 51.

Baumel, Tal et al. (2018). “Multi-label classification of patient notes: case study on ICD code assignment”. In: *Workshops at the thirty-second AAAI conference on artificial intelligence*.

Beltagy, Iz, Kyle Lo, and Arman Cohan (2019). “SciBERT: A pretrained language model for scientific text”. In: *arXiv preprint arXiv:1903.10676*.

Beneš, Karel and Lukáš Burget (2020). “Text Augmentation for Language Models in High Error Recognition Scenario”. In: *arXiv preprint arXiv:2011.06056*.

Bengio, Yoshua et al. (2003). “A neural probabilistic language model”. In: *The journal of machine learning research* 3, pp. 1137–1155.

Bhatia, Kush et al. (2015). “Sparse Local Embeddings for Extreme Multi-label Classification.” In: *NIPS*. Vol. 29, pp. 730–738.

Bi, Wei and James Kwok (2013). “Efficient multi-label classification with many labels”. In: *International Conference on Machine Learning*. PMLR, pp. 405–413.

Bian, Jiang, Bin Gao, and Tie-Yan Liu (2014). “Knowledge-powered deep learning for word embedding”. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, pp. 132–148.

Bingel, Joachim and Anders Søgaard (2017). “Identifying beneficial task relations for multi-task learning in deep neural networks”. In: *arXiv preprint arXiv:1702.08303*.

Biron, P et al. (2014). “An information retrieval system for computerized patient records in the context of a daily hospital practice: the example of the Léon Bérard Cancer Center (France)”. In: *Applied clinical informatics* 5.1, p. 191.

Biseda, Brent et al. (2020). “Prediction of ICD Codes with Clinical BERT Embeddings and Text Augmentation with Label Balancing using MIMIC-III”. In: *arXiv preprint arXiv:2008.10492*.

Blanco, Alberto, Alicia Pérez, and Arantza Casillas (2020). “Extreme Multi-Label ICD Classification: Sensitivity to Hospital Service and Time”. In: *IEEE Access* 8, pp. 183534–183545.

Blanco, Alberto et al. (2020). “Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity”. In: *Computer methods and programs in biomedicine* 188, p. 105264.

Bodenreider, Olivier (2004). “The Unified Medical Language System (UMLS): integrating biomedical terminology”. In: *PubMed* 32.1, pp. D267–70. DOI: 10.1093/nar/gkh061.

Bojanowski, Piotr et al. (2016). “Enriching Word Vectors with Subword Information”. In: *arXiv preprint arXiv:1607.04606*.

Bowman, Samuel R, Christopher Potts, and Christopher D Manning (2014). “Recursive neural networks for learning logical semantics”. In: *CoRR, abs/1406.1827* 5.

Boytcheva, Svetla (2011). “Automatic matching of ICD-10 codes to diagnoses in discharge letters”. In: *Proceedings of the Second Workshop on Biomedical Natural Language Processing*, pp. 11–18.

Brown, Tom B et al. (2020). “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165*.

Campbell, Sharon and Katrina Giadresco (2020). “Computer-assisted clinical coding: A narrative review of the literature on its benefits, limitations, implementation and impact on clinical coding professionals”. In: *Health Information Management Journal* 49.1, pp. 5–18.

Cardellino, Cristian (2019). *Spanish Billion Words Corpus and Embeddings*. URL: https://crscardellino.github.io/SBWCE/.

Carroll, Robert J et al. (2012). “Portability of an algorithm to identify rheumatoid arthritis in electronic health records”. In: *Journal of the American Medical Informatics Association* 19.e1, e162–e169.

Cañete, José (2019). *Spanish Unannotated Corpora*. URL: `https://github.com/josecannete/spanish-corpora`.

Cer, Daniel et al. (2018). "Universal sentence encoder". In: *arXiv preprint arXiv:1803.11175*.

Chalkidis, Ilias et al. (2019a). "Extreme multi-label legal text classification: A case study in EU legislation". In: *arXiv preprint arXiv:1905.10892*.

Chalkidis, Ilias et al. (2019b). "Large-scale multi-label text classification on EU legislation". In: *arXiv preprint arXiv:1906.02192*.

Chalkidis, Ilias et al. (2020). "An empirical study on large-scale multi-label text classification including few and zero-shot labels". In: *arXiv preprint arXiv:2010.01653*.

Chang, Kai-Wei, Wen-tau Yih, and Christopher Meek (2013). "Multi-relational latent semantic analysis". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1602–1612.

Chang, Wei-Cheng et al. (2019). "X-bert: extreme multi-label text classification with using bidirectional encoder representations from transformers". In: *arXiv preprint arXiv:1905.02331*.

Chang, Wei-Cheng et al. (2020). "Taming pretrained transformers for extreme multi-label text classification". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3163–3171.

Changpinyo, Soravit, Hexiang Hu, and Fei Sha (2018). "Multi-task learning for sequence tagging: An empirical study". In: *arXiv preprint arXiv:1808.04151*.

Chawla, Nitesh V, Nathalie Japkowicz, and Aleksander Kotcz (2004). "Special issue on learning from imbalanced data sets". In: *ACM SIGKDD explorations newsletter* 6.1, pp. 1–6.

Chen, Ping, Araly Barrera, and Chris Rhodes (2010). "Semantic analysis of free text and its application on automatically assigning ICD-9-CM codes to patient records". In: *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*. IEEE, pp. 68–74.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chen, Yao-Nan and Hsuan-Tien Lin (2012). "Feature-aware label space dimension reduction for multi-label classification". In: *Advances in neural information processing systems* 25, pp. 1529–1537.

Chen, YunZhi, HuiJuan Lu, and LanJuan Li (2017). "Automatic ICD-10 coding algorithm using an improved longest common subsequence based on semantic similarity". In: *PloS one* 12.3, e0173410.

Chen, Zhao et al. (2018). "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks". In: *International Conference on Machine Learning*. PMLR, pp. 794–803.

Cheng, Heng-Tze et al. (2016). “Wide & deep learning for recommender systems”. In: *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10.

Chiaravalloti, Maria Teresa et al. (2014). “A Coding Support System for the ICD-9-CM standard”. In: *2014 IEEE International Conference on Healthcare Informatics*. IEEE, pp. 71–78.

Chomsky, Noam (2014). *Aspects of the Theory of Syntax*. Vol. 11. MIT press.

Cimino, James J (1996). “Coding systems in health care”. In: *Methods of information in medicine* 35.04/05, pp. 273–284.

Ciolino, Matthew, David Noever, and Josh Kalin (2021). “Multilingual Augmenter: The Model Chooses”. In: *arXiv preprint arXiv:2102.09708*.

Cissé, Moustapha et al. (2013). “Robust bloom filters for large multilabel classification tasks”. In: *Advances in Neural Information Processing Systems 26*, pp. 1851–1859.

Clark, Kevin et al. (2020). “Electra: Pre-training text encoders as discriminators rather than generators”. In: *arXiv preprint arXiv:2003.10555*.

Coldewey, Devin and Frederic Lardinois (2017). “DeepL schools other online translators with clever machine learning”. In: *Techcrunch. com*.

Collobert, Ronan et al. (2011). “Natural language processing (almost) from scratch”. In: *Journal of machine learning research* 12.ARTICLE, pp. 2493–2537.

Conneau, Alexis et al. (2017). “Supervised learning of universal sentence representations from natural language inference data”. In: *arXiv preprint arXiv:1705.02364*.

Coulombe, Claude (2018). “Text data augmentation made simple by leveraging nlp cloud apis”. In: *arXiv preprint arXiv:1812.04718*.

Council, National Research, Committee on Population, et al. (2011). *International differences in mortality at older ages: Dimensions and sources*. National Academies Press.

Crammer, Koby et al. (2007). “Automatic code assignment to medical text”. In: *Biological, translational, and clinical language processing*, pp. 129–136.

Dahiya, Kunal et al. (2019). “DeepXML: Scalable & Accurate Deep Extreme Classification for Matching User Queries to Advertiser Bid Phrases”. In:

Dahiya, Kunal et al. (2021). “DeepXML: A Deep Extreme Multi-Label Learning Framework Applied to Short Text Documents”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 31–39.

Das, Debasmita et al. (July 2020). “Information Retrieval and Extraction on COVID-19 Clinical Articles Using Graph Community Detection and Bio-BERT Embeddings”. In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/2020.nlpcovid19-acl.7`.

Dermouche, Mohamed et al. (2016). “Supervised topic models for diagnosis code assignment to discharge summaries”. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 485–497.

Devlin, Jacob et al. (2018). “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805*.

Dhrangadhariya, Anjani et al. (2021). “Classification of Noisy Free-Text Prostate Cancer Pathology Reports Using Natural Language Processing”. In: *Pattern Recognition. ICPR International Workshops and Challenges*.

Di Renzo, Andrea (2020). “Documentos médico-legales: estudio comparativo y propuesta de traducción de informes médicos”. PhD thesis. Università degli Studi di Padova.

Dong, Hang et al. (2021). “Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation”. In: *Journal of biomedical informatics* 116, p. 103728.

Donnelly, Kevin (2006). “SNOMED-CT: The advanced terminology and coding system for eHealth”. In: *Studies in health technology and informatics* 121, p. 279.

Doval, Yerai et al. (2018). “Improving cross-lingual word embeddings by meeting in the middle”. In: *arXiv preprint arXiv:1808.08780*.

Drazen, Jeffrey M and Gregory D Curfman (2004). *Public access to biomedical research*.

Edinger, Tracy et al. (2012). “Barriers to retrieving patient information from electronic health record data: failure analysis from the TREC medical records track”. In: *AMIA annual symposium proceedings*. Vol. 2012. American Medical Informatics Association, p. 180.

Erhan, Dumitru et al. (2010). “Why does unsupervised pre-training help deep learning?” In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 201–208.

Erraguntla, Madhav et al. (2012). “Inference of missing ICD 9 codes using text mining and nearest neighbor techniques”. In: *2012 45th hawaii international conference on system sciences*. IEEE, pp. 1060–1069.

Europe, World Health Organization. Regional Office for (1999). *Healthy living : what is a healthy lifestyle?*

Fadaee, Marzieh, Arianna Bisazza, and Christof Monz (2017). “Data augmentation for low-resource neural machine translation”. In: *arXiv preprint arXiv:1705.00440*.

Fader, Anthony, Luke Zettlemoyer, and Oren Etzioni (2013). “Paraphrase-driven learning for open question answering”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1608–1618.

Faruqui, Manaal et al. (2014). “Retrofitting word vectors to semantic lexicons”. In: *arXiv preprint arXiv:1411.4166*.

Felbo, Bjarke et al. (2017). “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm”. In: *arXiv preprint arXiv:1708.00524*.

Ferng, Chung-Sung and Hsuan-Tien Lin (2011). “Multi-label classification with error-correcting codes”. In: *Asian conference on machine learning*. PMLR, pp. 281–295.

Ferraro, Jeffrey P et al. (2013). “Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation”. In: *Journal of the American Medical Informatics Association* 20.5, pp. 931–939.

Ford, Elizabeth et al. (2013). “Optimising the use of electronic health records to estimate the incidence of rheumatoid arthritis in primary care: what information is hidden in free text?” In: *BMC medical research methodology* 13.1, pp. 1–12.

Forman, George (2008). “BNS feature scaling: an improved representation over tf-idf for svm text classification”. In: *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 263–270.

Foulds, J. and Eibe Frank (2010). “A review of multi-instance learning assumptions”. In: *The Knowledge Engineering Review* 25, pp. 1–25.

Freund, Yoav and Robert E Schapire (1997). “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1, pp. 119–139.

Fried, Daniel and Kevin Duh (2014). “Incorporating both distributional and relational semantics in word representations”. In: *arXiv preprint arXiv:1412.4369*.

Friedman, Carol et al. (2004). “Automated encoding of clinical documents based on natural language processing”. In: *Journal of the American Medical Informatics Association* 11.5, pp. 392–402.

Fung, Kin Wah, Julia Xu, and Olivier Bodenreider (2020). “The new International Classification of Diseases 11th edition: a comparative analysis with ICD-10 and ICD-10-CM”. In: *Journal of the American Medical Informatics Association* 27.5, pp. 738–746.

Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch (2013). “PPDB: The paraphrase database”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 758–764.

Garcıa-Santa, Nuria and Kendrick Cetina (2020). “FLE at CLEF eHealth 2020: Text Mining and Semantic Knowledge for Automated Clinical Encoding”. In: *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*.

Garg, Siddhant and Goutham Ramakrishnan (2020). “Bae: Bert-based adversarial examples for text classification”. In: *arXiv preprint arXiv:2004.01970*.

Goldstein, Ira, Anna Arzumtsyan, and Özlem Uzuner (2007). “Three approaches to automatic assignment of ICD-9-CM codes to radiology reports”. In: *AMIA Annual Symposium Proceedings*. Vol. 2007. American Medical Informatics Association, p. 279.

Gordon, Mitchell A, Kevin Duh, and Nicholas Andrews (2020). “Compressing bert: Studying the effects of weight pruning on transfer learning”. In: *arXiv preprint arXiv:2002.08307*.

Greenwood, RM (1972). “Kodiac, a system for disease coding by a medium-sized computer”. In: *International journal of bio-medical computing* 3.2, pp. 123–134.

Gregg, William et al. (2003). “StarTracker: an integrated, web-based clinical search engine”. In: *AMIA annual symposium proceedings*. Vol. 2003. American Medical Informatics Association, p. 855.

Grimes, Seth (2008). “Unstructured data and the 80 percent rule”. In: *Carabridge Bridgepoints* 10.

Grover, Aditya and Jure Leskovec (2016). “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864.

Gundersen, Michael L et al. (1996). “Development and evaluation of a computerized admission diagnoses encoding system”. In: *Computers and Biomedical Research* 29.5, pp. 351–372.

Guo, Hongyu, Yongyi Mao, and Richong Zhang (2019). “Augmenting data with mixup for sentence classification: An empirical study”. In: *arXiv preprint arXiv:1905.08941*.

Hanauer, David A et al. (2015). “Supporting information retrieval from electronic health records: A report of University of Michigan’s nine-year experience in developing and using the Electronic Medical Record Search Engine (EMERSE)”. In: *Journal of biomedical informatics* 55, pp. 290–300.

Harris, Zellig S (1954). “Distributional structure”. In: *Word* 10.2-3, pp. 146–162.

Harteloh, Peter (2020). “The implementation of an automated coding system for cause-of-death statistics”. In: *Informatics for Health and Social Care* 45.1, pp. 1–14.

Häyrinen, Kristiina, Kaija Saranto, and Pirkko Nykänen (2008). “Definition, structure, content, use and impacts of electronic health records: a review of the research literature”. In: *International journal of medical informatics* 77.5, pp. 291–304.

He, Haibo and Yunqian Ma (2013). “Imbalanced learning: foundations, algorithms, and applications”. In:

Heidari, Alireza et al. (2019). “Holodetect: Few-shot learning for error detection”. In: *Proceedings of the 2019 International Conference on Management of Data*, pp. 829–846.

Henriksson, Aron, Martin Hassel, and Maria Kvist (2011). “Diagnosis code assignment support using random indexing of patient records–a qualitative feasibility study”. In: *Conference on Artificial Intelligence in Medicine in Europe*. Springer, pp. 348–352.

Hill, Felix, Roi Reichart, and Anna Korhonen (2015). “Simlex-999: Evaluating semantic models with (genuine) similarity estimation”. In: *Computational Linguistics* 41.4, pp. 665–695.

Hill, Felix et al. (2017). “The representational geometry of word meanings acquired by neural machine translation models”. In: *Machine Translation* 31.1, pp. 3–18.

Ho-Dac, Lydia Mai et al. (2017). “LITL at CLEF eHealth2017: automatic classication of death reports”. In: *CLEF (Working Notes)*.

Hou, Yutai et al. (2018). “Sequence-to-sequence data augmentation for dialogue language understanding”. In: *arXiv preprint arXiv:1807.01554*.

Houlsby, Neil et al. (2019). “Parameter-efficient transfer learning for NLP”. In: *International Conference on Machine Learning*. PMLR, pp. 2790–2799.

Howard, Jeremy and Sebastian Ruder (2018). “Universal language model fine-tuning for text classification”. In: *arXiv preprint arXiv:1801.06146*.

Howell, RW and Ruth M Loy (1968). “Disease coding by computer. The” fruit machine” method.” In: *British journal of preventive & social medicine* 22.3, p. 178.

HSCIC, Health & Social Care Information Centre (2014). *OPCS Classification of Interventions and Procedures Version 4.7 combined Volumes I & II / Health and Social Care Information Centre*. English. 4th revision. TSO (The Stationery Office). ISBN: 9780113229901 9780113229918 9780113229925.

Huang, Kexin, Jaan Altosaar, and Rajesh Ranganath (2019). “Clinicalbert: Modeling clinical notes and predicting hospital readmission”. In: *arXiv preprint arXiv:1904.05342*.

Huesch, Marco D and Timothy J Mosher (2017). “Using it or losing it? The case for data scientists inside health care”. In: *Nejm Catalyst* 3.3.

Ive, Julia et al. (2018). “KCL-Health-NLP@ CLEF eHealth 2018 Task 1: ICD-10 Coding of French and Italian Death Certificates with Character-Level Convolutional Neural Networks”. In: *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*. Vol. 2125. CEUR-WS.

Jain, Himanshu, Yashoteja Prabhu, and Manik Varma (2016). “Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–944.

Jain, Himanshu et al. (2019). “Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 528–536.

Jalan, Ankit and Purushottam Kar (2019). “Accelerating extreme classification via adaptive feature agglomeration”. In: *arXiv preprint arXiv:1905.11769*.

Jasinska, Kalina et al. (2016). “Extreme f-measure maximization using sparse probability estimates”. In: *International conference on machine learning*. PMLR, pp. 1435–1444.

Jatunarapit, Pornrat, Krerk Piromsopa, and Chris Charoeanlap (2016). “Development of thai text-mining model for classifying ICD-10 TM”. In: *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, pp. 1–6.

Jeblee, Serena et al. (2018). “Toronto CL at CLEF 2018 eHealth Task 1: Multi-lingual ICD-10 Coding using an Ensemble of Recurrent and Convolutional Neural Networks”. In: *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*.

Jernite, Yacine, Anna Choromanska, and David Sontag (2017). “Simultaneous learning of trees and representations for extreme classification and density estimation”. In: *International Conference on Machine Learning*. PMLR, pp. 1665–1674.

Ji, Shaoxiong, Matti Hölttä, and Pekka Marttinen (2021). “Does the Magic of BERT Apply to Medical Code Assignment? A Quantitative Study”. In: *arXiv preprint arXiv:2103.06511*.

Ji, Shaoxiong, Shirui Pan, and Pekka Marttinen (2020). “Medical Code Assignment with Gated Convolution and Note-Code Interaction”. In: *arXiv preprint arXiv:2010.06975*.

Jia, Robin and Percy Liang (2016). “Data recombination for neural semantic parsing”. In: *arXiv preprint arXiv:1606.03622*.

Jia, Zheng et al. (2019). “Using the distance between sets of hierarchical taxonomic clinical concepts to measure patient similarity”. In: *BMC medical informatics and decision making* 19.1, p. 91.

Jiang, Ting et al. (2021). “LightXML: Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification”. In: *arXiv preprint arXiv:2101.03305*.

Jiao, Xiaoqi et al. (2019). “Tinybert: Distilling bert for natural language understanding”. In: *arXiv preprint arXiv:1909.10351*.

Johnson, Alistair EW et al. (2016). “MIMIC-III, a freely accessible critical care database”. In: *Scientific data* 3.1, pp. 1–9.

Jones, K Sparck, Steve Walker, and Stephen E. Robertson (2000). “A probabilistic model of information retrieval: Development and comparative experiments”. In: *Information processing & management* 36.6, pp. 779–840.

Junczys-Dowmunt, Marcin et al. (2018). “Marian: Fast neural machine translation in C++”. In: *arXiv preprint arXiv:1804.00344*.

Kafle, Kushal, Mohammed Yousefhussien, and Christopher Kanan (2017). “Data augmentation for visual question answering”. In: *Proceedings of the 10th International Conference on Natural Language Generation*, pp. 198–202.

Kalyan, Katikapalli Subramanyam and S Sangeetha (2020). “Secnlp: A survey of embeddings in clinical natural language processing”. In: *Journal of biomedical informatics* 101, p. 103323.

Kang, Tian et al. (2017). “EliIE: An open-source information extraction system for clinical trial eligibility criteria”. In: *Journal of the American Medical Informatics Association* 24.6, pp. 1062–1071.

Kapoor, Ashish, Raajay Viswanathan, and Prateek Jain (2012). “Multilabel classification using bayesian compressed sensing”. In: *Advances in neural information processing systems* 25, pp. 2645–2653.

Kavuluru, Ramakanth, Anthony Rios, and Yuan Lu (2015). “An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records”. In: *Artificial intelligence in medicine* 65.2, pp. 155–166.

Khandagale, Sujay, Han Xiao, and Rohit Babbar (2020). “Bonsai: diverse and shallow trees for extreme multi-label classification”. In: *Machine Learning* 109.11, pp. 2099–2119.

Khattak, Faiza Khan et al. (2019). “A survey of word embeddings for clinical text”. In: *Journal of Biomedical Informatics: X* 4, p. 100057.

Kingma, Diederik P and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.

Kiros, Ryan et al. (2015). “Skip-thought vectors”. In: *arXiv preprint arXiv:1506.06726*.

Kobayashi, Sosuke (2018). “Contextual augmentation: Data augmentation by words with paradigmatic relations”. In: *arXiv preprint arXiv:1805.06201*.

Koehn, Philipp (2005). “Europarl: A parallel corpus for statistical machine translation”. In: *MT summit*. Vol. 5. Citeseer, pp. 79–86.

Koopman, Bevan et al. (2012a). “Graph-based concept weighting for medical information retrieval”. In: *Proceedings of the Seventeenth Australasian Document Computing Symposium*, pp. 80–87.

Koopman, Bevan et al. (2012b). “Towards semantic search and inference in electronic medical records: an approach using concept-based information retrieval”. In: *The Australasian medical journal* 5.9, p. 482.

Kotzias, Dimitrios et al. (2014). “Deep multi-instance transfer learning”. In: *arXiv preprint arXiv:1411.3128*.

Kreuzthaler, Markus et al. (2017). “Semantic Technologies for Re-Use of Clinical Routine Data.” In: *eHealth*, pp. 24–31.

Kudo, Taku (2018). “Subword regularization: Improving neural network translation models with multiple subword candidates”. In: *arXiv preprint arXiv:1804.10959*.

Kull, Meelis et al. (2019). “Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration”. In: *arXiv preprint arXiv:1910.12656*.

Kumar, Varun, Ashutosh Choudhary, and Eunah Cho (2020). “Data augmentation using pre-trained transformer models”. In: *arXiv preprint arXiv:2003.02245*.

Kusner, Matt et al. (2015). “From word embeddings to document distances”. In: *International conference on machine learning*. PMLR, pp. 957–966.

Lai, Kenneth H et al. (2015a). “Automated misspelling detection and correction in clinical free-text records”. In: *Journal of biomedical informatics* 55, pp. 188–195.

Lai, Siwei et al. (2015b). “Recurrent convolutional neural networks for text classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29.

Lample, Guillaume and Alexis Conneau (2019). “Cross-lingual language model pretraining”. In: *arXiv preprint arXiv:1901.07291*.

Lample, Guillaume et al. (2016). “Neural architectures for named entity recognition”. In: *arXiv preprint arXiv:1603.01360*.

Lan, Zhenzhong et al. (2019). “Albert: A lite bert for self-supervised learning of language representations”. In: *arXiv preprint arXiv:1909.11942*.

Landauer, Thomas K (2007). “LSA as a theory of meaning”. In: *Handbook of latent semantic analysis* 3, p. 32.

Le, Quoc and Tomas Mikolov (2014). “Distributed representations of sentences and documents”. In: *International conference on machine learning*. PMLR, pp. 1188–1196.

Le, Xuan-Hien et al. (2019). “Application of long short-term memory (LSTM) neural network for flood forecasting”. In: *Water* 11.7, p. 1387.

Leacock, Claudia and Martin Chodorow (1998). “Combining local context and WordNet similarity for word sense identification”. In: *WordNet: An electronic lexical database* 49.2, pp. 265–283.

Lee, Jinhyuk et al. (2020). “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinformatics* 36.4, pp. 1234–1240.

Levy, Omer and Yoav Goldberg (2014). “Neural word embedding as implicit matrix factorization”. In: *Advances in neural information processing systems* 27, pp. 2177–2185.

Li, Fei and Hong Yu (2020). “ICD coding from clinical text using multi-filter residual convolutional neural network”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 8180–8187.

Li, Min et al. (2018). “Automated ICD-9 coding via a deep learning approach”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 16.4, pp. 1193–1202.

Lima, Luciano RS de, Alberto HF Laender, and Berthier A Ribeiro-Neto (1998). “A hierarchical approach to the automatic categorization of medical documents”. In: *Proceedings of the seventh international conference on Information and knowledge management*, pp. 132–139.

Lin, Dekang et al. (1998). “An information-theoretic definition of similarity.” In: *Icml*. Vol. 98. 1998, pp. 296–304.

Lin, Tsung-Yi et al. (2017). “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.

Lipscomb, Carolyn E (2000). “Medical subject headings (MeSH)”. In: *Bulletin of the Medical Library Association* 88.3, p. 265.

Lita, Lucian Vlad et al. (2008). “Large scale diagnostic code classification for medical patient records”. In: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Liu, Hongfang et al. (2013). “An information extraction framework for cohort identification using electronic health records”. In: *AMIA Summits on Translational Science Proceedings* 2013, p. 149.

Liu, Jingzhou et al. (2017). “Deep learning for extreme multi-label text classification”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–124.

Liu, Liyuan et al. (2018). “Empower sequence labeling with task-aware neural language model”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32.

Liu, Quan et al. (2015). “Learning semantic word embeddings based on ordinal knowledge constraints”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1501–1511.

Liu, Sijia et al. (2019a). “Create: Cohort retrieval enhanced by analysis of text from electronic health records using omop common data model”. In: *arXiv preprint arXiv:1901.07601*.

Liu, Yinhan et al. (2019b). “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.

López-García, Guillermo et al. (2021). “Transformers for Clinical Coding in Spanish”. In: *IEEE Access*.

Lu, Jueqing et al. (2020). “Multi-label Few/Zero-shot Learning with Knowledge Aggregated from Multiple Label Graphs”. In: *arXiv preprint arXiv:2010.07459*.

Lund, Kevin and Curt Burgess (1996). “Producing high-dimensional semantic spaces from lexical co-occurrence”. In: *Behavior research methods, instruments, & computers* 28.2, pp. 203–208.

Luo, Jake et al. (2016). “Big data application in biomedical research and health care: a literature review”. In: *Biomedical informatics insights* 8, BII–S31559.

Luong, Minh-Thang et al. (2015). “Multi-task sequence to sequence learning”. In: *arXiv preprint arXiv:1511.06114*.

Luque, Franco M (2019). “Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis”. In: *arXiv preprint arXiv:1909.11241*.

Lussier, Yves A, Lyudmila Shagina, and Carol Friedman (2000). “Automating icd-9-cm encoding using medical language processing: A feasibility study”. In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association, p. 1072.

Lutz, Bernhard, Nicolas Pröllochs, and Dirk Neumann (2018). “Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning”. In: *arXiv preprint arXiv:1901.00400*.

M. Cowie, J. et al. (2001). “A review of Clinical Terms Version 3 (Read Codes) for speech and language record keeping”. In: *International Journal of Language & Communication Disorders* 36.1, pp. 117–126. DOI: 10.1080/13682820116848.

eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1080/13682820116848. URL: https://onlinelibrary.wiley.com/doi/abs/10.1080/13682820116848.

Ma, Xuezhe and Eduard Hovy (2016). “End-to-end sequence labeling via bi-directional lstm-cnns-crf”. In: *arXiv preprint arXiv:1603.01354*.

Manginas, Nikolaos, Ilias Chalkidis, and Prodromos Malakasiotis (2020). “Layer-wise Guided Training for BERT: Learning Incrementally Refined Document Representations”. In: *arXiv preprint arXiv:2010.05763*.

Marimon, Montserrat et al. (2013). “Tibidabo Treebank and IULA Spanish LSP Treebank Train and Test Partitions”. In:

Marimon, Montserrat et al. (2019). “Automatic De-identification of Medical Texts in Spanish: the MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results.” In: *IberLEF@ SEPLN*, pp. 618–638.

Matos-Junior, Osvaldo et al. (2012). “Using taxonomies for product recommendation”. In: *Journal of Information and Data Management* 3.2, pp. 85–85.

McAuley, Julian and Jure Leskovec (2013). “Hidden factors and hidden topics: understanding rating dimensions with review text”. In: *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172.

McAuley, Julian, Rahul Pandey, and Jure Leskovec (2015). “Inferring networks of substitutable and complementary products”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794.

McAuley, Julian et al. (2015). “Image-based recommendations on styles and substitutes”. In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52.

McCann, Bryan et al. (2017). “Learned in translation: Contextualized word vectors”. In: *arXiv preprint arXiv:1708.00107*.

McDonald, Clement J et al. (2003). “LOINC, a universal standard for identifying laboratory observations: a 5-year update”. In: *Clinical chemistry* 49.4, pp. 624–633.

Medini, Tharun et al. (2019). “Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products”. In: *arXiv preprint arXiv:1910.13830*.

Medori, Julia and Cédrick Fairon (2010). “Machine learning and features selection for semi-automatic ICD-9-CM encoding”. In: *Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, pp. 84–89.

Merity, Stephen, Nitish Shirish Keskar, and Richard Socher (2017). “Regularizing and optimizing LSTM language models”. In: *arXiv preprint arXiv:1708.02182*.

Michel, Paul, Omer Levy, and Graham Neubig (2019). “Are sixteen heads really better than one?” In: *arXiv preprint arXiv:1905.10650*.

Miftahutdinov, Zulfat and Elena Tutubalina (2017). “KFU at CLEF eHealth 2017 Task 1: ICD-10 Coding of English Death Certificates with Recurrent Neural Networks.” In: *CLEF (Working Notes)*.

Mikolov, Tomas et al. (2013). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*.

Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

— (1998). *WordNet: An electronic lexical database*. MIT press.

Min, Sewon, Minjoon Seo, and Hannaneh Hajishirzi (2017). “Question answering through transfer learning from large fine-grained supervision data”. In: *arXiv preprint arXiv:1702.02171*.

Mineiro, Paul and Nikos Karampatziakis (2015). “Fast label embeddings for extremely large output spaces”. In: *arXiv preprint arXiv:1503.08873*.

Miranda-Escalada, Antonio et al. (2020). “Overview of automatic clinical coding: annotations, guidelines, and solutions for non-english clinical cases at codiesp track of CLEF eHealth 2020”. In: *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*.

Mirhosseini, Shahin et al. (2014). “Medical free-text to concept mapping as an information retrieval problem”. In: *Proceedings of the 2014 Australasian document computing symposium*, pp. 93–96.

Mittal, Anshul et al. (2021a). “DECAF: Deep Extreme Classification with Label Features”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 49–57.

Mittal, Anshul et al. (Apr. 2021b). “ECLARE: Extreme classification with label graph correlations”. In: *Proceedings of The ACM International World Wide Web Conference*.

Moen, Hans et al. (2015). “Care episode retrieval: distributional semantic models for information retrieval in the clinical domain”. In: *BMC medical informatics and decision making*. Vol. 15. 2. BioMed Central, pp. 1–19.

Mondschein, Christopher F and Cosimo Monda (2019). “The EU’s General Data Protection Regulation (GDPR) in a research context”. In: *Fundamentals of Clinical Data Science*. Springer, Cham, pp. 55–71.

Mrkšić, Nikola et al. (2016). “Counter-fitting word vectors to linguistic constraints”. In: *arXiv preprint arXiv:1603.00892*.

Mrkšić, Nikola et al. (2017). “Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints”. In: *Transactions of the association for Computational Linguistics* 5, pp. 309–324.

Mueller, Jonas and Aditya Thyagarajan (2016). “Siamese recurrent architectures for learning sentence similarity”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1.

Mulyar, Andriy and Bridget T McInnes (2020). “MT-Clinical BERT: Scaling Clinical Information Extraction with Multitask Learning”. In: *arXiv preprint arXiv:2004.10220*.

Munkres, James (1957). “Algorithms for the assignment and transportation problems”. In: *Journal of the society for industrial and applied mathematics* 5.1, pp. 32–38.

Murdoch, Travis B and Allan S Detsky (2013). “The inevitable application of big data to health care”. In: *Jama* 309.13, pp. 1351–1352.

Nallapati, Ramesh et al. (2016). “Abstractive text summarization using sequence-to-sequence rnns and beyond”. In: *arXiv preprint arXiv:1602.06023*.

Narayan, Shashi, Siva Reddy, and Shay B Cohen (2016). “Paraphrase generation from Latent-Variable PCFGs for semantic parsing”. In: *arXiv preprint arXiv:1601.06068*.

Névéol, Aurélie et al. (2016). “Clinical information extraction at the CLEF eHealth evaluation lab 2016”. In: *CEUR workshop proceedings*. Vol. 1609. NIH Public Access, p. 28.

Névéol, Aurélie et al. (2017). “CLEF eHealth 2017 Multilingual Information Extraction task Overview: ICD10 Coding of Death Certificates in English and French.” In: *CLEF (Working Notes)*.

Névéol, Aurélie et al. (2018). “CLEF eHealth 2018 Multilingual Information Extraction Task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian.” In: *CLEF (Working Notes)*.

Neves, M et al. (2019). “Overview of task 1 in CLEF eHealth 2019: indexing German non-technical summaries of animal experiments”. In: CLEF.

Ning, Wenxin, Ming Yu, and Runtong Zhang (2016). “A hierarchical method to automatically encode Chinese diagnoses through semantic similarity estimation”. In: *BMC medical informatics and decision making* 16.1, p. 30.

Nitsuwat, Supot and Wansa Paoin (2012). “Development of ICD-10-TM ontology for a semi-automated morbidity coding system in Thailand”. In: *Methods of information in medicine* 51.06, pp. 519–528.

Olivas, Emilio Soria et al. (2009). *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI Global.

Ollagnier, Anaıs and Hywel Williams (2020). “Text Augmentation Techniques for Clinical Case Classification”. In: *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*.

Organization, World Health (2017). *Global Health Expenditure Database*. `http://apps.who.int/nha/database/Regional_Averages/Index/en`. Accessed: Apr 1, 2021.

Pakhomov, Serguei et al. (2010). “Semantic similarity and relatedness between clinical terms: an experimental study”. In: *AMIA annual symposium proceedings*. Vol. 2010. American Medical Informatics Association, p. 572.

Pakhomov, Serguei VS, James D Buntrock, and Christopher G Chute (2006). "Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques". In: *Journal of the American Medical Informatics Association* 13.5, pp. 516–525.

Pakhomov, Serguei VS et al. (2011). "Towards a framework for developing semantic relatedness reference standards". In: *Journal of biomedical informatics* 44.2, pp. 251–265.

Pan, Sinno Jialin and Qiang Yang (2009). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Park, Hee et al. (2019). "An Information Retrieval Approach to ICD-10 Classification". In: *MEDINFO 2019: Health and Wellbeing e-Networks for All*. IOS Press, pp. 1564–1565.

Partalas, Ioannis et al. (2015). "Lshtc: A benchmark for large-scale text classification". In: *arXiv preprint arXiv:1503.08581*.

Patel, Kevin et al. (2017). "Adapting pre-trained word embeddings for use in medical coding". In: *BioNLP 2017*, pp. 302–306.

Patrick, Jon, Yitao Zhang, and Yefeng Wang (2007). "Developing feature types for classifying clinical notes". In: *Biological, translational, and clinical language processing*, pp. 191–192.

Pattisapu, Nikhil et al. (2020). "Medical concept normalization by encoding target knowledge". In: *Machine Learning for Health Workshop*. PMLR, pp. 246–259.

Pavillon, Gérard and Françoise Laurent (2003). "Certification et codification des causes médicales de décès". In: *Bulletin épidémiologique hebdomadaire* 31.30, pp. 134–138.

Peng, Yifan, Shankai Yan, and Zhiyong Lu (2019). "Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets". In: *arXiv preprint arXiv:1906.05474*.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Pereira, Luis et al. (2013). "ICD9-based text mining approach to children epilepsy classification". In: *Procedia Technology* 9, pp. 1351–1360.

Pereira, Suzanne et al. (2006). "Construction of a semi-automated ICD-10 coding help system to optimize medical and economic coding." In: *MIE*. Citeseer, pp. 845–850.

Perera, Sujan et al. (2013). "Challenges in understanding clinical notes: Why nlp engines fall short and where background knowledge can help". In: *Proceedings of the 2013 international workshop on Data management & analytics for healthcare*, pp. 21–26.

Pérez, Alicia et al. (2018). “Inferred joint multigram models for medical term normalization according to ICD”. In: *International journal of medical informatics* 110, pp. 111–117.

Perotte, Adler et al. (2011). “Hierarchically supervised latent Dirichlet allocation”. In: *Advances in neural information processing systems* 24, pp. 2609–2617.

Perotte, Adler et al. (2014). “Diagnosis code assignment: models and evaluation metrics”. In: *Journal of the American Medical Informatics Association* 21.2, pp. 231–237.

Pestian, John et al. (2007). “A shared task involving multi-label classification of clinical free text”. In: *Biological, translational, and clinical language processing*, pp. 97–104.

Peters, Matthew E et al. (2018). “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365*.

Phang, Jason, Thibault Févry, and Samuel R Bowman (2018). “Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks”. In: *arXiv preprint arXiv:1811.01088*.

Piqueras, Cristina Gómez (2009). “Disociación/anonimización de los datos de salud”. In: *DS: Derecho y salud* 18.1, pp. 43–56.

Polignano, Marco et al. (2020). “A study of Machine Learning models for Clinical Coding of Medical Reports at CodiEsp 2020”. In: *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*.

Popa, I Sandu et al. (2007). “Text categorization for multi-label documents and many categories”. In: *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS’07)*. IEEE, pp. 421–426.

Porter, Martin F (2001). *Snowball: A language for stemming algorithms*.

Porter, Martin F et al. (1980). “An algorithm for suffix stripping.” In: *Program* 14.3, pp. 130–137.

Porter, Michael E and Thomas H Lee (2013). “The strategy that will fix health care”. In: *Harvard business review* 91.12, pp. 24–24.

Pourpanah, Farhad et al. (2020). “A Review of Generalized Zero-Shot Learning Methods”. In: *arXiv preprint arXiv:2011.08641*.

Prabhu, Yashoteja and Manik Varma (2014). “Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 263–272.

Prabhu, Yashoteja et al. (2018a). “Extreme multi-label learning with label features for warm-start tagging, ranking & recommendation”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 441–449.

Prabhu, Yashoteja et al. (2018b). “Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising”. In: *Proceedings of the 2018 World Wide Web Conference*, pp. 993–1002.

Pradhan, Sameer et al. (2014). "Semeval-2014 task 7: Analysis of clinical text". In: *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014*. Citeseer.

Prüss-Üstün, Annette et al. (2016). *Preventing disease through healthy environments: a global assessment of the burden of disease from environmental risks*. World Health Organization.

Pushp, Pushpankar Kumar and Muktabh Mayank Srivastava (2017). "Train once, test anywhere: Zero-shot learning for text classification". In: *arXiv preprint arXiv:1712.05972*.

Radford, Alec et al. (2018). "Improving language understanding by generative pre-training". In:

Radford, Alec et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Ramachandran, Prajit, Peter J Liu, and Quoc V Le (2016). "Unsupervised pretraining for sequence to sequence learning". In: *arXiv preprint arXiv:1611.02683*.

Rei, Marek (2017). "Semi-supervised multitask learning for sequence labeling". In: *arXiv preprint arXiv:1704.07156*.

Rios, Anthony and Ramakanth Kavuluru (2018). "Few-shot and zero-shot multi-label learning for structured label spaces". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2018. NIH Public Access, p. 3132.

Rivera, Renzo and Paloma Martínez (2019). "Deep neural model with enhanced embeddings for pharmaceutical and chemical entities recognition in Spanish clinical text". In: *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pp. 38–46.

Rizzo, Stefano Giovanni et al. (2015). "ICD code retrieval: Novel approach for assisted disease classification". In: *International Conference on Data Integration in the Life Sciences*. Springer, pp. 147–161.

Rodrigues, Jean-Marie et al. (2015). "Semantic alignment between ICD-11 and SNOMED CT". In: *MEDINFO 2015: eHealth-enabled Health*. IOS Press, pp. 790–794.

Rodrigues, Jean Marie et al. (2017). "Is the application of SNOMED CT concept model sufficiently quality assured?" In: *AMIA Annual Symposium Proceedings*. Vol. 2017. American Medical Informatics Association, p. 1488.

Rosenbloom, S Trent et al. (2011). "Data from clinical notes: a perspective on the tension between structure and flexible documentation". In: *Journal of the American Medical Informatics Association* 18.2, pp. 181–186.

Rubin, Timothy N et al. (2012). "Statistical topic models for multi-label document classification". In: *Machine learning* 88.1-2, pp. 157–208.

Ruch, Patrick et al. (2008a). "Automatic medical encoding with SNOMED categories". In: *BMC medical informatics and decision making*. Vol. 8. 1. BioMed Central, pp. 1–8.

Ruch, Patrick et al. (2008b). “From episodes of care to diagnosis codes: automatic text categorization for medico-economic encoding”. In: *AMIA Annual Symposium Proceedings*. Vol. 2008. American Medical Informatics Association, p. 636.

Ruder, Sebastian (2019). “Neural transfer learning for natural language processing”. PhD thesis. NUI Galway.

Rydning, David Reinsel-John Gantz-John (2018). “The digitization of the world from edge to core”. In: *Framingham: International Data Corporation*.

Şahin, Gözde Gül and Mark Steedman (2019). “Data augmentation via dependency tree morphing for low-resource languages”. In: *arXiv preprint arXiv:1903.09460*.

Saini, Deepak et al. (Apr. 2021). “GalaXC: Graph neural networks with labelwise attention for extreme classification”. In: *Proceedings of The ACM International World Wide Web Conference*.

Sajjad, Hassan et al. (2020). “Poor Man’s BERT: Smaller and Faster Transformer Models”. In: *arXiv preprint arXiv:2004.03844*.

Sänger, Mario et al. (2019). “Classifying German Animal Experiment Summaries with Multi-lingual BERT at CLEF eHealth 2019 Task 1.” In: *CLEF (Working Notes)*.

Sanh, Victor et al. (2019). “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108*.

Santiso, Sara et al. (2019). “Word embeddings for negation detection in health records written in Spanish”. In: *Soft Computing* 23.21, pp. 10969–10975.

Savova, Guergana K et al. (2010). “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications”. In: *Journal of the American Medical Informatics Association* 17.5, pp. 507–513.

Schuster, Mike and Kaisuke Nakajima (2012). “Japanese and korean voice search”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5149–5152.

Sen, Cansu et al. (2021). “From Extreme Multi-label to Multi-class: A Hierarchical Approach for Automated ICD-10 Coding Using Phrase-level Attention”. In: *arXiv preprint arXiv:2102.09136*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2015a). “Improving neural machine translation models with monolingual data”. In: *arXiv preprint arXiv:1511.06709*.

— (2015b). “Neural machine translation of rare words with subword units”. In: *arXiv preprint arXiv:1508.07909*.

Seo, Minjoon et al. (2016). “Bidirectional attention flow for machine comprehension”. In: *arXiv preprint arXiv:1611.01603*.

Ševa, Jurica, Mario Sänger, and Ulf Leser (2018). “WBI at CLEF eHealth 2018 Task 1: Language-independent ICD-10 coding using multi-lingual embeddings and recurrent neural networks”. In: *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*.

Severyn, Aliaksei and Alessandro Moschitti (2015). “Twitter sentiment analysis with deep convolutional neural networks”. In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 959–962.

Shimodaira, Hidetoshi (2000). “Improving predictive inference under covariate shift by weighting the log-likelihood function”. In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Siblini, Wissam, Pascale Kuntz, and Frank Meyer (2018). “Craftml, an efficient clustering-based random forest for extreme multi-label learning”. In: *International Conference on Machine Learning*. PMLR, pp. 4664–4673.

Silvestri, Stefano et al. (2020). “Exploit Multilingual Language Model at Scale for ICD-10 Clinical Text Classification”. In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 1–7.

Soares, Felipe et al. (2019). “Medical word embeddings for Spanish: Development and evaluation”. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 124–133.

Socher, Richard et al. (2013). “Recursive deep models for semantic compositionality over a sentiment treebank”. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Søgaard, Anders and Yoav Goldberg (2016). “Deep multi-task learning with low level tasks supervised at lower layers”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–235.

Song, Congzheng et al. (2020). “Generalized Zero-Shot Text Classification for ICD Coding”. In: *IJCAI*.

Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). “Conceptnet 5.5: An open multilingual graph of general knowledge”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31.

Stanfill, Mary H et al. (2010). “A systematic literature review of automated clinical coding and classification systems”. In: *Journal of the American Medical Informatics Association* 17.6, pp. 646–651.

Stickland, Asa Cooper and Iain Murray (2019). “BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning”. In: *International Conference on Machine Learning*. PMLR, pp. 5986–5995.

Stuart-Buttle, Charlie D. G. et al. (1996). “A language of health in action: Read Codes, classifications and groupings.” In: *Proceedings : a conference of the American Medical Informatics Association. AMIA Fall Symposium*, pp. 75–83.

Subotin, Michael and Anthony Davis (2014). “A system for predicting ICD-10-PCS codes from electronic health records”. In: *Proceedings of BioNLP 2014*, pp. 59–67.

Subramanian, Sandeep et al. (2018). “Learning general purpose distributed sentence representations via large scale multi-task learning”. In: *arXiv preprint arXiv:1804.00079*.

Sun, Wei et al. (2021). “Multitask Recalibrated Aggregation Network for Medical Code Prediction”. In: *arXiv preprint arXiv:2104.00952*.

Sun, Zhiqing et al. (2020). “Mobilebert: a compact task-agnostic bert for resource-limited devices”. In: *arXiv preprint arXiv:2004.02984*.

Tagami, Yukihiro (2017). “Annexml: Approximate nearest neighbor search for extreme multi-label classification”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 455–464.

Tai, Farbound and Hsuan-Tien Lin (2012). “Multilabel classification with principal label space transformation”. In: *Neural Computation* 24.9, pp. 2508–2542.

Taulé, Mariona, Maria Antonia Martí, and Marta Recasens (2008). “AnCora: Multilevel Annotated Corpora for Catalan and Spanish.” In: *Lrec*.

Tenney, Ian et al. (2019). “What do you learn from context? probing for sentence structure in contextualized word representations”. In: *arXiv preprint arXiv:1905.06316*.

Tiedemann, Jörg (2012). “Parallel Data, Tools and Interfaces in OPUS.” In: *Lrec*. Vol. 2012, pp. 2214–2218.

Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). “Word representations: a simple and general method for semi-supervised learning”. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 384–394.

Van Mulligen, Erik M et al. (2016). “Erasmus MC at CLEF eHealth 2016: Concept recognition and coding in French texts”. In: *CLEF (Working Notes)*.

Velichkov, Boris et al. (2020). “Automatic ICD-10 codes association to diagnosis: Bulgarian case”. In: *CSBio’20: Proceedings of the Eleventh International Conference on Computational Systems-Biology and Bioinformatics*, pp. 46–53.

Voita, Elena et al. (2019). “Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned”. In: *arXiv preprint arXiv:1905.09418*.

Vulić, Ivan et al. (2018). “Post-specialisation: Retrofitting vectors of words unseen in lexical resources”. In: *arXiv preprint arXiv:1805.03228*.

Wadbude, Rahul et al. (2017). “Leveraging Distributional Semantics for Multi-Label Learning”. In: *arXiv preprint arXiv:1709.05976*.

Wang, Alex et al. (2018). “Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling”. In: *arXiv preprint arXiv:1812.10860*.

Wang, Sen et al. (2017). “Learning multiple diagnosis codes for ICU patients with local disease correlation mining”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.3, pp. 1–21.

Wang, William Yang and Diyi Yang (2015). “That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2557–2563.

Wang, Xiaoyan et al. (2009). "Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study". In: *Journal of the American Medical Informatics Association* 16.3, pp. 328–337.

Wang, Yanshan et al. (2019). "Test collections for electronic health record-based clinical information retrieval". In: *JAMIA open* 2.3, pp. 360–368.

Wei, Jason and Kai Zou (2019). "Eda: Easy data augmentation techniques for boosting performance on text classification tasks". In: *arXiv preprint arXiv:1901.11196*.

Wei, Xing and Carsten Eickhoff (2018). "Embedding electronic health records for clinical information retrieval". In: *arXiv preprint arXiv:1811.05402*.

Weinberg, Jason et al. (2015). "Aligning computer-assisted coding and information governance efforts". In: *Journal of AHIMA* 86.10, pp. 36–40.

Weiss, Karl, Taghi M Khoshgoftaar, and DingDing Wang (2016). "A survey of transfer learning". In: *Journal of Big data* 3.1, pp. 1–40.

Weston, Jason, Samy Bengio, and Nicolas Usunier (2011). "Wsabie: Scaling up to large vocabulary image annotation". In:

Weston, Jason, Ameesh Makadia, and Hector Yee (2013). "Label partitioning for sublinear ranking". In: *International conference on machine learning*. PMLR, pp. 181–189.

Wetzker, Robert, Carsten Zimmermann, and Christian Bauckhage (2008). "Analyzing social bookmarking systems: A del. icio. us cookbook". In: *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pp. 26–30.

WHO, World Health Organization (2004). *ICD-10 : international statistical classification of diseases and related health problems / World Health Organization*. English. 10th revision, 2nd ed. World Health Organization Geneva, 3 v. : ISBN: 9241546492 9241546530 9241546549.

Wieting, John and Kevin Gimpel (2017). "Revisiting recurrent networks for paraphrastic sentence embeddings". In: *arXiv preprint arXiv:1705.00364*.

Wieting, John and Douwe Kiela (2019). "No training required: Exploring random encoders for sentence classification". In: *arXiv preprint arXiv:1901.10444*.

Wieting, John et al. (2015). "Towards universal paraphrastic sentence embeddings". In: *arXiv preprint arXiv:1511.08198*.

Wiley, Miriam M (2014). "Diagnosis related groups (DRGs): Measuring hospital case mix". In: *Wiley StatsRef: Statistics Reference Online*.

Wu, Zhibiao and Martha Palmer (1994). "Verb semantics and lexical selection". In: *arXiv preprint cmp-lg/9406033*.

Xie, Qizhe et al. (2019). "Unsupervised data augmentation for consistency training". In: *arXiv preprint arXiv:1904.12848*.

Xie, Ziang et al. (2017). "Data noising as smoothing in neural network language models". In: *arXiv preprint arXiv:1703.02573*.

Xu, Chang, Dacheng Tao, and Chao Xu (2016). "Robust extreme multi-label learning". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1275–1284.

Xu, Chang et al. (2014). "Rc-net: A general framework for incorporating knowledge into word representations". In: *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 1219–1228.

Xu, Hua et al. (2010). "MedEx: a medication information extraction system for clinical narratives". In: *Journal of the American Medical Informatics Association* 17.1, pp. 19–24.

Yamada, Hitomi et al. (2010). "Analysis of Human and System Factors on Errors in ICD Coding with Electronic Discharge Summary System". In: *First IMIA/IFIP Joint Symposium on E-Health (E-HEALTH)/Held as Part of World Computer Congress (WCC)*. Springer, pp. 234–235.

Yan, Yan et al. (2010). "Medical coding classification by leveraging inter-code relationships". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 193–202.

Yang, Jie, Yue Zhang, and Fei Dong (2017). "Neural word segmentation with rich pretraining". In: *arXiv preprint arXiv:1704.08960*.

Yang, Zhilin et al. (2019). "Xlnet: Generalized autoregressive pretraining for language understanding". In: *arXiv preprint arXiv:1906.08237*.

Yang, Zichao et al. (2016). "Hierarchical attention networks for document classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489.

Yao, Yi and Gianfranco Doretto (2010). "Boosting for transfer learning with multiple sources". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, pp. 1855–1862.

Yeh, Chih-Kuan et al. (2017). "Learning deep latent space for multi-label classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1.

Yen, Ian EH et al. (2017). "Ppdsparse: A parallel primal-dual sparse method for extreme classification". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 545–553.

Yih, Wen-tau, Geoffrey Zweig, and John C Platt (2012). "Polarity inducing latent semantic analysis". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1212–1222.

You, Ronghui et al. (2018). "Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification". In: *arXiv preprint arXiv:1811.01727*.

You, Ronghui et al. (2019). "HAXMLNet: Hierarchical attention network for extreme multi-label text classification". In: *arXiv preprint arXiv:1904.12578*.

Yu, Hsiang-Fu et al. (2014). "Large-scale multi-label learning with missing labels". In: *International conference on machine learning*. PMLR, pp. 593–601.

Yu, Mo and Mark Dredze (2014). "Improving lexical embeddings with semantic knowledge". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 545–550.

Yu, Sheng, Damon Berry, and Jesus Bisbal (2011). "Performance analysis and assessment of a tf-idf based archetype-SNOMED-CT binding algorithm". In: *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, pp. 1–6.

Zhang, Danchen et al. (2017). "Enhancing automatic icd-9-cm code assignment for medical texts with pubmed". In: *BioNLP 2017*, pp. 263–271.

Zhang, Jingqing, Piyawat Lertvittayakumjorn, and Yike Guo (2019). "Integrating semantic knowledge to tackle zero-shot text classification". In: *arXiv preprint arXiv:1903.12626*.

Zhang, Kelly W and Samuel R Bowman (2018). "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis". In: *arXiv preprint arXiv:1809.10040*.

Zhang, Wenjie et al. (2018). "Deep extreme multi-label learning". In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 100–107.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). "Character-level convolutional networks for text classification". In: *arXiv preprint arXiv:1509.01626*.

Zhang, Yijia et al. (2019). "BioWordVec, improving biomedical word embeddings with subword information and MeSH". In: *Scientific data* 6.1, pp. 1–9.

Zhang, Yitao (2008). "A hierarchical approach to encoding medical concepts for clinical notes". In: *proceedings of the ACL-08: HLT Student Research Workshop*, pp. 67–72.

Zhang, Zachariah, Jingshu Liu, and Narges Razavian (2020). "BERT-XML: Large Scale Automated ICD Coding Using BERT Pretraining". In: *arXiv preprint arXiv:2006.03685*.

Zhu, Dongqing et al. (2013). "Using Discharge Summaries to Improve Information Retrieval in Clinical Domain." In: *CLEF (Working Notes)*.

Zipf, George Kingsley (1950). "Human behavior and the principle of least effort: An introduction to human eoclogy". In: *Social Forces* 28.3, pp. 340–341.

Zoph, Barret et al. (2016). "Transfer learning for low-resource neural machine translation". In: *arXiv preprint arXiv:1604.02201*.

Zubiaga, Arkaitz (2012). "Enhancing navigation on wikipedia with social tags". In: *arXiv preprint arXiv:1202.5469*.

Zuccon, Guido et al. (2012). “Exploiting medical hierarchies for concept-based information retrieval”. In: *Proceedings of the seventeenth Australasian document computing symposium*, pp. 111–114.

Zweigenbaum, Pierre and Thomas Lavergne (2016). “Hybrid methods for ICD-10 coding of death certificates”. In: *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pp. 96–105.