UNED
Escuela Internacional de Doctorado
EIDUNED

DOCTORAL THESIS

2023

Recent advances on sentence similarity methods, software and resources for the biomedical domain

ALICIA LARA CLARES

DOCTORAL PROGRAMME ON INTELLIGENT SYSTEMS

Supervisors: Prof. Dr. Ana García Serrano
and Dr. Juan José Lastra Díaz

This page intentionally left blank.

A José Javier Mesas Clares,
la estrella que nació
para ser infinita.

A mi familia,
porque este triunfo
es gracias a vosotros.

Contents

List of Tables

List of Figures

Abstract

Measuring semantic similarity between sentences is a significant task in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and biomedical text mining. For this reason, the proposal of sentence similarity methods for the biomedical domain has attracted a lot of attention in recent years. However, most sentence similarity methods and experimental results reported in the biomedical domain cannot be reproduced for multiple reasons: the copying of previous results without confirmation, the lack of source code and data to replicate both methods and experiments, and the lack of a detailed definition of the experimental setup, among others. As a consequence of this reproducibility gap, the state of the problem can be neither elucidated nor new lines of research be soundly established.

In addition, there are other significant gaps in the literature on biomedical sentence similarity, such as: (1) the performance and scalability drawbacks in current state-of-the-art semantic measures libraries for the biomedical domain; (2) the lack of an efficient shortest-path algorithm for real-time computation of path-based semantic similarity measures; (3) the evaluation of several unexplored sentence similarity methods which deserve to be studied; (4) the evaluation of an unexplored benchmark on biomedical sentence similarity, called Corpus-Transcriptional-Regulation (CTR); (5) a study on the impact of the pre-processing stage and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods; and finally, (6) the lack of software and data resources for the reproducibility of methods and experiments in this line of research. Despite the research effort carried out in this area, we believe that there is room for improvement in the development of specific methods, since current methods are adaptations of general domain methods. In addition, the research community has focused on Deep Learning methods with no previous evaluation of different alternatives.

This thesis introduces the largest, and for the first time, a reproducible experimental survey on biomedical sentence similarity, as well as the proposal and evaluation of new methods for estimating the degree of similarity between sentences. In addition, this thesis makes several significant contributions to the reproducibility of sentence similarity benchmarks and measures as follows: (1) a detailed reproducibility protocol together with a collection of software tools and dataset; (2) an updated and extended version of the *Half-Edge Semantic Measures Library (HESML)* for the biomedical domain, called *HESML V1R5*; (3) a fast approximation of Dijkstra's algorithm for taxonomies based on a relaxed graph spanner, called *Ancestors-based Shortest-Path Length (AncSPL)*; (4) the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity; (5) the evaluation of a set of previously unexplored methods, such as a new string-based sentence similarity method, called *LiBlock*, eight variants of the current ontology-based methods from the literature, and a new pre-trained Word Embedding (WE) model based on FastText and trained on the full-text of articles in the PMC-BioC corpus; (6) the evaluation for the first time of an unexplored benchmark, called *Corpus-Transcriptional-Regulation (CTR)*; (7) the study on the impact of the pre-processing stage and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods; (8) the integration for the first time of most sentence similarity methods for the biomedical domain into the same software library, called *HESML for Semantic Textual Similarity (HESML-STS)*; and finally, (9) an analysis of the drawbacks and limitations of the current state-of-the-art methods.

Our experiments show that our novel string-based measure establishes the new state of the art for the sentence similarity task in the biomedical domain and significantly outperforms all the methods evaluated herein, with the only exception of one ontology-based method. Likewise, our experiments confirm that the pre-processing stages, and the choice of the NER tool for ontology-based methods, have a very significant impact on the performance of the sentence similarity methods. We also detail some drawbacks and limitations of current methods, and warn about the need to refine the current benchmarks. Finally, a noticeable finding is that our new string-based method significantly outperforms all state-of-the-art Machine Learning (ML) models evaluated herein.

Keywords: HESML, semantic similarity measures, sentence similarity, semantic measures libraries, ontology-based semantic similarity measures, SNOMED-CT, MeSH, reproducible survey.

Resumen

Medir la similitud semántica entre oraciones es una tarea importante en los campos del Procesamiento del Lenguaje Natural (PLN), la Recuperación de Información (RI) y la minería de textos biomédicos. Por este motivo, la propuesta de métodos de similitud de frases para el ámbito biomédico ha atraído mucha atención en los últimos años. Sin embargo, la mayoría de los métodos de similitud de frases y resultados experimentales reportados en el dominio biomédico no pueden ser reproducidos por múltiples razones como las siguientes: la copia de resultados previos sin confirmación, la falta de código fuente y datos para replicar tanto los métodos como los experimentos, y la falta de una definición detallada de la configuración experimental, entre otras. Como consecuencia de este vacío de reproducibilidad, no se puede dilucidar el estado del problema ni establecer sólidamente nuevas líneas de investigación.

Por otro lado, existen otras lagunas significativas en la literatura sobre similitud de frases biomédicas, como son: (1) las limitaciones de rendimiento y escalabilidad de las actuales bibliotecas de medidas semánticas de última generación para el ámbito biomédico; (2) la falta de un algoritmo eficiente de camino más corto para el cálculo en tiempo real de medidas de similitud semántica basadas en caminos; (3) la evaluación de varios métodos de similitud de oraciones inexplorados que merecen ser estudiados; (4) la evaluación de un conjunto de datos inexplorado sobre similitud de oraciones biomédicas, denominado Corpus-Transcriptional-Regulation (CTR); (5) el estudio sobre el impacto de la etapa de preprocesamiento y las herramientas de Reconocimiento de Entidades Nombradas (NER) sobre el rendimiento de los métodos de similitud de oraciones; y, por último, (6) la falta de recursos de software y datos para la reproducibilidad de métodos y experimentos en esta línea de investigación. A pesar del esfuerzo investigador realizado en este campo, creemos que hay margen de mejora en el desarrollo de métodos específicos, ya que los actuales son adaptaciones de métodos de dominio general. Además, la comunidad investigadora se ha centrado en métodos de Deep Learning sin una evaluación previa de diferentes alternativas.

Esta tesis introduce el mayor, y por primera vez, estudio experimental reproducible sobre similitud de frases biomédicas, así como la propuesta y evaluación de nuevos métodos para estimar el grado de similitud entre oraciones. Además, esta tesis introduce varias contribuciones significativas a la reproducibilidad de las medidas de similitud entre oraciones, a saber (1) un protocolo detallado de reproducibilidad junto con una colección de herramientas de software y un conjunto de datos; (2) una versión actualizada y ampliada de la *Half-Edge Semantic Measures Library (HESML)* para el dominio biomédico, llamada *HESML V1R5*; (3) una aproximación rápida del algoritmo de Dijkstra para taxonomías basado en grafos, llamado *Ancestors-based Shortest-Path Length (AncSPL)*; (5) la evaluación de un conjunto de métodos hasta ahora inexplorados, como un nuevo método de similitud de oraciones basado en cadenas, denominado *LiBlock*, ocho variantes de los métodos actuales basados en ontologías y un nuevo modelo de *word embeddings* (Word Embedding, WE) preentrenado basado en FastText y entrenado con el texto completo de los artículos del corpus PMC-BioC; (6) la evaluación por primera vez de un conjunto de datos, denominado *Corpus-Transcriptional-Regulation (CTR)*; (7) el estudio del impacto de la etapa de preprocesamiento y de las herramientas de reconocimiento de entidades con nombre (NER) en el rendimiento de los métodos de similitud de frases; (8) la integración por primera vez de la mayoría de los métodos de similitud de oraciones para el ámbito biomédico en la misma biblioteca de software, denominada *HESML for Semantic Textual Similarity (HESML-STS)*; y, por último, (9) un análisis de los inconvenientes y limitaciones de los métodos actuales de vanguardia.

Nuestros experimentos demuestran que nuestra novedosa medida basada en cadenas establece el nuevo estado del arte en la tarea de similitud de oraciones en el ámbito biomédico y supera significativamente a todos los métodos aquí evaluados, con la única excepción de un método basado en ontologías. Asimismo, nuestros experimentos confirman que las etapas de preprocesamiento, y la elección de la herramienta NER para los métodos basados en ontologías, tienen un impacto muy significativo en el rendimiento de los métodos de similitud de frases. También detallamos algunos inconvenientes y limitaciones de los métodos actuales, y advertimos de la necesidad de perfeccionar las pruebas de referencia actuales. Por último, un hallazgo notable es que nuestro nuevo método basado en cadenas supera significativamente a todos los modelos de Aprendizaje Automático (Machine Learning, ML) de última generación aquí evaluados.

Palabras clave: HESML, medidas de similitud semántica, similitud de oraciones, librerías de medidas semánticas, medidas de similitud semántica basadas en ontologías, SNOMED-CT, MeSH, survey reproducible.

Acknowledgements

“So much universe, and so little time.”
Terry Pratchett

This thesis is a milestone in a new career, an experience acquired over the last few years, a new way of approaching my work and personal life, and a new beginning. For this reason, I would like to thank all the people who, directly or indirectly, have contributed to making this possible:

To my directors, Ana María García Serrano and Juan José Lastra Díaz, for their tireless support, the paperwork and revisions, the infinite patience and all the recommendations and learning that you have given me. If I take anything away with me after finishing my thesis, it is precisely what I have learned from you.

To all the members of the LSI department of the UNED, who were for years the pillar of my stay in Madrid, from whom I have received hundreds of pieces of advice, who have been there in the best and in the worst moments. To all the colleagues who were there and those who are still there: Andrés Duque, Agustín Delgado, Bernardo Cabaleiro, Ángel Castellanos, Javier Rodríguez, Hermenegildo Fabregat, Mario Almagro, and a long list of people who have made all this even more worthwhile. In addition, I would like to thank Covadonga Rodríguez and Luis Miguel de Frutos for allowing me to be part of a research team that is as necessary as it is interesting.

To all the people who have contributed to the development of this thesis. To Fernando González and Juan Corrales for their invaluable help in creating the datasets, to Gizem Sogancioglu and Kathrin Blagec for answering the questions to replicate their methods and experiments, to Hongfang Liu and Yanshan Wang for providing the MedSTS dataset, to David Pritchard for reviewing the use of English in the articles and of course to all the anonymous reviewers for their comments that have improved the quality of the articles and the work done in this thesis.

And above all, I would like to thank my family and friends, my sustenance and pillar, my greatest source of strength and encouragement, those who make every step along the way worthwhile, those who never give up, and never let me give up. All of this is as much yours as it is mine.

Alicia Lara-Clares
Madrid, April 2023

Institutional acknowledgements

This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018) and the CLARA-HD (PID2020-116001RB-C32) project.

Part I

Thesis by Compendium

Chapter 1

Introduction

Measuring semantic similarity between sentences is an important task in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and biomedical text mining, among others. For instance, the estimation of the degree of semantic similarity between sentences is used in text classification [124, 55, 29], question answering [114, 58], evidence sentence retrieval to extract biological expression language statements [111, 110], biomedical document labeling [37], biomedical event extraction [88], named entity recognition [42], evidence-based medicine [56, 46], biomedical document clustering [21], prediction of adverse drug reactions [34], entity linking [60], document summarization [9, 117] and sentence-driven search of biomedical literature [10], among other applications. In the question answering task, Sarrouti and El Alaomi [114] build a ranking of plausible answers by computing the similarity scores between each biomedical question and the candidate sentences extracted from a knowledge corpus. Allot et al. [10] introduce a system to retrieve the most similar sentences in the BioC biomedical corpus [31], called Litsense [10], which is based on the comparison of the user query with all sentences in the aforementioned corpus. In this area, the relevance of the research is also endorsed by recent works based on sentence similarity measures, such as the work of Aliguliyev [9] in automatic document summarization, which shows that the performance of these applications depends significantly on the sentence similarity measures used.

The aim of any semantic similarity method is to estimate the degree of similarity between two textual semantic units as perceived by a human being, such as words, phrases, sentences, short texts, or documents. Unlike sentences from the language in general use, whose vocabulary and syntax is limited both in extension and complexity, most sentences in the biomedical domain are comprised of a huge specialized vocabulary made up of all sorts of biological and clinical terms, in addition to an uncountable list of acronyms, which are combined in complex lexical and syntactic forms. For these latter reasons, any kind of language processing task for the biomedical domain, such as that tackled in this thesis, is extremely challenging.

Most methods on biomedical sentence similarity are adaptations from methods for the general language domain, which are mainly based on the use of biomedical ontologies, as well as word and sentence embedding models trained on biomedical text corpora. For instance, Socioanglu et al. [120] introduce a set of sentence similarity measures for the biomedical domain, which are based on adaptations from the Li et al. [84] measure. Zhang et al. [135] introduce a set of pre-trained word embedding model called BioWordVec, which is based on a FastText [19] model trained on the titles and abstracts from PubMed articles and term sequences from the Medical Subject Headings (MeSH) thesaurus [97], whilst Chen et al. [30] introduce a set of pre-trained sentence embedding models called BioSentVec, which is based on a Sent2vec [101] model trained on the full text of PubMed articles and Medical Information Mart for Intensive Care (MIMIC-III) clinical notes [51], and Blagec et al. [18] introduce a set of word and sentence embedding models based on the training of FastText [19], Sent2Vec [101], Paragraph vector [80], and Skip-thoughts vectors [57] models on the full-text PubMed Central (PMC) Open Access dataset. Likewise, several contextualized word representation models, also known as language models, have also been adapted to the biomedical domain. For instance, Lee et al. [81] and Peng et al. [104] introduce two language models based on the Bidirectional Encoder Representations from Transformers (BERT) architecture [33], which are called BERT for Biomedical text mining (BioBERT) and Biomedical Language Understanding Evaluation of BERT (BlueBERT), respectively. Despite the research effort carried out in this area, we believe that there is room for improvement in the development of specific methods since current methods are adaptations of general domain methods. In addition, the research community has focused on Deep Learning methods with no previous evaluation of different alternatives.

Currently, there are several works in the literature that experimentally evaluate multiple methods on biomedical sentence similarity. However, they are either theoretical or have a limited scope and cannot be reproduced. Moreover, there is also a significant lack of reproducibility software and data resources in this area. For instance, Kalyan et al. [53], Khattak et al. [54], and Alsentzer et al. [11] introduce theoretical surveys on biomedical embeddings with a limited scope; and the experimental surveys introduced by Sogancioglu et al. [120], Blagec et al. [18], Peng et al. [104], and Chen et al. [30] among other authors, cannot be reproduced because of the lack of source code and data to replicate both methods and experiments, or the lack of a detailed definition of their experimental setups. Likewise, there are other recent works whose results need to be confirmed. For instance, Tawfik and Spruit [125] experimentally evaluate a set of pre-trained language models, whilst Chen et al. [27] propose a system to study the impact of a set of similarity measures on a Deep Learning ensemble model, which is based on a Random Forest model [22].

Ontology-based semantic similarity measures based on SNOMED-CT, MeSH, and Gene Ontology are being extensively used in many applications in biomedical text mining and genomics respectively, which has encouraged the development of semantic measures libraries based on the aforementioned ontologies. However, current state-of-the-art semantic measures libraries have some performance and scalability drawbacks derived from their ontology representations based on relational databases, or naive in-memory graph representations, which limit their use in high-throughput experiments and applications. Likewise, a recent reproducible survey on word similarity [71] shows that one hybrid IC-based measure which integrates a shortest-path computation [68, coswJ&C] establishes the state of the art in the family of ontology-based semantic measures. However, the lack of an efficient shortest-path algorithm for their real-time computation prevents both their practical use in any application and the use of any other path-based semantic similarity measure.

The main ontologies used for biomedical text mining and information retrieval applications in health sciences are SNOMED-CT and MeSH, although there are many other ontologies[1] based on the OBO file format [119]. At the present time there are only two semantic measures libraries based on the two aforementioned ontologies as follows: (1) the pioneering Perl software library and online web interface called UMLS::Similarity [91], and (2) the most recent Java software library called SML [44], which introduces several significant contributions, such as portable and efficient object-oriented language programming, as well as a significant number of methods, and the implementation for the first time of the most significant biomedical ontologies and WordNet into a single software library. Despite the UMLS::Similarity library [91] having been extensively used in the literature, it has several significant drawbacks that prevent its use in high-throughput standalone applications, namely a poor performance in the evaluation of measures, as well as a tedious, complex, and long setup process to build several pre-calculated data structures and values stored into an auxiliary database called UMLS::Interface. Likewise, SML [44] has several significant performance and scalability drawbacks derived from the use of a naive in-memory graph representation based on hash tables and caching, which significantly impacts its overall performance, and very especially, its computation of path-based measures and scalability regarding the ontology size [69]. In addition to the aforementioned drawbacks which encourage our research in this thesis, neither UMLS:Similarity nor SML implement most of the measures based on embedding and language models, such as those evaluated in this thesis [66].

The main aim of this thesis is to bridge the gaps introduced in the paragraphs above, which we can summarize as follows: (1) the proposal and evaluation of new sentence similarity methods for the biomedical domain that are not adaptations of the general domain; (2) the proposal and development of a new benchmark on semantic measures libraries for the biomedical domain; (3) the proposal and evaluation of a protocol for reproducing all the results reported in this thesis; (3) the development of reproducible software and resources for the evaluation of most sentence similarity methods for the biomedical domain into the same software library; (4) the proposal and evaluation of an extension of Half-Edge Semantic Measures Library (HESML) library [69] to integrate most important biomedical ontologies; and (5) the proposal and evaluation of an efficient shortest-path algorithm for real-time computation of path-based word similarity methods.

1.1 Definition of the research problem

The two main research problems tackled by this thesis are the design and implementation of reproducible benchmarks on biomedical sentence similarity, and the proposal and evaluation of new methods for estimating the degree of similarity between sentences in the biomedical domain.

The research of this thesis tackles the following five main research problems: (1) the design and implementation of a reproducible experimental survey on sentence similarity measures for the biomedical domain; (2) the proposal and evaluation of a new method for the approximation of Dijkstra's algorithm for taxonomies, called Ancestors-based Shortest-Path Length (AncSPL), which allows the real-time com-

[1] http://www.obofoundry.org

putation of any path-based semantic similarity measure; (3) the development and release of an updated version of the HESML [69] library especially designed for the biomedical domain, called HESML V1R5 [74], as well as all the necessary resources for reproducing all experiments reported in the latter article; (4) the development of a new aggregated measure for biomedical sentence similarity; and (5) the evaluation of a set of unexplored methods based on adaptations from previous methods used in the general language domain.

All our experiments are based on our software implementation and evaluation of all methods analyzed in this thesis into two releases of a common and new software platform based on an extension of the Half-Edge Semantic Measures Library (HESML) [69, 77] [2]. The first release, HESML V1R5 [74], extends HESML to the biomedical domain by implementing the SNOMED-CT, MeSH, GO [14, 127], and OBO file format ontologies [119], in addition to WordNet [93]. The second release, HESML V2R1 implements most of the known methods for biomedical sentence similarity, as well as a set of new sentence similarity methods adapted from their definitions for the general-language domain. All our experiments have been recorded in Docker virtualization images that are provided as supplementary material with our software, detailed reproducibility protocols and datasets [76, 64, 63] to allow the easy replication of all our methods, experiments, and results. Thus, this thesis focuses on the reproducibility of the results to elucidate the current state of the art in the task of semantic similarity of sentences in the biomedical domain.

1.2 Brief review of the literature

Current methods on sentence semantic similarity can be categorized into two classes as follows: (a) the methods proposed for the general domain; and (b) the methods proposed for the biomedical domain. For a more detailed presentation of the methods categorized herein, we refer the reader to our protocol [63] and reproducibility survey [61], as well as the surveys on biomedical embedding models [53, 54, 11], ontology-based semantic similarity measures [73, 68], word embeddings [73, 54], sentence embeddings [95, 53], and neural language models [53, 15].

1.2.1 Methods proposed for the general language domain

There is a large corpus of literature on sentence similarity methods for the general language domain as the result of a significant research effort during the last decade. However, the literature for the biomedical domain is much more limited. Research for the general language domain has mainly been boosted by the SemEval Short Text Similarity (STS) evaluation series since 2012 [5, 6, 3, 2, 4, 24], which has generated a large number of contributions in the area [113, 16, 43, 121, 123], as well as an STS benchmark dataset [24]. On the other hand, the development of sentence similarity benchmarks for the biomedical domain is much more recent. Currently, there are only three datasets for the evaluation of methods on biomedical sentence similarity, called BIOSSES [120], MedSTS [130], and CTR [87]. BIOSSES was introduced in 2017 and it is limited to 100 sentence pairs with their corresponding similarity

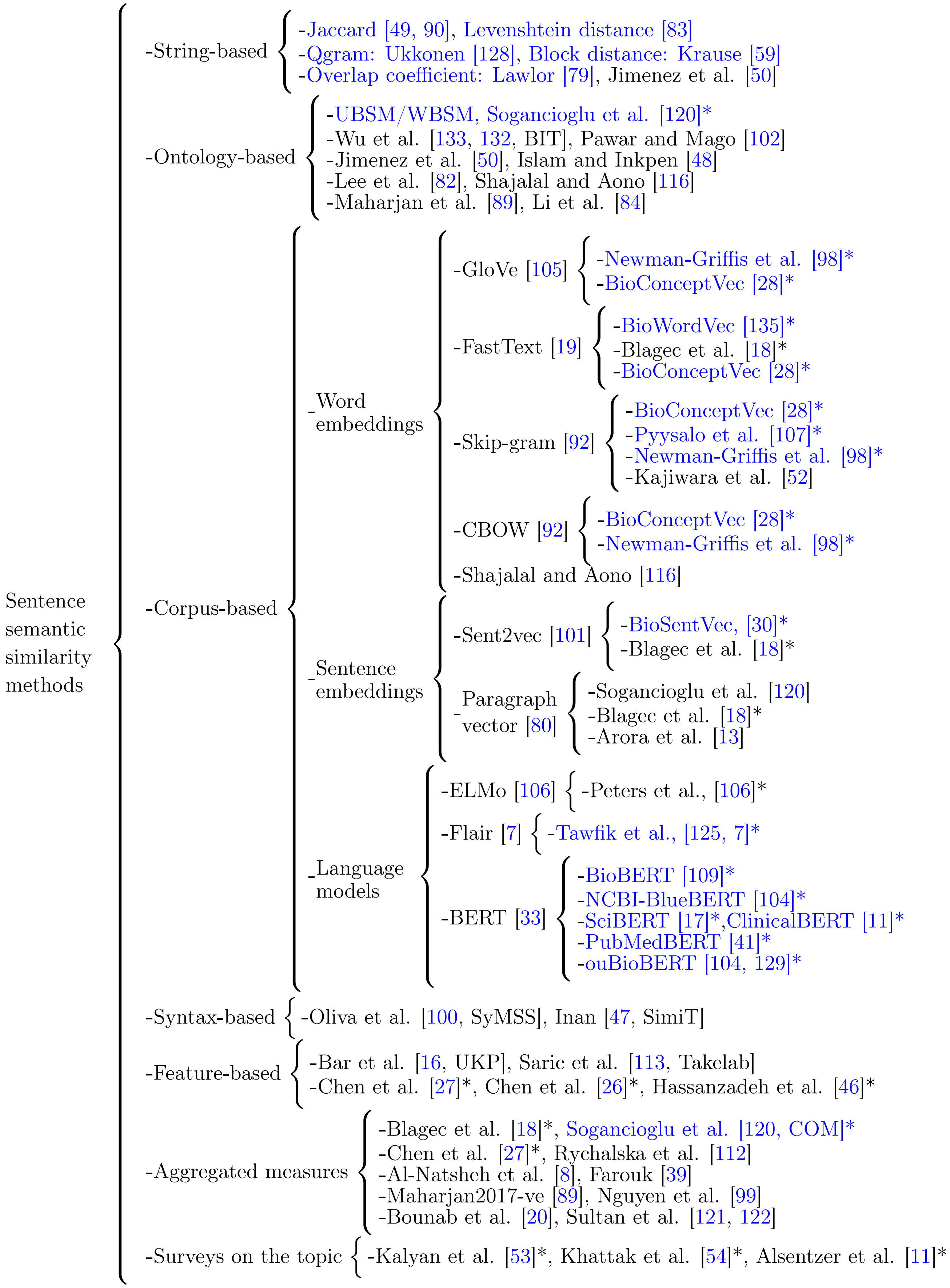
[2] http://hesml.lsi.uned.es

scores, whilst MedSTS$_{full}$ is made up of 1,068 scored sentence pairs from the MedSTS dataset [131], which contains 174,629 sentence pairs gathered from a clinical corpus on biomedical sentence similarity. Finally, the CTR dataset includes 171 sentence pairs, but it has not been evaluated yet because of its recent publication in 2019.

Figure 1.1 shows our categorization of the current sentence semantic similarity measures into six subfamilies as follows. First, string-based measures, whose main feature is the use of the explicit information contained at the character or word level in the sentences to estimate their similarity. Second, ontology-based measures, such as those introduced by Sogancioglu et al. [120], whose main feature is the computation of the similarity between sentences by combining the pairwise similarity scores of their constituent words and concepts [73] based on the Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [36] and WordNet [94] ontologies, and the MeSH thesaurus [97]. Third, corpus-based methods based on the distributional hypothesis [45], such as the work of Pyysalo et al. [107], which states that words sharing semantic relationships tend to occur in similar contexts. The corpus-based methods can be divided into three subcategories as follows: (a) methods based on word embeddings, (b) sentence embeddings, and (c) language models. Methods based on word embeddings combine the word vectors corresponding to the words contained in a sentence to build a sentence vector, such as the averaging Simple Word EMbeddings (SWEM) models introduced by Shen et al. [118], whilst methods based on sentence embeddings directly compute a vector representation for each sentence. Then, the similarity between sentence pairs is calculated using any vector-based similarity metric, such as the cosine function. In contrast, language models explore the concept of Transfer Learning by creating a pre-trained model on a large raw text corpus and fine-tuning that model in downstream tasks, such as sentence semantic similarity, with the pioneering work of Peng et al. [104]. The fourth subfamily consists of the syntax-based methods, which rely on the use of explicit syntax information, as well as the structure of the words that compound the sentences, such as the pioneering work of Oliva et al. [100]. Fifth, feature-based approaches, such as the work of Chen et al. [26], whose main idea is to compute the similarity of two sentences by measuring, according to different language perspectives, the properties that they have in common or not, such as lexical patterns, word semantics and named entities. Finally, aggregated methods, whose main feature is the combination of other sentence similarity methods.

1.2.2 Methods proposed for the biomedical domain

Like that mentioned in the introduction, most methods on biomedical sentence similarity are adaptations from the general domain, such as the methods which are evaluated in this thesis. Sogancioglu et al. [120] proposed a set of ontology-based measures called WordNet-based Similarity Measure (WBSM) and UMLS-based Similarity Measure (UBSM), which are based on the Li et al. [84] measure. All word and sentence embedding models for the biomedical domain in the literature are based on well-known models from the general domain. Pyysalo et al. [107] train a Skip-gram [92] model on document titles and abstracts from the PubMed XML dataset, and all text content of the PMC Open Access dataset. Newman-Griffis et al. [98] and Chen et al. [28] train GloVe [105], Skip-gram, and Continuous Bag of



Sentence semantic similarity methods

- -String-based
  - -Jaccard [49, 90], Levenshtein distance [83]
  - -Qgram: Ukkonen [128], Block distance: Krause [59]
  - -Overlap coefficient: Lawlor [79], Jimenez et al. [50]
- -Ontology-based
  - -UBSM/WBSM, Sogancioglu et al. [120]*
  - -Wu et al. [133, 132, BIT], Pawar and Mago [102]
  - -Jimenez et al. [50], Islam and Inkpen [48]
  - -Lee et al. [82], Shajalal and Aono [116]
  - -Maharjan et al. [89], Li et al. [84]
- -Corpus-based
  - -Word embeddings
    - -GloVe [105]
      - -Newman-Griffis et al. [98]*
      - -BioConceptVec [28]*
    - -FastText [19]
      - -BioWordVec [135]*
      - -Blagec et al. [18]*
      - -BioConceptVec [28]*
    - -Skip-gram [92]
      - -BioConceptVec [28]*
      - -Pyysalo et al. [107]*
      - -Newman-Griffis et al. [98]*
      - -Kajiwara et al. [52]
    - -CBOW [92]
      - -BioConceptVec [28]*
      - -Newman-Griffis et al. [98]*
    - -Shajalal and Aono [116]
  - -Sentence embeddings
    - -Sent2vec [101]
      - -BioSentVec, [30]*
      - -Blagec et al. [18]*
    - -Paragraph vector [80]
      - -Sogancioglu et al. [120]
      - -Blagec et al. [18]*
      - -Arora et al. [13]
  - -Language models
    - -ELMo [106]
      - -Peters et al., [106]*
    - -Flair [7]
      - -Tawfik et al., [125, 7]*
    - -BERT [33]
      - -BioBERT [109]*
      - -NCBI-BlueBERT [104]*
      - -SciBERT [17]*,ClinicalBERT [11]*
      - -PubMedBERT [41]*
      - -ouBioBERT [104, 129]*
- -Syntax-based
  - -Oliva et al. [100, SyMSS], Inan [47, SimiT]
- -Feature-based
  - -Bar et al. [16, UKP], Saric et al. [113, Takelab]
  - -Chen et al. [27]*, Chen et al. [26]*, Hassanzadeh et al. [46]*
- -Aggregated measures
  - -Blagec et al. [18]*, Sogancioglu et al. [120, COM]*
  - -Chen et al. [27]*, Rychalska et al. [112]
  - -Al-Natsheh et al. [8], Farouk [39]
  - -Maharjan2017-ve [89], Nguyen et al. [99]
  - -Bounab et al. [20], Sultan et al. [121, 122]
- -Surveys on the topic
  - -Kalyan et al. [53]*, Khattak et al. [54]*, Alsentzer et al. [11]*

Figure 1.1: Categorization of the main sentence similarity methods reported in the literature. Citations with an asterisk (*) point out adaptations for the biomedical domain.

Words (CBOW) [92] models using PubMed information, whilst Zhang et al. [135] and Chen et al. [28] train FastText [19] models using PubMed and MeSH. Blagec et al. [18] introduce a set of neural embedding models based on the training of FastText [19], Sent2Vec [101], Paragraph vector [80], and Skip-thoughts vectors [57] models on the PMC dataset. Chen et al. [30] also introduce a sentence embedding model called BioSentVec, which is based on Sent2vec [101]. Likewise, we also find adaptations from several contextualized word representation models, also known as language models, for the biomedical domain. Tawfik and Spruit [125] evaluate a Flair-based [7] model trained on PubMed abstracts. Ranashinghe et al. [109], Peng et al. [104], Beltagy et al. [17] , Alsentzer et al. [11], Gu et al. [41] and Wada et al. [104, 129] introduce BERT-based models [33] trained on biomedical information. However, these latter models do not perform well in an unsupervised context because they are trained for downstream tasks using a supervised approach, which has encouraged Ranashinghe et al. [109] to explore a set of unsupervised approximations for evaluating BioBERT [106] and Embeddings for Language Models (ELMo) [106] models in the biomedical domain.

1.2.3 Biomedical semantic measures libraries

The main ontologies used for biomedical text mining and information retrieval applications in health sciences are SNOMED-CT and MeSH, although there are many other ontologies[3] based on the OBO file format [119]. By the time this thesis started, there were only two libraries for word-based semantic similarity based on the two aforementioned ontologies: (1) the pioneering Perl software library and online web interface called UMLS::Similarity [91], and (2) the most recent Java software library called SML [44], which introduces several significant contributions, such as portable and efficient object-oriented language programming, as well as a significant number of methods, and the implementation for the first time of the most significant biomedical ontologies and WordNet into a single software library. However, both UMLS::Similarity and SML have several significant performance and scalability drawbacks previously detailed in [77] which encourage our research and the extension of the HESML [75, 62] semantic measures library to the biomedical domain. In addition, there is only one library for sentence-based semantic measures libraries in the biomedical domain, named BIOSSES, which was introduced by Sogancioglu et al. [120], and uses UMLS::Similarity [91] for calculating the similarity of concepts. Thus, BIOSSES [120] inherits the drawbacks found in [77] by the use of the UMLS::Similarity [91] library, and also does not evaluate most of the measures based on embeddings and language models, such as those evaluated in this thesis [66].

[3] http://www.obofoundry.org

1.3 Structure of this thesis

This thesis is structured in three parts as follows. Part I is the main body of this thesis by compendium, whilst part II introduces the full-text of all of the publications derived from this thesis, and finally, part III introduces our software libraries, reproducibility protocol and datasets.

In turn, part I is structured as follows. Chapter 2 introduces a summary of the main motivations, research questions and hypotheses, research problems, and objectives tackled by each publication derived from this thesis. Chapter 3 details the theoretical foundations and domain knowledge of this thesis together with our research methodology. Chapter 4 introduces our main conclusions and forthcoming activities. Finally, chapter 5 enumerates our scientific contributions, both research articles, software libraries and datasets, whilst chapter 6 introduces a summary table detailing the quality metrics of our main publications.

Chapter 2

Hypotheses, research questions and Objectives

This chapter introduces a summary of the main motivation, hypotheses, research questions, research problems, and objectives tackled by each publication derived from this thesis. We have structured this chapter into three sections, one for each of our main publications. In turn, each section is divided into two other subsections detailing our main motivation and research questions or hypothesis, as well as the research problems and objectives tackled by each publication.

2.1 Protocol for a reproducible experimental survey on biomedical sentence similarity

The main aim of this publication is the introduction of a very detailed experimental setup for the development of the largest, and for the first time, reproducible experimental survey of methods on biomedical sentence similarity in order to elucidate the state of the problem. The research detailed in this section corresponds to the content introduced by Lara-Clares et al. [61].

2.1.1 Main motivations and research questions

The main motivation for the research detailed in this section is the lack of a reproducible experimental survey on biomedical sentence similarity, and the impossibility of reproducing most of the methods in this line of research. For instance, Sogancioglu et al. [120] provide neither the pre-trained models used in their experiments nor a detailed guide for replicating them, and their software artifacts do not reproduce all of their results. Blagec et al. [18] provide neither a detailed definition of their experimental setup nor their source code and pre-processed data, nor the pre-trained models used in their experiments. Chen et al. [30] based their assessment of the state of the art in biomedical sentence similarity exclusively on the results from Blagec et al. [18]; thus, their work allows neither previous results to be confirmed nor are they directly comparable with other works. In several cases, biomedical language models based on BERT, such as BioBERT [81] and NCBI-BlueBERT [104], can be reproduced neither in an unsupervised context nor in any other supervised way, because of the high computational requirements and the non-deterministic nature of the methods used for their training, respectively. Therefore, these two reproducibility gaps prevent the elucidation of the state of the art in a sound and reproducible way. For this reason, our research introduces a detailed protocol to implement a reproducible experimental survey that allows answering the following research questions:

RQ1 Which methods get the best results on biomedical sentence similarity?

RQ2 Is there a statistically significant difference between the best performing methods and the remaining ones?

RQ3 What is the impact of the biomedical Named Entity Recognition (NER) tools on the performance of the methods on biomedical sentence similarity?

RQ4 What is the impact of the pre-processing stage on the performance of the methods on biomedical sentence similarity?

RQ5 What are the main drawbacks and limitations of current methods on biomedical sentence similarity?

A second motivation is the implementation of a set of unexplored methods which are based on adaptations from other methods proposed for the general language domain.

A third motivation is the evaluation in the same software platform of the following benchmarks on biomedical sentence similarity reported in the literature: (1) the Biomedical Semantic Similarity Estimation System (BIOSSES) [120] dataset; (2) the Medical Semantic Textual Similarity (MedSTS) [130] dataset; and (3) the evaluation for the first time of the Microbial Transcriptional Regulation (CTR) [87] dataset in a sentence similarity task, despite it having been previously evaluated in other related tasks, such as the curation of gene expressions from scientific publications [86].

A fourth motivation is a study of the impact of the pre-processing stage and NER tools on the performance of the sentence similarity methods, such as that done by Gerlach et al. [40] for stop-words in the topic modeling task.

And finally, our fifth motivation is the lack of reproducibility software and data resources on this task, which allow an easy replication and confirmation of previous methods, experiments, and results in this line of research, as well as encouraging the development and evaluation of new sentence similarity methods.

2.1.2 Definition of the problem and objectives

The main research problem tackled in this work is the design and proposal of a comprehensive and reproducible experimental survey on sentence similarity measures for the biomedical domain. The main objectives of the research detailed herein are as follows:

1. To introduce the largest, and for the first time, a reproducible experimental survey on biomedical sentence similarity.

2. To introduce the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity.

3. To identify the existing gaps in the literature on biomedical sentence similarity.

4. To introduce a comprehensive and updated categorization of the literature on biomedical sentence similarity.

5. To check the reproducibility of previous methods reported in the literature, as well as pointing out the irreproducibility of others.

6. To propose the evaluation of a set of previously unexplored methods, as well as the proposal of a new word embedding model based on FastText and trained on the full-text of articles in the PMC-BioC corpus [31].

7. To integrate for the first time most of the sentence similarity methods for the biomedical domain into the same software library, called HESML-STS.

8. To introduce a detailed reproducibility protocol together with a collection of software tools and datasets, which will be provided as supplementary material to allow the exact replication of all our experiments and results.

2.2 A real-time semantic measures library for the biomedical domain

The main aim of this work is to introduce an updated and extended version of the HESML [69] semantic measures library for the biomedical domain, called HESML V1R5 [74], together with a fast approximation of Dijkstra's algorithm [35] for taxonomies based on a relaxed graph spanner, called Ancestors-based Shortest-Path Length (AncSPL), which allows for the first time the real-time computation of any path-based similarity measure on large ontologies, such as SNOMED-CT, GO, and WordNet. The research detailed in this section was first published by Lastra-Diaz et al. [77].

2.2.1 Main motivation and hypothesis

The main motivation of this work is to overcome some performance and scalability drawbacks in the current family of state-of-the-art semantic measure libraries for the biomedical domain, which are mainly used in the fields of biomedical text mining and genomics. Despite the UMLS::Similarity similarity software library having been extensively used in the literature, it has several significant drawbacks that prevent its use in high-throughput standalone applications, such as poor performance in the evaluation of measures, as well as a tedious, complex, and long setup process to build several pre-calculated data structures and values stored in an auxiliary database called UMLS::Interface. The drawbacks of UMLS::Similarity are mainly derived from its use of a scripting programming language like Perl and an ontology representation based on a relational database, which strongly impacts its performance and software architecture. More recently, Harispe et al. [44] introduced the SML Java software library, implementing for the first time the most significant ontologies in a single library, such as WordNet [93], SNOMED-CT [36], MeSH [1], the Gene Ontology [14, 127] and all others based on the OBO [119] and OWL file formats. However, SML has several significant performance and scalability drawbacks derived from the use of a naive in-memory graph representation based on hash tables and caching, which significantly impacts its overall performance, and very especially, its computation of path-based measures and scalability regarding the ontology size [69, §1.1.1]. To bridge the aforementioned drawbacks, Lastra-Diaz et al. [69] introduce the HESML Java software library based on WordNet, together with a very efficient and linearly scalable taxonomy representation called PosetHERep which allows this library to outperform SML by several orders of magnitude [69]. However, the field of biomedical research has not benefited yet from these aforementioned advances because previous HESML versions implement none of the most significant biomedical ontologies, such as SNOMED-CT, MeSH, GO and others based on the OBO file format. Our main hypothesis is that the efficient and scalable in-memory representation for ontologies implemented by HESML should solve these performance and scalability drawbacks, as detailed in hypothesis 1 below.

Hypothesis 1 (H1) *A HESML implementation of the main biomedical ontologies should significantly outperform the state-of-the-art biomedical semantic measures libraries in the evaluation of ontology-based semantic similarity measures, as previously shown for WordNet ontology [69].*

The second motivation of our work is to overcome a significant performance and scalability drawback of all path-based semantic similarity measures, which prevents their use in high-throughput experiments, or any practical application demanding their real-time computation. This problem is especially relevant because a recent reproducible survey on word similarity [71, 72, 70] shows that one hybrid IC-based similarity measure [68, coswJ&C] sets the state of the art in the family of ontology-based measures for the general domain. However, its practical use in any application is limited because of the lack of an efficient shortest-path algorithm for its real-time computation. Path-based similarity measures require an efficient implementation of a shortest-path algorithm, such as Dijkstra's algorithm [25]. However, its computational complexity prevents its practical use in high-throughput applications based on large ontologies like SNOMED-CT, GO, or WordNet. A common strategy followed by most of the software libraries and tools to tackle the aforementioned problem is to pre-calculate some auxiliary data structures, or all pairwise similarity scores, with the aim of speeding-up the subsequent evaluation of any path-based measure, as done by UMLS::Similarity, whilst other libraries like SML compute the path-based measures on-the-fly, and store the resulting similarity scores in a cache. The caching of auxiliary data structures and values demands large quantities of memory and complex setup processes, which does not solve the main practical problem in the real-time computation of path-based measures, and leads to poor performance, long setup processes, and running out of memory on large ontologies when they are used on average workstations. Our hypothesis with respect to the aforementioned problem of performance and scalability of path-based similarity measures is that a new approximated shortest-path algorithm, specifically designed for taxonomies, should overcome this problem, as detailed in hypothesis 2 below.

Hypothesis 2 (H2) *A new approximated shortest-path algorithm specifically designed for taxonomies could provide an efficient and linearly scalable method for reformulating and evaluating any path-based semantic similarity measure in real time, whereby the resulting similarity values would show a high-correlation value as regards the implementation of the measure using any exact shortest-path algorithm.*

And finally, a third motivation is to provide a larger and most updated set of ontology-based semantic similarity measures and Information Content (IC) models [67, 71] than those provided by the UMLS::Similarity and SML libraries.

2.2.2 Research problem and objectives

The two main research problems tackled in this work are as follows: (1) the development and release of an updated version of the HESML [69] library especially designed for the biomedical domain, called HESML V1R5 [74]; and (2) the proposal and evaluation of a new method for the approximation of Dijkstra's algorithm for taxonomies, called Ancestors-based Shortest-Path Length (AncSPL), which allows the real-time computation of any path-based semantic similarity measure.

The aim of this work is to introduce an updated version of the HESML [69] library especially designed for the biomedical domain, called HESML V1R5 [74], together with a fast approximation of Dijkstra's algorithm [35] for taxonomies based on a relaxed graph spanner called Ancestors-based Shortest-Path Length (AncSPL), which allows for the first time the real-time computation of any path-based similarity measure on large ontologies, such as SNOMED-CT, GO, and WordNet. HESML V1R5 implements most of the ontology-based similarity measures and IC models reported in the literature, as well as a very efficient and scalable in-memory representation of WordNet [93], SNOMED-CT, MeSH, GO [14], and other ontologies based on the OBO file format [119]. We introduce a set of reproducible benchmarks for testing our main hypothesis (H1) by comparing the performance of HESML with the UMLS::Similarity and SML libraries on the three most significant biomedical ontologies, as well as several experiments for testing our second hypothesis (H2) as regards the new AncSPL algorithm. Finally, we introduce a reproducibility dataset [76] together with a detailed reproducibility protocol, which is provided as supplementary material to allow the exact replication of all our experiments and results.

2.3 The reproducible experimental survey

The main aim of this work is to introduce a comprehensive and very detailed reproducible experimental survey of methods on biomedical sentence similarity to elucidate the state of the problem by implementing our previous registered report protocol [61] introduced in section 2.1.

The research detailed in this section was first published by Lara-Clares et al. [66]. In addition, the experiments detailed in this publication are based on our software implementation and evaluation of all methods analyzed herein in a common and new software platform based on an extension of HESML[1] [69, 77], which is called HESML for Semantic Textual Similarity (HESML-STS). In addition, all the experiments detailed in this publication have been recorded in a Docker virtualization image that is provided as supplementary material together with our software [64] and a detailed reproducibility protocol [63] and dataset [65] to allow the easy replication of all our methods, experiments, and results.

2.3.1 Main motivations and research questions

Our main motivation is the lack of a comprehensive and reproducible experimental survey on biomedical sentence similarity that would allow a sound and reproducible

[1] http://hesml.lsi.uned.es

survey of the state of the problem, as detailed in our previous registered report protocol [61]. Our main research questions are the same introduced in section 2.1.1, which we reproduce below for the sake of completeness:

RQ1 Which methods get the best results on biomedical sentence similarity?

RQ2 Is there a statistically significant difference between the best-performing methods and the remaining ones?

RQ3 What is the impact of the biomedical Named Entity Recognition (NER) tools on the performance of the methods on biomedical sentence similarity?

RQ4 What is the impact of the pre-processing stage on the performance of the methods on biomedical sentence similarity?

RQ5 What are the main drawbacks and limitations of current methods on biomedical sentence similarity?

A second motivation of this research is the implementation of a set of unexplored methods based on adaptations from other methods proposed for the general language domain.

A third motivation is the evaluation in the same software platform of the three known benchmarks on biomedical sentence similarity reported in the literature: (1) the Biomedical Semantic Similarity Estimation System (BIOSSES) [120] dataset; the Medical Semantic Textual Similarity (MedSTS) [130] dataset; and (3) the evaluation for the first time of the Microbial Transcriptional Regulation (CTR) [87] dataset in a sentence similarity task, despite it having been previously evaluated in other related tasks, such as the curation of gene expressions from scientific publications [86].

A fourth motivation is a study on the impact of the pre-processing stage and NER tools on the performance of the sentence similarity methods, such as that done by Gerlach et al. [40] for stop-words in the topic modeling task.

A fifth motivation is the proposal and evaluation of a new string-based sentence similarity method, based on Li et al. [84] and Block distance [59], eight variants of the current ontology-based methods from the literature based on the work of Sogancioglu et al. [120], and a new pre-trained Word Embedding (WE) model based on FastText [19] and trained on the full-text of articles in the PMC-BioC corpus [31].

And finally, our sixth motivation is the lack of reproducibility software and data resources for this task, which allow an easy replication and confirmation of previous methods, experiments, and results in this line of research, as well as encouraging the development and evaluation of new sentence similarity methods.

2.3.2 Definition of the problem and objectives

The two main research problems tackled in this work are the implementation of a comprehensive and reproducible experimental survey on sentence similarity measures for the biomedical domain, and the evaluation of a set of unexplored methods based on adaptations from previous methods used in the general language domain. The main objectives of the research detailed herein are as follows:

1. To introduce the largest, and for the first time, a reproducible experimental survey on biomedical sentence similarity.

2. To introduce the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity.

3. To evaluate several unexplored sentence similarity methods.

4. To propose and evaluate a new string-based sentence similarity method, based on Li et al. [84] and Block distance [59], eight variants of the current ontology-based methods from the literature based on the work of Sogancioglu et al. [120], and a new pre-trained Word Embedding (WE) model based on FastText [19] and trained on the full-text of articles in the PMC-BioC corpus [31].

5. To evaluate for the first time an unexplored benchmark, called Corpus-Transcriptional-Regulation (CTR) [87].

6. To carry out a study on the impact of the pre-processing stages and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods.

7. To bridge the lack of software and data reproducibility resources for methods and experiments in this line of research, integrating for the first time most sentence similarity methods for the biomedical domain into the same software library, called HESML-STS, which is available both on Github [2] and in a reproducible dataset [65], together with a detailed reproducibility protocol together with a collection of software tools and datasets provided as supplementary material to allow the exact replication of all our experiments and results.

8. To elucidate the state of the art of the problem, with an analysis of the drawbacks and limitations of the current state-of-the-art methods.

In addition, our reproducible experimental survey is based on a single software platform called HESML-STS, which is available on Github [3], and is provided with a detailed reproducibility protocol and dataset [65] as supplementary material to allow the exact replication of all our experiments and results.

[2] https://github.com/jjlastra/HESML

[3] https://github.com/jjlastra/HESML/tree/HESML-STS_master_dev

Chapter 3

Theoretical Foundations and Methodology

This chapter briefly introduces the theoretical foundations of this thesis, which are defined by the theories of Cognitive Lexical Semantics and Semantic Compositionality. These two theories are closely related to the main conclusions of this thesis, highlighting the capabilities and limitations of different families of similarity measures depending on their way of extracting meaning from sentences.

The rest of the chapter is structured as follows: section 3.1 details the theoretical foundations and closely related lines of research of this thesis, whilst section 3.2 introduces our research methodology.

3.1 Theoretical foundations

The aim of any semantic similarity measure is to estimate the degree of similarity between two textual semantic units as perceived by a human being, such as words, phrases, sentences, short texts, or documents. The notion of semantic similarity can be studied at different linguistic levels such as: morphologic, syntactic, semantic, pragmatic, and discursive. Compositional semantics is devoted to the study of the building up of phrasal or sentence meaning from the meaning of smaller units, a problem extensively studied in the field of cognitive semantics. Taylor et al. [126] points out that the knowledge of a language can be partitioned into two main components: (1) the knowledge of the lexicon and (2) the knowledge of the syntax. The lexicon lists the words together with their meaning and syntactic category, whilst the syntax comprises the rules for combining elements belonging to certain syntactic categories into larger configurations, such as sentences. Thus, the meaning of a sentence depends not only on the meaning of the words that constitute it but also on how they are combined, i.e., the grammatical structure. However, this traditional point of view assumes that words are typically conceived as static lexical entries.

New advances in Cognitive Lexical Semantics focus on issues such as polysemy, semantic changes and meaning extensions, and extending the lexicon to an encyclopedic knowledge network. In this context, Evans [38] proposes the Theory of Lexical Concepts and Cognitive Models and holds that meaning is not a property of words but rather of a function of context. Evans integrates the Principle of Semantic

Compositionality from Pelletier [103], which states that the meaning of a complex expression (for example, a sentence) is a function of the meaning of its parts together with how those parts are combined. Thus, the aforementioned theories on compositional semantics suggest that given two sentences, their semantic similarity should be computed by dividing them into semantic and syntactic units and composing the semantic similarity between these units as mentioned above.

3.2 Research methodology

Our research methodology is based on the methodology followed in [78]. Firstly, we developed an extensive reproducible protocol [61] with a detailed description of the methodology, organization and schedule of this thesis. In summary, it is defined by the workflow shown in figure 3.1 and detailed in steps 1 to 14 below:

1. Definition of our main research problem.

2. Comprehensive review of the literature on the problem studied as well as other related problems and applications.

   Our literature review consisted of the following steps: (1) formulation of our research questions; (2) search of relevant publications on biomedical sentence similarity, especially all methods and works whose experimental evaluation is based on the sentence similarity benchmarks considered in our experimental setup; (3) definition of inclusion and exclusion criteria of the methods; (4) definition of the study limitations and risks; and (5) definition of the evaluation metrics. Publications on our research topic were mainly searched for in the Web Of Science (WOS) and Google Scholar databases, and the SemEval [5, 6, 3, 2, 4, 24] and BioCreative/OHNLP [131] conference series. In order to build a first set of relevant works on the topic, we selected a seed set of highlighted publications and datasets on biomedical sentence similarity [120, 84, 130, 18, 30, 81] from the aforementioned information sources. Then, we reviewed all the papers related to sentence similarity which cited any seed publication or dataset. Finally, starting from seed publications and datasets, we extracted those methods that could be implemented and evaluated in our experiments, and we downloaded and checked all the available pre-trained models. Our main goal was to attempt an independent replication or evaluation of all methods previously evaluated on the biomedical sentence similarity benchmarks considered in our experiments.

3. Synthesis and categorization of the literature based on features such as: strategy and tactics used, functional structure, application domain, specific problem or motivation, experimental setup, definition of the contingency plan and schedule, etc.

4. Identification of the gap to be bridged, such as: drawbacks, inconsistencies in the formulation of the models and methods, underlying assumptions, unexplored notions and strategies, formulation of novel hypotheses or research questions, refutation of previous conclusions, and studying the problem from a novel point of view and from different disciplines.

5. Selection of methods to be evaluated, by defining the following selection criteria: (a) identification of all the methods in the biomedical domain that were evaluated in the BIOSSES [120] and MedSTS [130] datasets; (b) identification of those methods reported for the general domain, but not yet evaluated in the biomedical domain; and (c) definition of the criteria for the selection and exclusion of methods. In addition, our selection criteria for the sentence similarity methods to be reproduced and evaluated herein have been significantly conditioned by the availability of multiple sources of information, namely: (1) pre-trained models; (2) source code; (3) reproducibility data; (4) detailed descriptions of the methods and experiments; (5) reproducibility guidelines; and finally, (6) the computational requirements for training several models. Thus, this thesis reproduces and evaluates most of the sentence similarity methods for the biomedical domain reported in the literature, as well as other methods that have not been explored in this domain yet.

6. Selection of the pre-processing methods evaluated in this work to ensure a fair comparison of the methods that are evaluated in a single end-to-end pipeline. The selection criteria of the pre-processing components have been conditioned by the following constraints: (a) the pre-processing methods and tools used by state-of-the-art methods; and (b) the availability of resources and software tools.

7. Proposal of novel methods and adaptations to bridge the previously identified gaps. Correlation and generation of ideas based on analogies and personal intuitions. Inquiry into related ideas in other fields of research, disciplines and related problems.

8. Designing or replication of experiments to evaluate our novel hypotheses and proposals.

9. Implementation of the experiments to evaluate our methods and hypotheses.

10. Replication and reproduction of related methods with the aim of comparing our results with the state of the art.

11. Verification and contrasting of our results, as well as the results obtained in the replication of other methods and those reported in the literature. Personal communication with the authors whenever it is necessary to clarify any issue regarding the precise replication and reproduction of their methods and results.

12. Critical discussion of the results and their consequences. Contrast of our results as regards previous methods and results reported in the literature. Confirmation and refutation of previous methods and results reported in the literature based on our own experimentation.

13. Identification of drawbacks and limitations in our novel proposals.

14. Formulation of new hypotheses and forthcoming activities. Identification of potential applications of our methods and results in other related problems or fields of application.

15. Publication and dissemination of our results.

16. Selection of a new research problem from our backlog of new hypotheses, ideas and forthcoming activities and start of a new iteration of our research methodology.

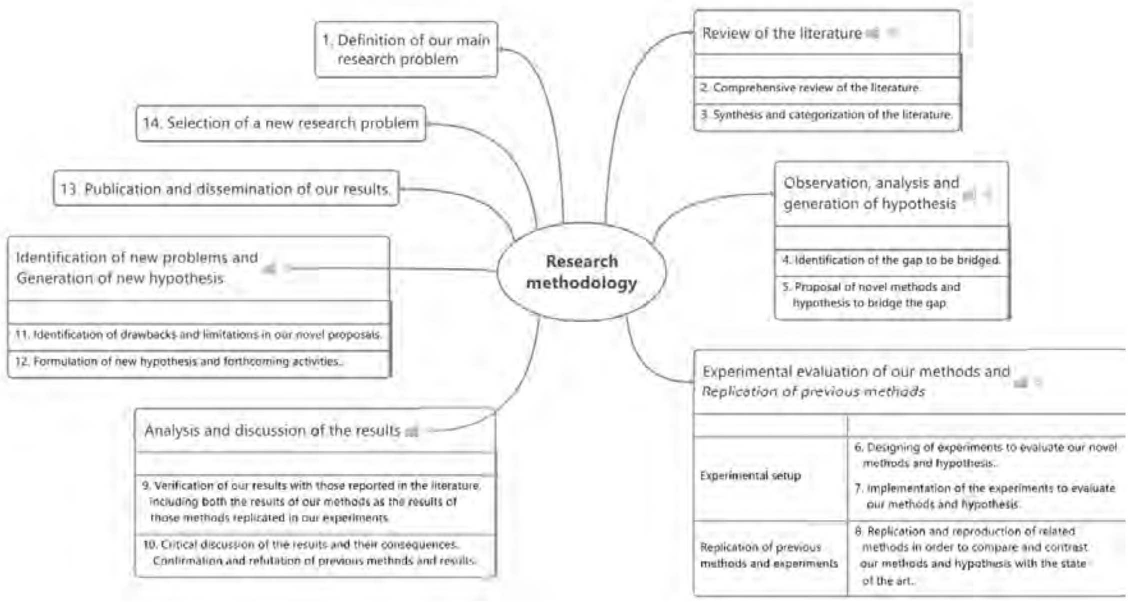

Research methodology
1. Definition of our main research problem
Review of the literature
2. Comprehensive review of the literature.
3. Synthesis and categorization of the literature.
14. Selection of a new research problem
13. Publication and dissemination of our results.
Observation, analysis and generation of hypothesis
4. Identification of the gap to be bridged.
5. Proposal of novel methods and hypothesis to bridge the gap.
Identification of new problems and Generation of new hypothesis
11. Identification of drawbacks and limitations in our novel proposals.
12. Formulation of new hypothesis and forthcoming activities.
Experimental evaluation of our methods and *Replication of previous methods*
Experimental setup
6. Designing of experiments to evaluate our novel methods and hypothesis.
7. Implementation of the experiments to evaluate our methods and hypothesis.
Replication of previous methods and experiments
8. Replication and reproduction of related methods in order to compare and contrast our methods and hypothesis with the state of the art.
Analysis and discussion of the results
9. Verification of our results with those reported in the literature, including both the results of our methods as the results of those methods replicated in our experiments.
10. Critical discussion of the results and their consequences. Confirmation and refutation of previous methods and results.

Figure 3.1: Research methodology adopted in this thesis, which is based on the methodology proposed by [78]. We place a special emphasis in the replications of previous methods and results, as well as their confirmation and refutation.

Chapter 4

Conclusions and Future Work

This chapter introduces a summary of the main conclusions derived from the research carried out in this thesis.

4.1 Main conclusions

The main conclusions drawn from the research introduced from our research (Lara-Clares et al. [61]) introducing a protocol for a reproducible experimental survey on biomedical sentence similarity are detailed below.

1.1 We introduce a detailed experimental setup to reproduce, evaluate, and compare the most comprehensive set of methods on biomedical sentence similarity reported in the literature, as detailed in [61, pp. 7-20]. Our experimental setup proposed includes a selection of sentence similarity methods, language pre-processing methods and tools [61, pp. 7-10], a detailed software integration and contingency plan [61, pp. 11-13], a detailed workflow of our experiments [61, pp. 13-14], and a detailed description of the evaluation metrics [61, pp. 14-15].

1.2 We introduce a comprehensive and updated categorization of the literature on sentence semantic similarity measures for the biomedical language domain, as shown in [61, fig. 1], and detailed in [61, pp. 4-7].

1.3 We propose a protocol and development plan to build the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity based on the same software platform, as detailed in [61, pp. 15-20].

1.4 We propose for the first time the evaluation of the CTR [87] dataset, as shown in [61, table 5].

1.5 We propose the evaluation of most biomedical sentence similarity methods, as well as a set of new sentence similarity methods adapted from their definitions in the general-language domain, as detailed in [61, tables 1-4].

1.6 We propose the evaluation of a new word embedding model based on FastText and trained on the full text of the articles in the PMC-BioC corpus [31], as shown in [61, table 3].

1.7 We propose the study of the impact of different pre-processing configurations on the performance of the sentence similarity methods, as detailed in [61, pp. 10-11, figure 6].

1.8 We propose the study of the impact of different Name Entity Recognition (NER) tools, such as MetaMap [12] and clinic Text Analysis and Knowledge Extraction System (cTAKES) [115], on the performance of the sentence similarity methods, as detailed in [61, pp. 10-11, figure 6].

1.9 We propose a detailed statistical significance analysis of the results, as detailed in [61, pp. 14-15].

1.10 We point out the existence of several reproducibility problems in the replication of some methods and experimental results previously reported in the literature, as detailed in [61, pp. 1-3].

The main conclusions drawn from the research introduced from our research (Lastra-Díaz et al. [77]) introducing a real-time semantic measures library for the biomedical domain with a reproducible survey are as follows:

2.1 We introduce a new semantic measures library for the biomedical domain called HESML V1R5, which implements the largest set of ontology-based semantic similarity measures and IC models for the SNOMED-CT, MeSH, GO [14, 127], and OBO file format ontologies [119], in addition to WordNet [93], as detailed in [77, pp. 9-12].

2.2 We propose and evaluate a new approximated shortest-path algorithm called AncSPL which provides a real-time and highly-correlated reformulation of any path-based semantic similarity measure, as detailed in [77, pp. 12-17].

2.3 We introduce a set of reproducible benchmarks for testing our main hypothesis (H1) by comparing the performance of HESML with the UMLS::Similarity and SML libraries on the three most significant biomedical ontologies, as detailed in [77, pp. 19, tables 6-9].

2.4 In addition, we introduce several experiments for testing our second hypothesis (H2) as regards the new AncSPL algorithm, as detailed in [77, pp. 19, tables 10-12].

2.5 We confirm that HESML outperforms by four orders of magnitude the implementation of the Rada et al. [108] path-based measure of UMLS::Similarity in the MeSH ontology as shown in [77, tables 7-9]. However, the UMLS::Similarity implementation of the Rada et al. [108] measure based on caching is roughly three times faster than the HESML real-time implementation in the large SNOMED-CT ontology, as shown in [77, table 6].

2.6 HESML outperforms by six and three orders of magnitude, respectively, the implementation of the Lin [85] IC-based measure of UMLS::Similarity in the SNOMED-CT and MeSH ontologies, as shown in [77, tables 6, 7, 9].

2.7 HESML outperforms by seven and four orders of magnitude, respectively, the implementation of the depth-based approximation of the Wu and Palmer [134] measure of UMLS::Similarity in the SNOMED-CT and MeSH ontologies, as shown in [77, tables 6, 7, 9].

2.8 HESML outperforms by six, two, and four orders of magnitude the implementation of the Rada et al. [108] path-based measure of SML in the MeSH and GO ontologies as shown in [77, tables 7-9]. In addition, SML is unable to provide a practical implementation of the Rada et al. [108] measure on the large SNOMED-CT ontology, as shown in [77, table 6].

2.9 The HESML implementation of the Lin [85] IC-based measure is roughly 2.43 times faster than the implementation of SML based on SNOMED-CT as shown in [77, table 6], as well as roughly 1.55 times faster on MeSH as shown in [77, tables 7,9], and roughly 2.86 times faster on GO as shown in [77, table 8].

2.10 We positively confirm our hypothesis H1 which states that HESML significantly outperforms current state-of-the-art semantic measures libraries in the real-time evaluation of semantic similarity measures.

2.11 Path-based measures based on the new AncSPL algorithm are six and five orders of magnitude, respectively, faster than their exact implementation in the large ontologies with multiple inheritance, SNOMED-CT and GO, as shown in [77, tables 6,8], whilst AncSPL obtains similar performance to the exact implementation on tree-like ontologies like MeSH, as shown in [77, tables 7,9], because both implementations are identical by definition.

2.12 The results reported in [77, table 10] show that the reformulation of any path-based measure using AncSPL is highly correlated both in Pearson and Spearman correlation metrics with their corresponding exact implementations. Thus, this conclusion endorses the reformulation of any path-based similarity measure using AncSPL to obtain real-time approximations of any path-based measure on large ontologies with multiple inheritance, such as SNOMED-CT, GO, or WordNet.

2.13 Groupwise similarity measures based on GO implemented by HESML provide a high average speed in the evaluation of the pairwise protein similarity between two large organisms in a large-scale experiment, as shown in [77, table 11]. Thus, HESML can significantly contribute to improving the performance of any application using GO-based semantic similarity measures. Likewise, HESML opens the possibility of processing large-scale GO annotated data at high computation rates, which could encourage new applications like the similarity-based search of proteins in large GO-annotated databases, among others.

2.14 The shortest-path length estimated by AncSPL is always greater or equal to the exact value, as shown in [77, figure 2] by the empirical Cumulative Distribution Function (CDF) for SNOMED-CT, GO, and WordNet ontologies, respectively. The signed length error of AncSPL is 0 with a probability of

0.479, 0.581, and 0.612, on SNOMED-CT, GO, and WordNet, respectively. Thus, the AncSPL-based reformulations of any path-based similarity measure on non-tree-like ontologies always return a lower or equal value than their corresponding base measures evaluated using an exact shortest-path algorithm.

2.15 The signed length error of AncSPL is lower than or equal to 2 with a probability of 0.874, 0.898, and 0.8841, on SNOMED-CT, GO, and WordNet, respectively, as shown in [77, figure 2]. Thus, AncSPL-based reformulations obtain close results to the exact shortest-path algorithm with a high probability.

2.15 The signed length error of AncSPL decreases with the tree-like deviation, as shown in [77, figure 2]. This means that, the lower the number of concepts with multiple parents, the higher is the probability of obtaining an AncSPL length error equal to 0. However, looking at the correlation values reported in [77, table 10], we can observe that correlation values obtained by the AncSPL-based reformulations in WordNet are not significantly higher than the values obtained in SNOMED-CT and GO as would be expected. We conjecture that AncSPL-coswJ&C is more immune to the AncSPL approximation error than the edge-counting measures because it is defined by the length of the IC-based weighted shortest path between concepts.

2.16 The average running time of the AncSPL algorithm is linear regarding the dimension of the ancestor-based subgraph, as predicted by [77, theorem 1] and shown experimentally in [77, fig. 3] for SNOMED-CT, GO, and WordNet ontologies, respectively.

2.17 We confirm that the significant performance gain shown in [77, tables 6-9], together with the high-correlation values shown in [77, table 10], confirm positively our hypothesis H2 on the performance, scalability, and approximation quality of the new AncSPL algorithm.

2.18 We introduce a reproducibility dataset [76] together with a detailed reproducibility protocol, which is provided as supplementary material (see additional files) to allow the exact replication of all our experiments and results.

The main conclusions drawn from the research introduced from our research (Lara-Clares et al. [66]) introducing the experimental reproducible protocol of biomedical sentence similarity are detailed above. Because of the high number of factual conclusions, we have chosen the most important conclusions, and we refer the reader to the Discussion section [66, pp. 26-37].

3.1 We introduce the largest, detailed, and for the first time, reproducible experimental survey on biomedical sentence similarity reported in the literature.

3.2 We introduce a collection of self-contained and reproducible benchmarks on biomedical sentence similarity based on the same software platform, called HESML-STS, which has been especially developed for this work, being provided as part of the new HESML V2R1, as detailed in [66, page 9].

3.3 We introduce a new aggregated string-based sentence similarity method called LiBlock, together with eight variants of the ontology-based methods introduced by Sogancioglu et al. [120], and a new pre-trained word embedding model based on FastText [19] and trained on the full-text of the articles in the PMC-BioC corpus [31], as detailed in [66, pp. 4-8].

3.4 We evaluate for the first time the CTR [87] dataset in a benchmark on biomedical sentence similarity [66, page 8, table 1].

3.5 Concerning our RQ1 and RQ2 research questions, the string-based method LiBlock (M4) obtains the highest average harmonic score in all datasets, and significantly outperforms the remaining string-based methods, as well as all methods based on embeddings and BERT language models, and all the ontology-based methods with the only exceptions of COM (M17) and WBSM-Rada (M7), as detailed in [66, table 8].

3.6 Our LiBlock sentence similarity measure obtains the highest Spearman correlation values in the BIOSSES and MedSTS datasets, which contains 100 and 1068 sentence pairs respectively, as detailed in [66, table 8].

3.7 Concerning our RQ3 research question, our results show that the ontology-based methods obtain their best performance in the task of biomedical sentence similarity when they use either MetamapLite or cTAKES, as detailed in [66, tables 10,11]. Thus, Metamap should not be used in combination with any of the ontology-based methods evaluated herein in this task.

3.8 The results and p-values reported in [66, table 11] show that there is a significant difference in the performance of each ontology-based method according to the NER tool used in most cases. The conclusions above confirm that the selection of the NER tool significantly impacts the performance of the sentence similarity methods using it.

3.9 Our experiments show that there is no statistically significant difference between the performance of the LiBlock (M4) method using the cTakes or no NER tool, as detailed in [66, table 12]. Thus, using the LiBlock method without any NER tool could be a competitive and much more efficient solution for high-throughput applications.

3.10 Concerning our RQ4 research question, our results and the conclusions above show that the pre-processing configurations significantly impact the performance of the sentence similarity methods. Thus, it should be specifically defined for each method, as shown in [66, table 9].

3.11 Concerning our RQ5 research question, string-based methods capture neither the word semantics within the sentences nor the semantic relationships between words, such as synonymy and meronymy, and their effectiveness mainly relies on the word overlapping frequency in the sentences.

3.12 Ontology-based methods use NER and WSD tools to recognize the underlying concepts in the sentences, but are not able to correctly identify and disambiguate these concepts in many cases, as detailed in [66, tables 13,14]. In addition, they require external resources to capture the semantic information from the sentences, which limits their lexical coverage. Thus, ontology-based methods require both high word overlapping and high recognition coverage of named entities to properly estimate the similarity between sentences.

3.13 The methods based on pre-trained embeddings and language models need a large corpus for training, a complex training phase, and considerable computational resources to calculate the similarity between sentences. Moreover, those methods tend to obtain high similarity scores in most cases, as detailed in [66, tables 15,16,17], which may penalize them in a balanced dataset and in a real environment.

3.14 BERT-based methods are trained for downstream tasks, using a supervised approach, and do not perform well in an unsupervised context, as detailed in [66, table 8].

3.15 LiBlock method uses the NER tool to normalize and disambiguate the underlying concepts in a sentence, but unfortunately, it does not significantly outperform LiBlock with no use of a NER tool, as shown in [66, table 12]. We conjecture that this behaviour could be due to two reasons. Firstly, the incapability of LiBlock to capture semantic relationships beyond the synonymy, and secondly the current limitations of cTakes to recognize all mentions of biomedical entities.

3.16 We provide a detailed reproducibility protocol [63] and dataset [65] to allow the exact replication of all our experiments, methods, and results.

Finally, table 4.1 shows a summary of the confirmation of the main hypotheses and research questions studied by this thesis.

Id	Hypothesis	Results
H1	A HESML implementation of the main biomedical ontologies should significantly outperform the state-of-the-art biomedical semantic measures libraries in the evaluation of ontology-based semantic similarity measures, such as previously shown for WordNet ontology [69].	Positively confirmed
H2	A new approximated shortest-path algorithm specifically designed for taxonomies could provide an efficient and linearly scalable method for reformulating and evaluating any path-based semantic similarity measure in real time, whereby the resulting similarity values would show a high-correlation value as regards its implementation using any exact shortest-path algorithm.	Positively confirmed
RQ1	Which methods get the best results on biomedical sentence similarity?	Answered
RQ2	Is there a statistically significant difference between the best-performing methods and the remaining ones?	Answered
RQ3	What is the impact of the biomedical Named Entity Recognition (NER) tools on the performance of the methods on biomedical sentence similarity?	Answered
RQ4	What is the impact of the pre-processing stage on the performance of the methods on biomedical sentence similarity?	Answered
RQ5	What are the main drawbacks and limitations of current methods on biomedical sentence similarity?	Answered

Table 4.1: Results obtained for the main hypotheses and research questions studied by this thesis.

4.2 Future work

As forthcoming activities, we plan to continue our work in four complementary directions:

1. Evaluation of the sentence similarity methods implemented in HESML in an extrinsic task, such as semantic medical indexing [32] or summarization [96].

2. Extending the pre-processing configurations, such as biomedical NER systems based on recent Deep Learning techniques [42], or extending our experiments and research to the multilingual scenario by integrating multilingual biomedical NER systems like Cimind [23].

3. Extending the scope of the HESML library, developing new Python or R interfaces, since the research community currently tends to focus its developments on these languages.

4. Evaluating the new sentence similarity methods introduced in [66] in a benchmark for the general language domain.

Chapter 5

Scientific contributions

This chapter sets out all contributions derived directly from this thesis, which are divided into five types as follows: (1) peer-reviewed articles, (2) articles published in workshops, (3) protocol, (4) software libraries and (5) replication datasets and benchmarks.

5.1 Peer-reviewed articles

1. Lara-Clares A., Lastra-Díaz J. J., Garcia-Serrano A. (2021). Protocol for a reproducible experimental survey on biomedical sentence similarity. PLOS ONE, 16, 1-28.
   https://doi.org/10.1371/journal.pone.0248663

2. Lastra-Díaz, J.J., Lara-Clares, A., Garcia-Serrano, A. (2022). HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey. BMC Bioinformatics 23, 23-54.
   https://doi.org/10.1186/s12859-021-04539-0

3. Lara-Clares A., Lastra-Díaz J. J. , Garcia-Serrano A. (2022). A reproducible experimental survey on biomedical sentence similarity: A string-based method sets the state of the art. PLOS ONE, 17, 1-44.
   https://doi.org/10.1371/journal.pone.0276539

4. (Co-authored as reproducibility reviewer) Mandilaras G, Papadakis G, Gagliardelli L, Simonini G, Thanos E, Giannakopoulos G, Bergamaschi S., Palpanas T., Koubarakis M., Lara-Clares A., Fariña A. (2021). Reproducible experiments on Three-Dimensional Entity Resolution with JedAI. Information Systems, 102, 101830.
   https://doi:10.1016/j.is.2021.101830

5.2 Workshops

1. Lara-Clares A., Garcia-Serrano A. (2019) Key Phrases Annotation in Medical Documents: MEDDOCAN 2019 Anonymization Task. In IberLEF@ SEPLN

(pp. 755-760).
http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_15.pdf

2. Lara-Clares, A., García-Serrano, A. (2019). LSI2_UNED at eHealth-KD Challenge 2019: A Few-shot Learning Model for Knowledge Discovery from eHealth Documents. In IberLEF@ SEPLN (pp. 60-66).
https://ceur-ws.org/Vol-2421/eHealth-KD_paper_6.pdf

3. Lara-Clares, A., Garcia-Serrano, A. (2020). Statistical Graph Matching for Indexing Spanish Biomedical Documents. In CLEF Working Notes (pp. 60-66).
http://ceur-ws.org/Vol-2696/paper_80.pdf

5.3 Protocols

1. Lara-Clares A., Lastra-Díaz J. J., Garcia-Serrano A. (2022). A reproducibility protocol and dataset on the biomedical sentence similarity. protocols.io. Version created by Alicia Lara Clares
https://dx.doi.org/10.17504/protocols.io.36wgq429xvk5/v4

5.4 Software libraries

1. Lastra-Díaz J. J., Lara-Clares A., Garcia-Serrano A. (2020) HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models. e-cienciaDatos.
https://doi:10.21950/1RRAWJ

2. Lara-Clares, A.; Lastra-Díaz, J. J.; Garcia-Serrano, A. (2022) HESML V2R1 Java software library of semantic similarity measures for the biomedical domain, e-cienciaDatos, V2
https://doi.org/10.21950/AQLSMV

5.5 Replication datasets and benchmarks

1. Lastra-Díaz, J. J., Lara-Clares, A., Garcia-Serrano, A.. (2020) Reproducibility dataset for a benchmark of biomedical semantic measures libraries, e-cienciaDatos, V5
https://doi.org/10.21950/OTDA4Z

2. Lara-Clares, A., Lastra-Díaz, J. J., Garcia-Serrano, A. (2021) Reproducible experiments on word and sentence similarity measures for the biomedical domain, e-cienciaDatos, V2
https://doi.org/10.21950/EPNXTR

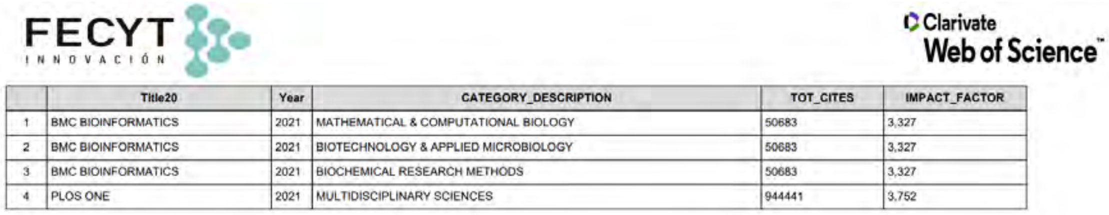
Chapter 6

Impact factor of the publications

Table 6.1 shows the JCR quartile and Impact Factor (IF) of our three main publications corresponding to the JCR-2021 ranking, as shown in figure 6.1.

Reference	Journal	IF-2021	Quartile
Lara-Clares et al. [61]	Plos One	3.752	Q2
Lastra-Díaz et al. [77]	BMC Bioinformatics	3.328	Q2
Lara-Clares et al. [66]	Plos One	3.752	Q2
	Overall Impact Factor	10.842	

Table 6.1: JCR impact factors of the three main publications derived from this thesis.



	Title20	Year	CATEGORY_DESCRIPTION	TOT_CITES	IMPACT_FACTOR
1	BMC BIOINFORMATICS	2021	MATHEMATICAL & COMPUTATIONAL BIOLOGY	50683	3,327
2	BMC BIOINFORMATICS	2021	BIOTECHNOLOGY & APPLIED MICROBIOLOGY	50683	3,327
3	BMC BIOINFORMATICS	2021	BIOCHEMICAL RESEARCH METHODS	50683	3,327
4	PLOS ONE	2021	MULTIDISCIPLINARY SCIENCES	944441	3,752

Figure 6.1: JCR-2021 Impact Factor of our two main publications (source: WoS-FECYT)

Bibliography

[1] Abdeddaïm, S., Vimard, S., Soualmia, L.F., 2019. The mesh-gram neural network model: Extending word embedding vectors with mesh concepts for semantic similarity, in: Ohno-Machado, L., Séroussi, B. (Eds.), MEDINFO 2019: Health and Wellbeing e-Networks for All - Proceedings of the 17th World Congress on Medical and Health Informatics, IOS Press. pp. 5–9. doi:10.3233/SHTI190172.

[2] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Others, 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability, in: Proc. of the 9th international workshop on semantic evaluation (SemEval 2015), ACL. pp. 252–263.

[3] Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J., 2014. Semeval-2014 task 10: Multilingual semantic textual similarity, in: Proc. of the 8th international workshop on semantic evaluation (SemEval 2014), ACL. pp. 81–91.

[4] Agirre, E., Banea, C., Cer, D., Diab, M., others, 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. 10th International Workshop on Semantic Evaluation (SemEval-2016) .

[5] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., 2012. Semeval-2012 task 6: A pilot on semantic textual similarity, in: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proc. of the main conference and the shared task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), ACL. pp. 385–393.

[6] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., 2013. * SEM 2013 shared task: Semantic textual similarity, in: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proc. of the Main Conference and the Shared Task: Semantic Textual Similarity, ACL. pp. 32–43.

[7] Akbik, A., Blythe, D., Vollgraf, R., 2018. Contextual string embeddings for sequence labeling, in: Proc. of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA. pp. 1638–1649.

[8] Al-Natsheh, H.T., Martinet, L., Muhlenbach, F., Zighed, D.A., 2017. UdL at SemEval-2017 task 1: Semantic textual similarity estimation of English sentence pairs using regression model over pairwise features, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada. pp. 115–119. doi:10.18653/v1/S17-2013.

[9] Aliguliyev, R.M., 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Syst. Appl. 36, 7764–7772.

[10] Allot, A., Chen, Q., Kim, S., Vera Alvarez, R., Comeau, D.C., Wilbur, W.J., Lu, Z., 2019. LitSense: making sense of biomedical literature at sentence level. Nucleic Acids Res. .

[11] Alsentzer, E., Murphy, J., Boag, W., Weng, W.H., Jindi, D., Naumann, T., McDermott, M., 2019. Publicly available clinical BERT embeddings, in: Proc. of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, USA. pp. 72–78. doi:10.18653/v1/W19-1909.

[12] Aronson, A.R., Lang, F.M., 2010. An overview of MetaMap: historical perspective and recent advances. J. Am. Med. Inform. Assoc. 17, 229–236.

[13] Arora, S., Liang, Y., Ma, T., 2017. A simple but tough-to-beat baseline for sentence embeddings, in: International Conference on Learning Representations, pp. 1–16.

[14] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Michael Cherry, J., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G., 2000. Gene Ontology: tool for the unification of biology. Nature Genetics 25, 25–29.

[15] Babić, K., Martinčić-Ipšić, S., Meštrović, A., 2020. Survey of Neural Text Representation Models. Information. An International Interdisciplinary Journal 11, 511.

[16] Bär, D., Biemann, C., Gurevych, I., Zesch, T., 2012. UKP: Computing semantic textual similarity by combining multiple content similarity measures, in: Proc. of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proc. of the Main Conference and the Shared Task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation, Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 435–440.

[17] Beltagy, I., Lo, K., Cohan, A., 2019. SciBERT: A pretrained language model for scientific text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China. pp. 3615–3620. doi:10.18653/v1/D19-1371.

[18] Blagec, K., Xu, H., Agibetov, A., Samwald, M., 2019. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. BMC Bioinformatics 20, 178.

[19] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146.

[20] Bounab, Y., Seppnen, J., Savusalo, M., Mkynen, R., Oussalah, M., 2019. Sentence to sentence similarity. a review, in: Conference of Open Innovations Association, FRUCT, elibrary.ru. pp. 439–443.

[21] Boyack, K.W., Newman, D., Duhon, R.J., Klavans, R., Patek, M., Biberstine, J.R., Schijvenaars, B., Skupin, A., Ma, N., Börner, K., 2011. Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. PLoS One 6, e18029.

[22] Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

[23] Cabot, C., Darmoni, S., Soualmia, L.F., 2019. Cimind: A phonetic-based tool for multilingual named entity recognition in biomedical texts. J. Biomed. Inform. 94, 103176.

[24] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L., 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada. pp. 1–14. doi:10.18653/v1/S17-2001.

[25] Chen, M., Chowdhury, R.A., Ramachandran, V., Roche, D.L., Tong, L., 2007. Priority queues and Dijkstra's algorithm. Technical Report TR-07-54. Computer Science Department, University of Texas at Austin.

[26] Chen, Q., Du, J., Kim, S., Wilbur, W.J., Lu, Z., 2018. Combining rich features and deep learning for finding similar sentences in electronic medical records. Proceedings of the BioCreative/OHNLP Challenge , 5–8.

[27] Chen, Q., Du, J., Kim, S., Wilbur, W.J., Lu, Z., 2020a. Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records. BMC Medical Informatics and Decision Making 20, 73. doi:https://doi.org/10.1186/s12911-020-1044-0.

[28] Chen, Q., Lee, K., Yan, S., Kim, S., Wei, C.H., Lu, Z., 2020b. Bioconceptvec: Creating and evaluating literature-based biomedical concept embeddings on a large scale. PLOS Computational Biology 16, 1–18. URL: https://doi.org/10.1371/journal.pcbi.1007617, doi:10.1371/journal.pcbi.1007617.

[29] Chen, Q., Panyam, N.C., Elangovan, A., Davis, M., Verspoor, K., 2017. Document triage and relation extraction for protein-protein interactions affected by mutations, in: Proc. of the BioCreative VI Workshop, pp. 52–51.

[30] Chen, Q., Peng, Y., Lu, Z., 2019. Biosentvec: creating sentence embeddings for biomedical texts, in: 2019 IEEE International Conference on Healthcare Informatics (ICHI), IEEE. pp. 1–5.

[31] Comeau, D.C., Wei, C.H., Islamaj Doğan, R., Lu, Z., 2019. PMC text mining subset in BioC: about three million full-text articles and growing. Bioinformatics .

[32] Couto, F.M., Krallinger, M., 2020. Proposal of the first international workshop on semantic indexing and information retrieval for health from heterogeneous content types and languages (SIIRH), in: Advances in Information Retrieval, Springer International Publishing. pp. 654–659.

[33] Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding, in: Burstein, J., Doran, C., Solorio, T. (Eds.), Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, (Long and Short Papers), Association for Computational Linguistics, Minneapolis, MN, USA. pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423, doi:10.18653/v1/n19-1423.

[34] Dey, S., Luo, H., Fokoue, A., Hu, J., Zhang, P., 2018. Predicting adverse drug reactions through interpretable deep learning framework. BMC Bioinformatics 19, 476.

[35] Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271.

[36] Donnelly, K., 2006. SNOMED-CT: The advanced terminology and coding system for ehealth. Books Google 121, 279–290.

[37] Du, J., Chen, Q., Peng, Y., Xiang, Y., Tao, C., Lu, Z., 2019. ML-Net: multi-label classification of biomedical texts with deep neural networks. J. Am. Med. Inform. Assoc. 26, 1279–1285.

[38] Evans, V., 2006. Lexical concepts, cognitive models and meaning-construction.

[39] Farouk, M., 2018. Sentence semantic similarity based on word embedding and WordNet, in: 2018 13th International Conference on Computer Engineering and Systems (ICCES), ieeexplore.ieee.org. pp. 33–37.

[40] Gerlach, M., Shi, H., Amaral, L.A.N., 2019. A universal information theoretic approach to the identification of stopwords. Nature Machine Intelligence 1, 606–612.

[41] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H., 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. arXiv e-prints , arXiv:2007.15779arXiv:2007.15779.

[42] Hahn, U., Oleynik, M., 2020. Medical information extraction in the age of deep learning. Yearb. Med. Inform. 29, 208–220.

[43] Han, L., Kashyap, A.L., Finin, T., Mayfield, J., Weese, J., 2013. UMBC_EBIQUITY-CORE: semantic textual similarity systems, in: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, ACL. pp. 44–52.

[44] Harispe, S., Ranwez, S., Janaqi, S., Montmain, J., 2014. The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. Bioinformatics 30, 740–742.

[45] Harris, Z., 1954. Distributional hypothesis. Word World 10, 146–162.

[46] Hassanzadeh, H., Groza, T., Nguyen, A., Hunter, J., 2015. A supervised approach to quantifying sentence similarity: with application to evidence based medicine. PLoS One 10, e0129392.

[47] Inan, E., 2020. SimiT: A text similarity method using lexicon and dependency representations. New Generation Computing , 1–22.

[48] Islam, A., Inkpen, D., 2008. Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data 2, 10:1–10:25.

[49] Jaccard, P., 1908. Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. sci. nat. 44, 223–270.

[50] Jimenez, S., Becerra, C., Gelbukh, A., 2012. Soft cardinality: A parameterized similarity function for text comparison, in: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proc. of the main conference and the shared task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), ACL. pp. 449–453.

[51] Johnson, A.E.W., Pollard, T.J., Shen, L., Lehman, L.W.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G., 2016. MIMIC-III, a freely accessible critical care database. Sci Data 3, 160035.

[52] Kajiwara, T., Bollegala, D., Yoshida, Y., Kawarabayashi, K.I., 2017. An iterative approach for the global estimation of sentence similarity. PLoS One 12, e0180885.

[53] Kalyan, K.S., Sangeetha, S., 2020. SECNLP: A survey of embeddings in clinical natural language processing. J. Biomed. Inform. 101, 103323.

[54] Khattak, F.K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., Rudzicz, F., 2019. A survey of word embeddings for clinical text. Journal of Biomedical Informatics: X 4, 100057.

[55] Kim, S., Kim, W., Comeau, D., Wilbur, W.J., 2012. Classifying gene sentences in biomedical literature by combining high-precision gene identifiers, in: Proc. of the 2012 Workshop on Biomedical Natural Language Processing, pp. 185–192.

[56] Kim, S.N., Martinez, D., Cavedon, L., Yencken, L., 2011. Automatic classification of sentences to support evidence based medicine. BMC Bioinformatics 12 Suppl 2, 5.

[57] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S., 2015. Skip-Thought vectors, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 28. Curran Associates, pp. 3294–3302.

[58] Kosorus, H., Bögl, A., Küng, J., 2012. Semantic similarity between queries in QA system using a domain-specific taxonomy, in: ICEIS (1), pp. 241–246.

[59] Krause, E.F., 1986. Taxicab Geometry: An Adventure in Non-Euclidean Geometry. Courier Corporation, Online.

[60] Lamurias, A., Ruas, P., Couto, F.M., 2019. PPR-SSM: personalized PageRank and semantic similarity measures for entity linking. BMC Bioinformatics 20, 534.

[61] Lara-Clares, A., Lastra-Díaz, J.J., Garcia-Serrano, A., 2021. Protocol for a reproducible experimental survey on biomedical sentence similarity. PLoS One 16, e0248663.

[62] Lara-Clares, A., Lastra-Díaz, J.J., Garcia-Serrano, A., 2022a. HESML V2R1 Java software library of semantic similarity measures for the biomedical domain. e-cienciaDatos, v1. https://doi.org/10.21950/DOI.

[63] Lara-Clares, A., Lastra-Díaz, J.J., Garcia-Serrano, A., 2022b. A reproducibility protocol and dataset on the biomedical sentence similarity. URL: https://www.protocols.io/view/a-reproducibility-protocol-and-dataset-on-the-biom-b5ckq2uw, doi:10.17504/protocols.io.b5ckq2uw.

[64] Lara-Clares, A., Lastra Diaz, J.J., Garcia Serrano, A., 2022c. Reproducible experiments on word and sentence similarity measures for the biomedical domain. URL: https://doi.org/10.21950/EPNXTR, doi:10.21950/EPNXTR.

[65] Lara-Clares, A., Lastra Diaz, J.J., Garcia Serrano, A., 2022d. Reproducible experiments on word and sentence similarity measures for the biomedical domain. URL: https://doi.org/10.21950/EPNXTR, doi:10.21950/EPNXTR.

[66] Lara-Clares, A., Lastra-Díaz, J.J., Garcia-Serrano, A., 2022e. A reproducible experimental survey on biomedical sentence similarity: A string-based method sets the state of the art. PLOS ONE 17, 1–44. URL: https://doi.org/10.1371/journal.pone.0276539, doi:10.1371/journal.pone.0276539.

[67] Lastra-Díaz, J.J., García-Serrano, A., 2015a. A new family of information content models with an experimental survey on WordNet. Knowledge-Based Systems 89, 509–526.

[68] Lastra-Díaz, J.J., García-Serrano, A., 2015b. A novel family of IC-based similarity measures with a detailed experimental survey on WordNet. Engineering Applications of Artificial Intelligence Journal 46, 140–153.

[69] Lastra-Díaz, J.J., García-Serrano, A., Batet, M., Fernández, M., Chirigati, F., 2017. HESML: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems 66, 97–118.

[70] Lastra-Díaz, J.J., Goikoetxea, J., Hadj Taieb, M., García-Serrano, A., Ben Aouicha, M., Agirre, E., Sánchez, D., 2021. A large reproducible benchmark of ontology-based methods and word embeddings for word similarity. Information Systems 96, 101636.

[71] Lastra-Díaz, J.J., Goikoetxea, J., Hadj Taieb, M.A., García-Serrano, A., Ben Aouicha, M., Agirre, E., 2019a. A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence 85, 645–665.

[72] Lastra-Díaz, J.J., Goikoetxea, J., Hadj Taieb, M.A., García-Serrano, A., Ben Aouicha, M., Agirre, E., 2019b. Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity. Data in Brief .

[73] Lastra-Diaz, J.J., Goikoetxea, J., Hadj Taieb, M.A., García-Serrano, A., Ben Aouicha, M., Agirre, E., 2019. A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence 85, 645 – 665.

[74] Lastra-Díaz, J.J., Lara-Clares, A., García-Serrano, A., 2020a. HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models. e-cienciaDatos, v2. https://doi.org/10.21950/1RRAWJ.

[75] Lastra-Díaz, J.J., Lara-Clares, A., Garcia-Serrano, A., 2020b. HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models. e-cienciaDatos, v1. https://doi.org/10.21950/1RRAWJ. URL: https://doi.org/10.21950/1RRAWJ, doi:10.21950/1RRAWJ.

[76] Lastra-Díaz, J.J., Lara-Clares, A., García-Serrano, A., 2020c. Reproducibility dataset for a benchmark of biomedical semantic measures libraries. e-cienciaDatos. https://doi.org/10.21950/OTDA4Z.

[77] Lastra-Díaz, J.J., Lara-Clares, A., Garcia-Serrano, A., 2022. HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey. BMC Bioinformatics 23, 23.

[78] Lastra-Díaz, J., 2017. Recent Advances in Ontology-based Semantic Similarity Measures and Information Content Models based on WordNet. Ph.D. thesis. UNED.

[79] Lawlor, L.R., 1980. Overlap, similarity, and competition coefficients. Ecology 61, 245–251.

[80] Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: International Conference on Machine Learning, Journal of Machine Learning Research. pp. 1188–1196.

[81] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J., 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36, 1234–1240. URL: https://doi.org/10.1093/bioinformatics/btz682, doi:10.1093/bioinformatics/btz682, arXiv:https://academic.oup.com/bioinformatics/article-pdf/36/4/1234/32527770/btz6

[82] Lee, M.C., Chang, J.W., Hsieh, T.C., 2014. A grammar-based semantic similarity algorithm for natural language sentences. ScientificWorldJournal 2014, 437162.

[83] Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, Springer. pp. 707–710.

[84] Li, Y., McLean, D., Bandar, Z.A., James, D.O., Crockett, K., 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. 18, 1138–1150.

[85] Lin, D., 1998. An information-theoretic definition of similarity, in: Proc. of the 15th International Conference on Machine Learning, Madison, WI. pp. 296–304.

[86] Lithgow-Serrano, O., Gama-Castro, S., Ishida-Gutiérrez, C., Collado-Vides, J., 2020. L-regulon: A novel soft-curation approach supported by a semantic enriched reading for regulondb literature. bioRxiv doi:10.1101/2020.04.26.062745.

[87] Lithgow-Serrano, O., Gama-Castro, S., Ishida-Gutiérrez, C., Mejía-Almonte, C., Tierrafría, V.H., Martínez-Luna, S., Santos-Zavaleta, A., Velázquez-Ramírez, D., Collado-Vides, J., 2019. Similarity corpus on microbial transcriptional regulation. Journal of Biomedical Semantics 10, 8.

[88] Liu, H., Hunter, L., Kešelj, V., Verspoor, K., 2013. Approximate subgraph matching-based literature mining for biomedical events and relations. PLoS One 8, e60954.

[89] Maharjan, N., Banjade, R., Gautam, D., Tamang, L.J., Rus, V., 2017. DT_Team at SemEval-2017 task 1: Semantic similarity using alignments, Sentence-Level embeddings and gaussian mixture model output, in: Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017), ACL. pp. 120–124.

[90] Manning, C.D., Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press, Online.

[91] McInnes, B.T., Pedersen, T., Pakhomov, S.V.S., 2009. UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity, in: Proc. of the Annual Symposium of AMIA, San Francisco, CA. pp. 431–435.

[92] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv arXiv:1301.3781.

[93] Miller, G.A., 1995a. WordNet: A Lexical Database for English. Communications of the ACM 38, 39–41.

[94] Miller, G.A., 1995b. WordNet: A lexical database for english. ACM 38, 39–41.

[95] Mishra, M.K., Viradiya, J., 2019. Survey of Sentence Embedding Methods. International Journal of Applied Science and Computations 6, 592–592.

[96] Mishra, R., Bian, J., Fiszman, M., Weir, C.R., Jonnalagadda, S., Mostafa, J., Del Fiol, G., 2014. Text summarization in the biomedical domain: a systematic review of recent research. J. Biomed. Inform. 52, 457–467.

[97] Nelson, S.J., Johnston, W.D., Humphreys, B.L., 2001. Relationships in medical subject headings (MeSH), in: Bean, C.A., Green, R. (Eds.), Relationships in the Organization of Knowledge. Springer Netherlands, Dordrecht, pp. 171–184.

[98] Newman-Griffis, D., Lai, A., Fosler-Lussier, E., 2017. Insights into analogy completion from the biomedical domain, in: BioNLP 2017, Association for Computational Linguistics, Vancouver, Canada,. pp. 19–28. doi:10.18653/v1/W17-2303.

[99] Nguyen, H.T., Duong, P.H., Cambria, E., 2019. Learning short-text semantic similarity with word embeddings and external knowledge sources. Elsevier 182, 104842.

[100] Oliva, J., Serrano, J.I., del Castillo, M.D., Iglesias, Á., 2011. SyMSS: A syntax-based measure for short-text semantic similarity. Data Knowl. Eng. 70, 390–405.

[101] Pagliardini, M., Gupta, P., Jaggi, M., 2018. Unsupervised learning of sentence embeddings using compositional n-gram features, in: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational

Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana. pp. 528–540. doi:10.18653/v1/N18-1049.

[102] Pawar, A., Mago, V., 2019. Challenging the boundaries of unsupervised learning for semantic similarity. IEEE Access 7, 16291–16308.

[103] Pelletier, F.J., 1994. The principle of semantic compositionality. Topoi 13, 11–24.

[104] Peng, Y., Yan, S., Lu, Z., 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets, in: Proc. of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy. pp. 58–65. doi:10.18653/v1/W19-5006.

[105] Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation, in: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP), ACL Web. pp. 1532–1543.

[106] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations, in: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana. pp. 2227–2237. doi:10.18653/v1/N18-1202.

[107] Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., Ananiadou, S., 2013. Distributional semantics resources for biomedical text processing. Proc. of LBM , 39–44.

[108] Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics 19, 17–30.

[109] Ranasinghe, T., Orasan, C., Mitkov, R., 2019. Enhancing unsupervised sentence similarity methods with deep contextualised word representations, in: Proc. of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), INCOMA Ltd., Varna, Bulgaria. pp. 994–1003.

[110] Rastegar-Mojarad, M., Komandur Elayavilli, R., Liu, H., 2016. BELTracker: evidence sentence retrieval for BEL statements. Database 2016.

[111] Ravikumar, K.E., Rastegar-Mojarad, M., Liu, H., 2017. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database 2017.

[112] Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P., 2016. Samsung poland NLP team at SemEval-2016 task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods

to measure semantic similarity, in: Proc. of the 10th International Workshop on Semantic Evaluation (SemEval-2016), ACL. pp. 602–608.

[113] Šarić, F., Glavaš, G., Karan, M., Šnajder, J., Bašić, B.D., 2012. TakeLab: Systems for measuring semantic text similarity, in: Proc. of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proc. of the Main Conference and the Shared Task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation, Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 441–448.

[114] Sarrouti, M., Ouatik El Alaoui, S., 2017. A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering. J. Biomedical Informatics 68, 96–103.

[115] Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., Chute, C.G., 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. J. Am. Med. Inform. Assoc. 17, 507–513.

[116] Shajalal, M., Aono, M., 2019. Semantic textual similarity between sentences using bilingual word semantics. Progress in Artificial Intelligence 8, 263–272.

[117] Shang, Y., Li, Y., Lin, H., Yang, Z., 2011. Enhancing biomedical text summarization using semantic relation extraction. PLoS One 6, e23862.

[118] Shen, D., Wang, G., Wang, W., Min, M.R., Su, Q., Zhang, Y., Li, C., Henao, R., Carin, L., 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia. pp. 440–450. doi:10.18653/v1/P18-1041.

[119] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., OBI Consortium, Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S., 2007. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology 25, 1251–1255.

[120] Sogancioglu, G., Öztürk, H., Özgür, A., 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics 33, 49–58.

[121] Sultan, M.A., Bethard, S., Sumner, T., 2014. DLS @ CU: Sentence similarity from word alignment, in: Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014), ACL. pp. 241–246.

[122] Sultan, M.A., Bethard, S., Sumner, T., 2015a. DLS @ CU: Sentence similarity from word alignment and semantic vector composition, in: Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015), ACL. pp. 148–153.

[123] Sultan, M.A., Bethard, S., Sumner, T., 2015b. Dls@ cu: Sentence similarity from word alignment and semantic vector composition, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), ACL. pp. 148–153.

[124] Tafti, A.P., Behravesh, E., Assefi, M., LaRose, E., Badger, J., Mayer, J., Doan, A., Page, D., Peissig, P., 2017. bigNN: An open-source big data toolkit focused on biomedical sentence classification, in: 2017 IEEE International Conference on Big Data (Big Data), pp. 3888–3896.

[125] Tawfik, N.S., Spruit, M.R., 2020. Evaluating sentence representations for biomedical text: Methods and experimental results. J. Biomed. Inform. , 103396.

[126] Taylor, J.R., Dirven, R., Cuyckens, H., 2003. Introduction: New directions in cognitive lexical semantic research, in: Taylor, J.R., Dirven, R., Cuyckens, H. (Eds.), Cognitive Approaches to Lexical Semantics. Mouton de Gruyter, pp. 1–28.

[127] The Gene Ontology Consortium, 2019. The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Research 47, D330–D338.

[128] Ukkonen, E., 1992. Approximate string-matching with q-grams and maximal matches. Theor. Comput. Sci. 92, 191–211.

[129] Wada, S., Takeda, T., Manabe, S., Konishi, S., Kamohara, J., Matsumura, Y., 2020. A pre-training technique to localize medical BERT and to enhance biomedical BERT. arXiv e-prints , arXiv:2005.07202arXiv:2005.07202.

[130] Wang, Y., Afzal, N., Fu, S., Wang, L., Shen, F., Rastegar-Mojarad, M., Liu, H., 2018a. MedSTS: a resource for clinical semantic textual similarity. Language Resources and Evaluation , 1–16.

[131] Wang, Y., Afzal, N., Liu, S., Rastegar-Mojarad, M., Wang, L., Shen, F., Fu, S., Liu, H., 2018b. Overview of the BioCreative/OHNLP challenge 2018 task 2: Clinical semantic textual similarity. Proc. of the BioCreative/OHNLP Challenge 2018.

[132] Wu, H., Huang, H., Jian, P., Guo, Y., Su, C., 2017. BIT at SemEval-2017 task 1: Using semantic information space to evaluate semantic textual similarity, in: Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017), ACL. pp. 77–84.

[133] Wu, H., Huang, H., Lu, W., 2016. Bit at semeval-2016 task 1: Sentence similarity based on alignments and vector with the weight of information content, in: Proc. of the 10th International Workshop on Semantic Evaluation (SemEval-2016), ACL. pp. 686–690.

[134] Wu, Z., Palmer, M., 1994. Verbs Semantics and Lexical Selection, in: Proc. of the Annual Meeting of ACL, ACL. pp. 133–138.

[135] Zhang, Y., Chen, Q., Yang, Z., Lin, H., Lu, Z., 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Sci Data 6, 52.

Part II

Publications

Chapter 7

Plos One Registered Report Protocol article

PLOS ONE

REGISTERED REPORT PROTOCOL

Protocol for a reproducible experimental survey on biomedical sentence similarity

Alicia Lara-Clares*, Juan J. Lastra-Díaz, Ana Garcia-Serrano

NLP & IR Research Group, E.T.S.I. Informática, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

* alara@lsi.uned.es

Abstract

Measuring semantic similarity between sentences is a significant task in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and biomedical text mining. For this reason, the proposal of sentence similarity methods for the biomedical domain has attracted a lot of attention in recent years. However, most sentence similarity methods and experimental results reported in the biomedical domain cannot be reproduced for multiple reasons as follows: the copying of previous results without confirmation, the lack of source code and data to replicate both methods and experiments, and the lack of a detailed definition of the experimental setup, among others. As a consequence of this reproducibility gap, the state of the problem can be neither elucidated nor new lines of research be soundly set. On the other hand, there are other significant gaps in the literature on biomedical sentence similarity as follows: (1) the evaluation of several unexplored sentence similarity methods which deserve to be studied; (2) the evaluation of an unexplored benchmark on biomedical sentence similarity, called Corpus-Transcriptional-Regulation (CTR); (3) a study on the impact of the pre-processing stage and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods; and finally, (4) the lack of software and data resources for the reproducibility of methods and experiments in this line of research. Identified these open problems, this registered report introduces a detailed experimental setup, together with a categorization of the literature, to develop the largest, updated, and for the first time, reproducible experimental survey on biomedical sentence similarity. Our aforementioned experimental survey will be based on our own software replication and the evaluation of all methods being studied on the same software platform, which will be specially developed for this work, and it will become the first publicly available software library for biomedical sentence similarity. Finally, we will provide a very detailed reproducibility protocol and dataset as supplementary material to allow the exact replication of all our experiments and results.

This is a Registered Report and may have an associated publication; please check the article page on the journal site for any related articles.

OPEN ACCESS

Citation: Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A (2021) Protocol for a reproducible experimental survey on biomedical sentence similarity. PLoS ONE 16(3): e0248663. https://doi.org/10.1371/journal.pone.0248663

Editor: Bridget McInnes, Virginia Commonwealth University, UNITED STATES

Received: November 9, 2020

Accepted: March 2, 2021

Published: March 24, 2021

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0248663

Introduction

Measuring semantic similarity between sentences is an important task in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and biomedical text mining, among

Data Availability Statement: All relevant data from this study will be made available upon study completion.

Funding: ALC UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018) https://www.uned.es/ The funders had and will not have a role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

others. For instance, the estimation of the degree of semantic similarity between sentences is used in text classification [1–3], question answering [4, 5], evidence sentence retrieval to extract biological expression language statements [6, 7], biomedical document labeling [8], biomedical event extraction [9], named entity recognition [10], evidence-based medicine [11, 12], biomedical document clustering [13], prediction of adverse drug reactions [14], entity linking [15], document summarization [16, 17] and sentence-driven search of biomedical literature [18], among other applications. In the question answering task, Sarrouti and El Alaomi [4] build a ranking of plausible answers by computing the similarity scores between each biomedical question and the candidate sentences extracted from a knowledge corpus. Allot et al. [18] introduce a system to retrieve the most similar sentences in the BioC biomedical corpus [19] called Litsense [18], which is based on the comparison of the user query with all sentences in the aforementioned corpus. Likewise, the relevance of the research in this area is endorsed by recent works based on sentence similarity measures, such as the work of Aliguliyev [16] in automatic document summarization, which shows that the performance of these applications depends significantly on the sentence similarity measures used.

The aim of any semantic similarity measure is to estimate the degree of similarity between two textual semantic units as perceived by a human being, such as words, phrases, sentences, short texts, or documents. Unlike sentences from the language in general use whose vocabulary and syntax is limited both in extension and complexity, most sentences in the biomedical domain are comprised of a huge specialized vocabulary made up of all sort of biological and clinical terms, in addition to an uncountable list of acronyms, which are combined in complex lexical and syntactic forms.

Most methods on biomedical sentence similarity are adaptations from methods for the general language domain, which are mainly based on the use of biomedical ontologies, as well as word and sentence embedding models trained on biomedical text corpora. For instance, Socioanglu et al. [20] introduce a set of sentence similarity measures for the biomedical domain, which are based on adaptations from the Li et al. [21] measure. Zhang et al. [22] introduce a set of pre-trained word embedding model called BioWordVec, which is based on a FastText [23] model trained on the titles and abstracts from PubMed articles and term sequences from the Medical Subject Headings (MeSH) thesaurus [24], whilst Chen et al. [25] introduce a set of pre-trained sentence embedding models called BioSentVec, which is based on a Sent2vec [26] model trained on the full text of PubMed articles and Medical Information Mart for Intensive Care (MIMIC-III) clinical notes [27], and Blagec et al. [28] introduce a set of word and sentence embedding models based on the training of FastText [23], Sent2Vec [26], Paragraph vector [29], and Skip-thoughts vectors [30] models on the full-text PubMed Central (PMC) Open Access dataset. Likewise, several contextualized word representation models, also known as language models, have also been adapted to the biomedical domain. For instance, Lee et al. [31] and Peng et al. [32] introduce two language models based on the Bidirectional Encoder Representations from Transformers (BERT) architecture [33], which are called BERT for Biomedical text mining (BioBERT) and Biomedical Language Understanding Evaluation of BERT (BlueBERT), respectively.

Nowadays, there are several works in the literature that experimentally evaluate multiple methods on biomedical sentence similarity. However, they are either theoretical or have a limited scope and cannot be reproduced. For instance, Kalyan et al. [34], Khattak et al. [35], and Alsentzer et al. [36] introduce theoretical surveys on biomedical embeddings with a limited scope. On the other hand, the experimental surveys introduced by Sogancioglu et al. [20], Blagec et al. [28], Peng et al. [32], and Chen et al. [25] among other authors, cannot be reproduced because of the lack of source code and data to replicate both methods and experiments, or the lack of a detailed definition of their experimental setups. Likewise, there are other recent works whose results need to be confirmed. For instance, Tawfik and Spruit [37] experimentally evaluate a set of pre-trained language models, whilst Chen et al. [38] propose a system to study the impact of a set of similarity measures on a Deep Learning ensembled model, which is based on a Random Forest model [39].

The main aim of this registered report is the introduction of a very detailed experimental setup for the development of the largest and reproducible experimental survey of methods on biomedical sentence similarity with the aim of elucidating the state of the problem, such as will be detailed in the motivation section. Our experiments will be based on our implementation and evaluation of all methods analyzed herein into a common and new software platform based on an extension of the Half-Edge Semantic Measures Library (HESML, http://hesml.lsi.uned.es) [40], called HESML for Semantic Textual Similarity (HESML-STS), as well as their subsequent recording with the Reprozip long-term reproducibility tool [41]. This work is based on our previous experience developing reproducible research in a series of publications in the area, such as the experimental surveys on word similarity introduced in [42–45], whose reproducibility protocols and datasets [46, 47] are detailed and independently confirmed in two reproducible papers [40, 48]. The experiments in this new software platform will evaluate most of the sentence similarity methods for the biomedical domain reported in the literature, as well as a set of unexplored methods which are based on adaptations from the general language domain.

Main motivations and research questions

Our main motivation is the lack of a reproducible experimental survey on biomedical sentence similarity, which allows the state of the problem to be elucidated in a sound and reproducible way by answering the following research questions:

RQ1. Which methods get the best results on biomedical sentence similarity?

RQ2. Is there a statistically significant difference between the best performing methods and the remaining ones?

RQ3. What is the impact of the biomedical Named Entity Recognition (NER) tools on the performance of the methods on biomedical sentence similarity?

RQ4. What is the impact of the pre-processing stage on the performance of the methods on biomedical sentence similarity?

RQ5. What are the main drawbacks and limitations of current methods on biomedical sentence similarity?

Most experimental results reported in this line of research cannot be reproduced for numerous reasons. For instance, Sogancioglu et al. [20] provide neither the pre-trained models used in their experiments nor a detailed guide for replicating them and their software artifacts do not reproduce all of their results. Blagec et al. [28] provide neither a detailed definition of their experimental setup nor their source code and pre-processed data, as well as the pre-trained models used in their experiments. Chen et al. [25] set the state of the art on biomedical sentence similarity by copying results from Blagec et al. [28]; thus, their work allows neither previous results to be confirmed nor are they directly compared with other works. In several cases, biomedical language models based on BERT, such as BioBERT [31] and NCBI-BlueBERT [32], can be reproduced neither in an unsupervised context nor in any other supervised way, because of the high computational requirements and the non-deterministic nature of the methods used for their training, respectively.

A second motivation is the implementation of a set of unexplored methods which are based on adaptations from other methods proposed for the general language domain. A third motivation is the evaluation in the same software platform of the benchmarks on biomedical sentence similarity reported in the literature as follows: Biomedical Semantic Similarity Estimation System (BIOSSES) [20] and Medical Semantic Textual Similarity (MedSTS) [49] datasets, as well as the evaluation for the first time of the Microbial Transcriptional Regulation (CTR) [50] dataset in a sentence similarity task, despite it having been previously evaluated in other related tasks, such as the curation of gene expressions from scientific publications [51]. A fourth motivation is a study on the impact of the pre-processing stage and NER tools on the performance of the sentence similarity methods, such as that done by Gerlach et al. [52] for stop-words in topic modeling task. And finally, our fifth motivation is the lack of reproducibility software and data resources on this task, which allow an easy replication and confirmation of previous methods, experiments, and results in this line of research, as well as encouraging the development and evaluation of new sentence similarity methods.

Definition of the problem and contributions

The main research problem tackled in this work is the design and implementation of a large and reproducible experimental survey on sentence similarity measures for the biomedical domain. Our main contributions are as follows: (1) the largest, and for the first time, reproducible experimental survey on biomedical sentence similarity; (2) the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity; (3) the evaluation of a set of previously unexplored methods, as well as the evaluation of a new word embedding model based on FastText and trained on the full-text of articles in the PMC-BioC corpus [19]; (4) the integration for the first time of most sentence similarity methods for the biomedical domain in the same software library called HESML-STS; and finally, (5) a detailed reproducibility protocol together with a collection of software tools and datasets, which will be provided as supplementary material to allow the exact replication of all our experiments and results.

The rest of the paper is structured as follows. First, we introduce a comprehensive and updated categorization of the literature on sentence semantic similarity measures for the general and biomedical language domains. Next, we describe a detailed experimental setup for our experiments on biomedical sentence similarity. Finally, we introduce our conclusions and future work.

Methods on sentence semantic similarity

This section introduces a comprehensive categorization of the methods on sentence semantic similarity for the general and biomedical language domains, which includes most of the methods reported in the literature. The categorization, shown in Fig 1, is organized into two classes as follows: (a) the methods proposed for the general domain; and (b) the methods proposed for the biomedical domain. For a more detailed presentation of the methods categorized herein, we refer the reader to several surveys on ontology-based semantic similarity measures [43, 45], word embeddings [35, 45], sentence embeddings [34, 53], and neural language models [34, 54].

Literature review methodology

We conducted our literature review following the next steps: (1) formulation of our research questions; (2) search of relevant publications on biomedical sentence similarity, especially all methods and works whose experimental evaluation is based on the sentence similarity benchmarks considered in our experimental setup; (3) definition of inclusion and exclusion criteria

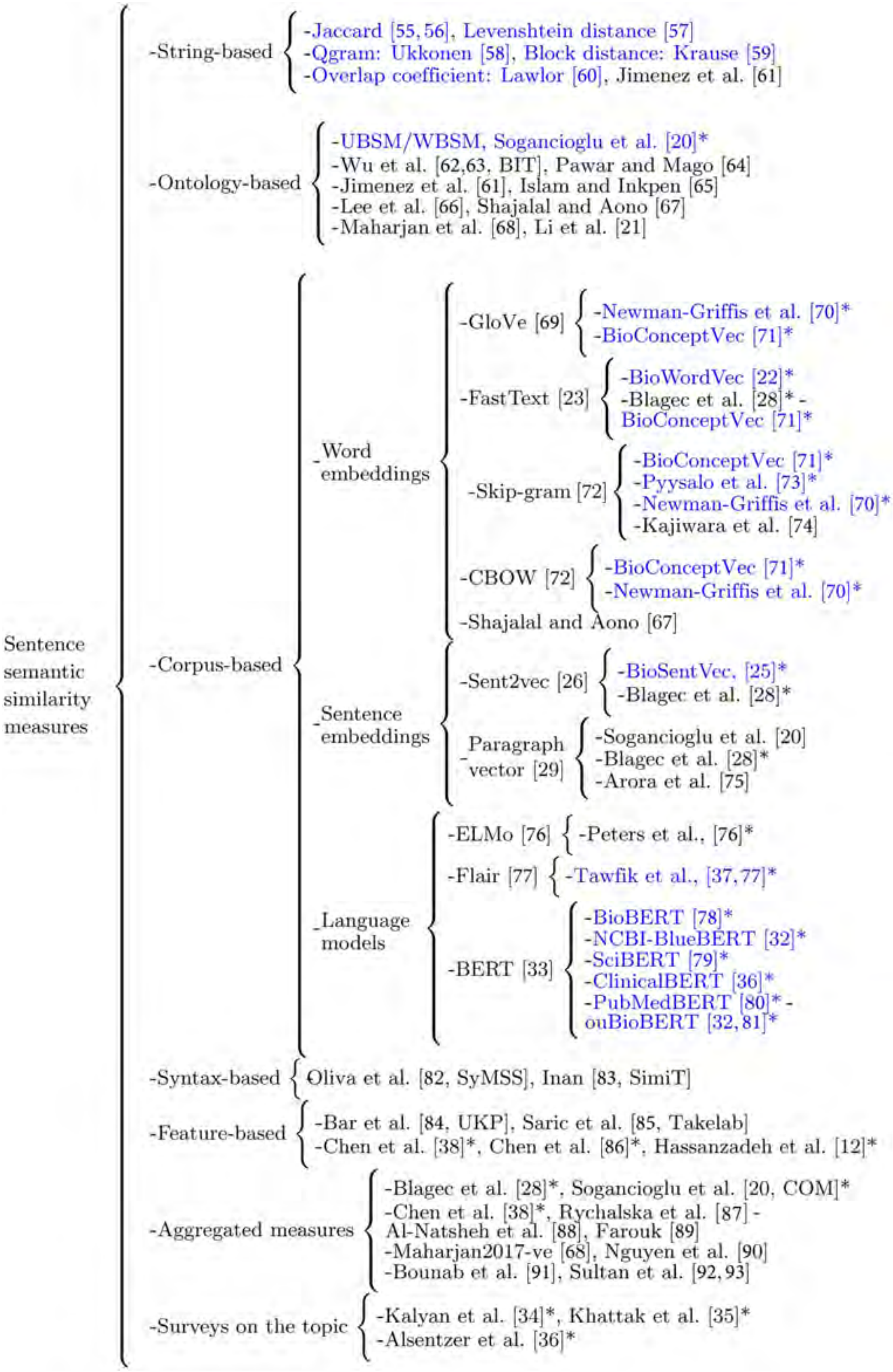

Sentence semantic similarity measures
-String-based
-Jaccard [55, 56], Levenshtein distance [57]
-Qgram: Ukkonen [58], Block distance: Krause [59]
-Overlap coefficient: Lawlor [60], Jimenez et al. [61]
-Ontology-based
-UBSM/WBSM, Sogancioglu et al. [20]*
-Wu et al. [62,63, BIT], Pawar and Mago [64]
-Jimenez et al. [61], Islam and Inkpen [65]
-Lee et al. [66], Shajalal and Aono [67]
-Maharjan et al. [68], Li et al. [21]
-Corpus-based
_Word embeddings
-GloVe [69]
-Newman-Griffis et al. [70]*
-BioConceptVec [71]*
-FastText [23]
-BioWordVec [22]*
-Blagec et al. [28]* - BioConceptVec [71]*
-Skip-gram [72]
-BioConceptVec [71]*
-Pyysalo et al. [73]*
-Newman-Griffis et al. [70]*
-Kajiwara et al. [74]
-CBOW [72]
-BioConceptVec [71]*
-Newman-Griffis et al. [70]*
-Shajalal and Aono [67]
_Sentence embeddings
-Sent2vec [26]
-BioSentVec, [25]*
-Blagec et al. [28]*
_Paragraph vector [29]
-Sogancioglu et al. [20]
-Blagec et al. [28]*
-Arora et al. [75]
_Language models
-ELMo [76]
-Peters et al., [76]*
-Flair [77]
-Tawfik et al., [37, 77]*
-BERT [33]
-BioBERT [78]*
-NCBI-BlueBERT [32]*
-SciBERT [79]*
-ClinicalBERT [36]*
-PubMedBERT [80]* - ouBioBERT [32, 81]*
-Syntax-based
Oliva et al. [82, SyMSS], Inan [83, SimiT]
-Feature-based
-Bar et al. [84, UKP], Saric et al. [85, Takelab]
-Chen et al. [38]*, Chen et al. [86]*, Hassanzadeh et al. [12]*
-Aggregated measures
-Blagec et al. [28]*, Sogancioglu et al. [20, COM]*
-Chen et al. [38]*, Rychalska et al. [87] - Al-Natsheh et al. [88], Farouk [89]
-Maharjan2017-ve [68], Nguyen et al. [90]
-Bounab et al. [91], Sultan et al. [92, 93]
-Surveys on the topic
-Kalyan et al. [34]*, Khattak et al. [35]*
-Alsentzer et al. [36]*

Fig 1. Categorization of the main sentence similarity methods reported in the literature. Citations with an asterisk (*) point out adaptations for the biomedical domain, whilst the citations in blue highlight those methods that will be reproduced and evaluated in our experiments (see Table 8). [12, 20–23, 25, 26, 28, 29, 32–38, 55–93].

https://doi.org/10.1371/journal.pone.0248663.g001

of the methods; (4) definition of the study limitations and risks; and (5) definition of the evaluation metrics. Publications on our research topic were mainly searched in the Web Of Science (WOS) and Google Scholar databases, and the SemEval [94–99] and BioCreative/OHNLP [100] conference series. In order to build a first set of relevant works on the topic, we selected a seed set of highlighted publications and datasets on biomedical sentence similarity [20, 21, 25, 28, 31, 49] from the aforementioned information sources. Then, we reviewed all the papers related to sentence similarity which cited any seed publication or dataset. Finally, starting from seed publications and datasets, we extracted those methods that could be implemented and evaluated in our experiments, and we downloaded and checked all the available pre-trained models. Our main goal was trying an independent replication or evaluation of all methods previously evaluated on the biomedical sentence similarity benchmarks considered in our experiments.

Methods proposed for the general language domain

There is a large corpus of literature on sentence similarity methods for the general language domain as the result of a significant research effort during the last decade. However, the literature for the biomedical domain is much more limited. Research for the general language domain has mainly been boosted by the SemEval Short Text Similarity (STS) evaluation series since 2012 [94–99], which has generated a large number of contributions in the area [84, 85, 92, 101, 102], as well as an STS benchmark dataset [99]. On the other hand, the development of sentence similarity benchmarks for the biomedical domain is much more recent. Currently, there are only three datasets for the evaluation of methods on biomedical sentence similarity, called BIOSSES [20], MedSTS [49], and CTR [50]. BIOSSES was introduced in 2017 and it is limited to 100 sentence pairs with their corresponding similarity scores, whilst MedSTS$_{full}$ is made up by 1,068 scored sentence pairs of the MedSTS dataset [100], which contains 174,629 sentence pairs gathered from a clinical corpus on biomedical sentence similarity. Finally, the CTR dataset includes 171 sentence pairs, but it has not been evaluated yet because of its recent publication in 2019.

Fig 1 shows our categorization of the current sentence semantic similarity measures into six subfamilies as follows. First, string-based measures, whose main feature is the use of the explicit information contained at the character or word level in the sentences to estimate their similarity. Second, ontology-based measures, such as those introduced by Sogancioglu et al. [20], whose main feature is the computation of the similarity between sentences by combining the pairwise similarity scores of their constituent words and concepts [45] based on the Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [103] and WordNet [104] ontologies, and the MeSH thesaurus [24]. Third, corpus-based methods based on the distributional hypothesis [105], such as the work of Pyysalo et al. [73], which states that words sharing semantic relationships tend to occur in similar contexts. The corpus-based methods can be divided into three subcategories as follows: (a) methods based on word embeddings, (b) sentence embeddings, and (c) language models. Methods based on word embeddings combine the word vectors corresponding to the words contained in a sentence to build a sentence vector, such as the averaging Simple Word EMbeddings (SWEM) models introduced by Shen et al. [106], whilst methods based on sentence embeddings directly compute a vector representation for each sentence. Then, the similarity between sentence pairs is calculated using any vector-based similarity metric, such as the cosine function. On the other hand, language models, which explore the concept of Transfer Learning by creating a pre-trained model on a large raw text corpus and fine-tuning those models in downstream tasks, such as sentence semantic similarity, with the pioneering work of Peng et al.

[32]. Fourth, syntax-based methods, which rely on the use of explicit syntax information, as well as the structure of the words that compound the sentences, such as the pioneering work of Oliva et al. [82]. Fifth, feature-based approaches, such as the work of Chen et al. [86], whose main idea is to compute the similarity of two sentences by measuring at different language perspectives the properties that they have in common or not, such as lexical patterns, word semantics and named entities. Finally, aggregated methods, whose main feature is the combination of other sentence similarity methods.

Methods proposed for the biomedical domain

Like that mentioned in the introduction, most methods on biomedical sentence similarity are adaptations from the general domain, such as the methods which will be evaluated in this work (see Table 8). Sogancioglu et al. [20] proposed a set of ontology-based measures called WordNet-based Similarity Measure (WBSM) and UMLS-based Similarity Measure (UBSM), which are based on the Li et al. [21] measure. All word and sentence embedding models for the biomedical domain in the literature are based on well-known models from the general domain. Pyysalo et al. [73] train a Skip-gram [72] model on document titles and abstracts from the PubMed XML dataset, and all text content of the PMC Open Access dataset. Newman-Griffis et al. [70] and Chen et al. [71] train GloVe [69], Skip-gram, and Continuous Bag of Words (CBOW) [72] models using PubMed information, whilst Zhang et al. [22] and Chen et al. [71] train FastText [23] models using PubMed and MeSH. Blagec et al. [28] introduce a set of neural embedding models based on the training of FastText [23], Sent2Vec [26], Paragraph vector [29], and Skip-thoughts vectors [30] models on the PMC dataset. Chen et al. [25] also introduce a sentence embedding model called BioSentVec, which is based on Sent2vec [26]. Likewise, we also find adaptations from several contextualized word representation models, also known as language models, for the biomedical domain. Tawfik and Spruit [37] evaluate a Flair-based [77] model trained on PubMed abstracts. Ranashinghe et al. [78], Peng et al. [32], Beltagy et al. [79], Alsentzer et al. [36], Gu et al. [80] and Wada et al. [32, 81] introduce BERT-based models [33] trained on biomedical information. However, these later models do not perform well in an unsupervised context because they are trained for downstream tasks using a supervised approach, which has encouraged Ranashinghe et al. [78] to explore a set of unsupervised approximations for evaluating BioBERT [76] and Embeddings for Language Models (ELMo) [76] models in the biomedical domain.

The reproducible experiments on biomedical sentence similarity

This section introduces a very detailed experimental setup describing our plan to evaluate and compare most of the sentence similarity methods for the biomedical domain. In order to set the state of the art of the problem in a sound and reproducible way, the goals of our experiments are as follows: (1) the evaluation of most of methods on biomedical sentence similarity onto the same software platform; (2) the evaluation of a set of new sentence similarity methods adapted from their definitions for the general-language domain; (3) the setting of the state of the art of the problem in a sound and reproducible way; (4) the replication and independent confirmation of previously reported methods and results; (5) a study on the impact of different pre-processing configurations on the performance of the sentence similarity methods; (6) a study on the impact of different Name Entity Recognition (NER) tools, such as MetaMap [107] and clinic Text Analysis and Knowledge Extraction System (cTAKES) [108], onto the performance of the sentence similarity methods; and finally, (7) a detailed statistical significance analysis of the results.

Selection of methods

The methodology for the selection of the sentence similarity methods was as follows: (a) identification of all the methods in the biomedical domain that were evaluated in BIOSSES [20] and MedSTS [49] datasets; (b) identification of those methods reported for the general domain not evaluated in the biomedical domain yet; and (c) definition of the criteria for the selection and exclusion of methods.

Our selection criteria for the sentence similarity methods to be reproduced and evaluated herein have been significantly conditioned by the availability of multiple sources of information, as follows: (1) pre-trained models; (2) source code; (3) reproducibility data; (4) detailed descriptions of the methods and experiments; (5) reproducibility guidelines; and finally, (6) the computational requirements for training several models. This work reproduces and evaluates most of the sentence similarity methods for the biomedical domain reported in the literature, as well as other methods that have not been explored in this domain yet. Some of these later unexplored methods are either variants or adaptations of methods previously proposed for the general or biomedical domain, which are evaluated for the first time in this work, such as the WBSM-cosJ&C [20, 43, 109], WBSM-coswJ&C [20, 43, 109], WBSM-Cai [20, 100], UBSM-cosJ&C [20, 43, 109], UBSM-coswJ&C [20, 43, 109], and UBSM-Cai [20, 100] methods detailed in Tables 2 and 3.

Biomedical methods not evaluated. We discard the evaluation of the pre-trained Paragraph vector model introduced by Sogancioglu et al. [20] because it is not provided by the authors, despite this model having achieved the best results in their work. Likewise, we also discard the evaluation of the pre-trained Paragraph vector, sent2vec, and fastText models introduced by Blagec et al. [28], because the authors provide neither their pre-trained models nor their source code and the detailed post-processing configuration used in their experiments. Thus, not all of the aforementioned models can be reproduced.

Tables 1 and 2 detail the configuration of the string-based measures and ontology-based measures that will be evaluated in this work, respectively. Both WBSM and UBSM methods will be evaluated in combination with the following word or concept similarity measures: Rada et al. [111], Jiang&Conrath [112], and three state-of-the-art unexplored measures, called cosJ&C [43], coswJ&C [43], and Cai et al. [110]. The word similarity measure which reports the best results will be used to evaluate the COM method [20]. Table 3 details the sentence similarity methods based on the evaluation of pre-trained character, word, and sentence

Table 1. Detailed setup for the string-based sentence similarity measures which will be evaluated in this work. All the string-based measures will follow the implementation of Sogancioglu et al. [20], who use the Simmetrics library [113].

ID	Method	Detailed setup of each method
M1	Qgram [58]	$sim(a,b) = \frac{2\times\lvert q{-}grams(a)\cup q{-}grams(b)\rvert}{\lvert q{-}grams(a)\rvert+\lvert q{-}grams(b)\rvert}$, being a and b sets of q words, and with q = 3.
M2	Jaccard [55, 56]	$sim(a,b) = \frac{\lvert a\cup b\rvert}{\lvert a\cap b\rvert}$, being a and b sets of words of the first and second sentence respectively.
M3	Block distance [59]	$sim(a,b) = 1 - \frac{\sum_{n=1}^{n=\lvert a\rvert+\lvert b\rvert}(v_{an}-v_{bn})}{\lvert a\rvert+\lvert b\rvert}$, being a and b sets of words of the first and second sentence respectively; and v_a and v_b the frequency vectors of a and b.
M4	Levenshtein distance [57]	Measures the minimal cost number of insertions, deletions and replacements needed for transforming the first into the second sentence. Insert, delete and substitution cost set to 1.
M5	Overlap coefficient [60]	$sim(a,b) = \frac{\lvert a\cap b\rvert}{\lvert Min(\lvert a\rvert,\lvert b\rvert)\rvert}$, being a and b sets of words of the first and second sentence respectively.

https://doi.org/10.1371/journal.pone.0248663.t001

Table 2. Detailed setup for the ontology-based sentence similarity measures which will be evaluated in this work.

ID	Sentence similarity method	Detailed setup of each method
M6	WBSM-Rada [20, 111]	WBSM [20] combined with Rada [111] measure
M7	WBSM-J&C [20, 112]	WBSM [20] combined with J&C [112] measure
M8	WBSM-cosJ&C [20, 43] (this work)	WBSM [20] with cosJ&C [43] measure and Sanchez et al. [109] IC model
M9	WBSM-coswJ&C [20, 43] (this work)	WBSM [20] with coswJ&C [43] measure and Sanchez et al. [109] IC model
M10	WBSM-Cai [20, 110] (this work)	WBSM [20] combined with Cai et al. [110] measure and Cai et al. [110] IC model
M11	UBSM-Rada [20, 111]	UBSM [20] with Rada et al. [111] measure
M12	UBSM-J&C [20, 112]	UBSM [20] combined with J&C [112] measure
M13	UBSM-cosJ&C [20, 43] (this work)	UBSM [20] with cosJ&C [43] measure and Sanchez et al. [109] IC model
M14	UBSM-coswJ&C [20, 43] (this work)	UBSM [20] with coswJ&C [43] measure and Sanchez et al. [109] IC model
M15	UBSM-Cai [20, 110] (this work)	UBSM [20] combined with Cai et al. [110] measure and Cai et al. [110] IC model
M16	COM [20]	$\lambda \cdot$WBSM + $(1 - \lambda) \cdot$ UBSM [20] with $\lambda = 0.5$ and the best word similarity measure

https://doi.org/10.1371/journal.pone.0248663.t002

embedding models that will be evaluated in this work. We will also evaluate for the first time a sentence similarity method, named FastText-SkGr-BioC and detailed in Table 3), which is based on a FastText [23] word embedding model trained on the full text of the PMC-BioC [19] articles. Finally, Table 4 details the pre-trained language models that will be evaluated in our experiments.

Table 3. Detailed setup for the sentence similarity methods based on pre-trained character, Word Embedding (WE), and Sentence Embedding (SE) models which will be evaluated in this work.

ID	Sentence similarity method	Detailed setup of each method
M17	Flair [77]	Contextual string embeddings trained on PubMed
M18	Pyysalo et al. [73]	Skip-gram trained on PubMed + PMC
M19	BioConceptVec [71]	Skip-gram WE model trained on PubMed using word2vec program
M20	BioConceptVec [71]	CBOW WE model trained on PubMed using word2vec program
M21	Newman-Griffis et al. [70]	Skip-gram WE model trained on PubMed using word2vec program
M22	Newman-Griffis et al. [70]	CBOW WE model trained on PubMed using word2vec program
M23	Newman-Griffis et al. [70]	GloVe WE model trained on PubMed
M24	BioConceptVec$_{GloVe}$ [71]	GloVe We model trained on PubMed
M25	BioWordVec$_{int}$ [22]	FastText [23] WE model trained on PubMed + MeSH
M26	BioWordVec$_{ext}$ [22]	FastText [23] trained on PubMed + MeSH
M27	BioNLP2016$_{win2}$ [114]	FastText [23] WE model based on skip-gram and trained on PubMed with training setup detailed in [114, table 18]
M28	BioNLP2016$_{win30}$ [114]	FastText [23] WE model based on skip-gram and trained on PubMed with training setup detailed in [114, table 18]
M29	BioConceptVec$_{fastText}$ [71]	FastText [23] WE model trained on PubMed
M30	Universal Sentence Encoder (USE) [115]	USE SE pre-trained model of Cer et al. [115]
M31	BioSentVec [25]	sent2vec [26] SE model trained on PubMed + MIMIC-III
M32	FastText-Skipgram-BioC (this work)	FastText [23] WE model based on Skip-gram and trained on PMC-BioC corpus (05,09,2019) with the following setup: vector dim. = 200, learning rate = 0.05, sampling thres. = 1e-4, and negative examples = 10

https://doi.org/10.1371/journal.pone.0248663.t003

Table 4. Detailed setup for the sentence similarity methods based on pre-trained language models which will be evaluated in this work.

ID	Sentence similarity method	Detailed setup of each method
M33	BioBERT Base 1.0 [31] (+ PubMed)	BERT [33] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M34	BioBERT Base 1.0 [31] (+ PMC)	BERT [33] trained on English Wikipedia + BooksCorpus + PMC full-text articles
M35	BioBERT Base 1.0 [31] (+ PubMed + PMC)	BERT [33] trained on English Wikipedia + BooksCorpus + PubMed abstracts + PMC full-text articles
M36	BioBERT Base 1.1 [31] (+ PubMed)	BERT [33] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M37	BioBERT Large 1.1 [31] (+ PubMed)	BERT [33] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M38	NCBI-BlueBERT Base [32] PubMed	BERT [33] trained on PubMed abstracts
M39	NCBI-BlueBERT Large [32] PubMed	BERT [33] trained on PubMed abstracts
M40	NCBI-BlueBERT Base [32] PubMed + MIMIC-III	BERT [33] trained on PubMed abstracts + MIMIC-III
M41	NCBI-BlueBERT Large [32] PubMed + MIMIC-III	BERT [33] trained on PubMed abstracts + MIMIC-III
M42	SciBERT [79]	BERT [33] trained on PubMed abstracts
M43	ClinicalBERT [116]	BERT [33] trained on PubMed abstracts
M44	PubMedBERT [80] (abstracts)	BERT [33] trained on PubMed abstracts
M45	PubMedBERT [80] (abstracts + full text)	BERT [33] trained on PubMed abstracts + full text
M46	ouBioBERT-Base [81] (Uncased)	BERT [33] trained on PubMed abstracts

https://doi.org/10.1371/journal.pone.0248663.t004

Selection of language pre-processing methods and tools

The pre-processing stage aims to ensure a fair comparison of the methods that will be evaluated in a single end-to-end pipeline. To achieve this later goal, the pre-processing stage normalizes and decomposes the sentences into a series of components that evaluate the same sequence of words applied to all the methods simultaneously. The selection criteria of the pre-processing components have been conditioned by the following constraints: (a) the pre-processing methods and tools used by state-of-the-art methods; and (b) the availability of resources and software tools.

Most methods receive as input a sequence of words making up the sentence to be evaluated. The process of splitting sentences into words can be carried out by tokenizers for all the methods to be evaluated in this work, such as the well-known general domain Stanford CoreNLP tokenizer [117], which is used by Blagec et al. [28], or the biomedical domain BioCNLPTokenizer [118]. On the other hand, the use of lexicons instead of tokenizers for sentence splitting would be inefficient because of the vast general and biomedical vocabulary. Besides, there would not be possible to provide a fair comparison of the methods because the pre-trained language models have no identical vocabularies.

The tokenized words that conform the sentence, named tokens, are usually pre-processed by removing special characters and lower-casing, and removing the stop words. To analyze all the possible combinations of token pre-processing configurations from the literature, for each method we will replicate the methods used by other authors, such as Blagec et al. [28] and Sogancioglu et al. [20], and we will also evaluate all the pre-processing configurations that have not been evaluated yet. We will also study the impact of pre-processing configurations by not removing special characters nor lower casing and not removing the stop words from the tokens.

Ontology-based sentence similarity methods estimate the similarity of a sentence by exploiting the 'is-a' relations between the concepts in an ontology. Therefore, the evaluation of any ontology-based method in this work will receive a set of concept-annotated pairs of sentences. The aim of the biomedical Named Entity Recognizers (NER) is to identify entities in pieces of raw text, such as diseases or drugs. In this work, we propose to evaluate the impact of three significant biomedical NER tools on the sentence similarity task, as follows: (a) MetaMap [107], (b) cTAKES [108], and (c) MetaMap Lite [119]. MetaMap tool [107] is used by UBSM and COM methods [20] for recognizing Unified Medical Language System (UMLS) [120] concepts in the sentences, which is the standard compendium of biomedical vocabularies. In this work, we will use the default configuration of MetaMap, using all the available semantic types, the MedPost Part-of-speech tagger [121] and with the MetaMap Word-Sense Disambiguation (WSD) module, but restricting UMLS sources to SNOMED-CT and MeSH, which are currently implemented by HESML V1R5 [122]. We will also evaluate cTAKES [108], which has demonstrated to be a robust and reliable tool to recognize biomedical entities [123]. Encouraged by the high computational cost of MetaMap in evaluating large text corpus, Demner-Fushman et al. [119] introduce a lighter MetaMap version, called Metamap Lite, which provides a real-time implementation of the basic MetaMap annotation capabilities without a large degradation of its performance.

Software integration and contingency plan

To mitigate the impact of potential development risks or unexpected barriers, we have elaborated a contingency plan based on identifying potential risk sources, as well as the testing and integration prototyping of all third-party software components shown in Fig 2. Next, we detail the main risk sources identified in our contingency analysis and the actions carried out to mitigate their impact on our study.

1. *Integration of the biomedical ontologies and thesaurus*. Recently published HESML V1R5 software library [122] integrates the real-time evaluation of ontology-based similarity measures based on MeSH [24] and SNOMED-CT [67], as well as any other biomedical ontology based on the OBO file format [124]. Thus, this risk has been completely mitigated.

2. *External NER tools*. We have confirmed the feasibility of integrating all biomedical NER tools considered in our experiments, such as MetaMap [107] or cTAKES [108], by prototyping the main functions for annotating testing sentences.

3. *Availability of the pre-trained models*. We have already gathered all the pre-trained embeddings [22, 25, 70, 71, 73, 77, 114, 115] and BERT-based language models [31, 32, 79–81, 116] required for our experiments. We have also checked the validity of all pre-trained model files by testing the evaluation of the models using the third-party libraries as detailed below.

4. *Evaluation of the pre-trained models*. The software replication required to evaluate sentence embeddings and language models is extremely complex and out of the scope of this work. For this reason, these models must be evaluated by using the software artifacts used to generate the aforementioned models. Our strategy is to implement Python wrappers for evaluating the available models by using the provided software artifacts as follows: (1) Sent2vec-based models [25] will be evaluated using the Sent2vec library [26]; (2) Flair models [77] will be evaluated using the flairNLP framework [77]; and USE models [115] will be evaluated using the open source platform TensorFlow [125]. All BERT-based pre-trained models will be evaluated using the open-source bert-as-a-service library [126]. On the

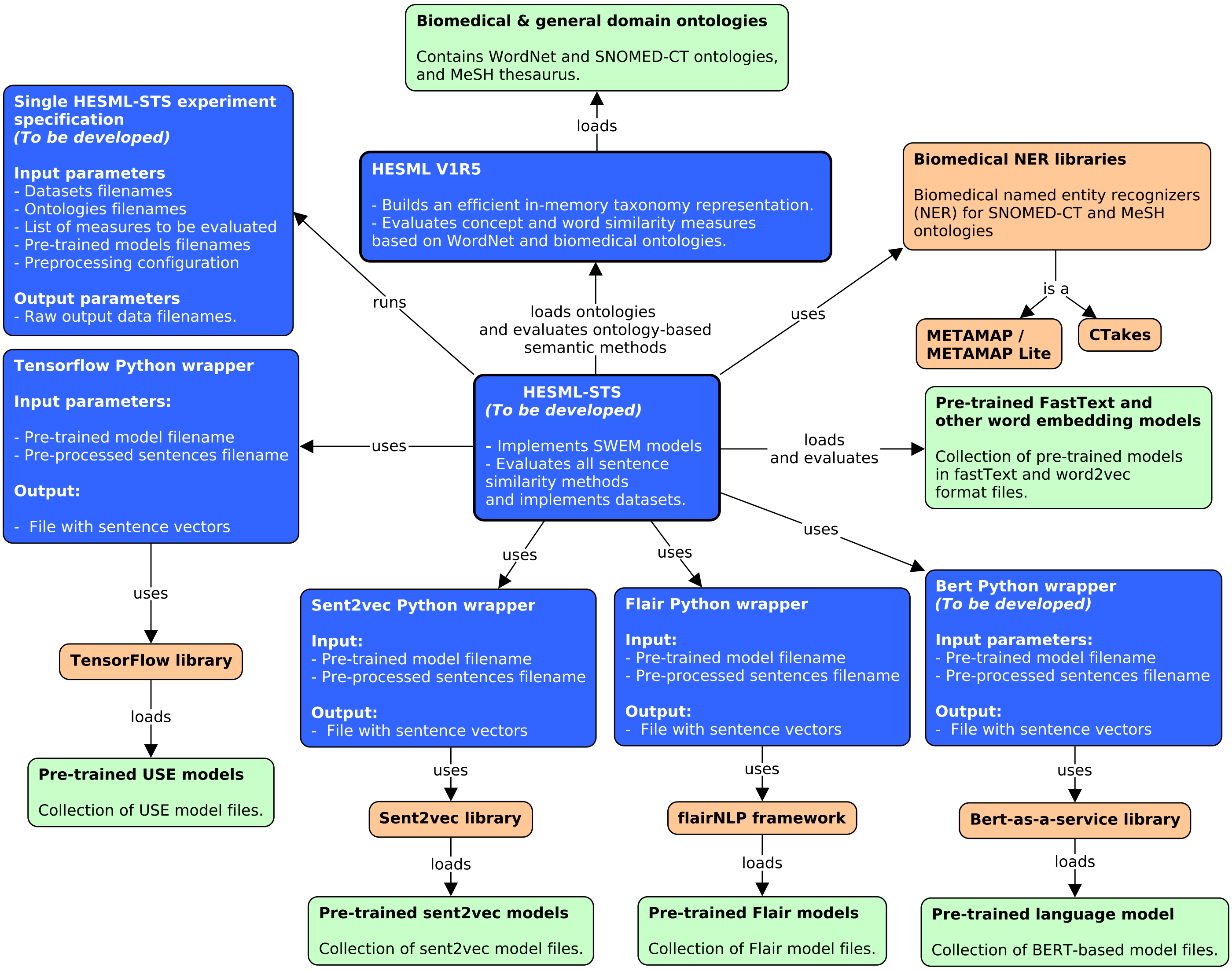

Biomedical & general domain ontologies
Contains WordNet and SNOMED-CT ontologies, and MeSH thesaurus.
loads
Single HESML-STS experiment specification
(To be developed)
Input parameters
- Datasets filenames
- Ontologies filenames
- List of measures to be evaluated
- Pre-trained models filenames
- Preprocessing configuration
Output parameters
- Raw output data filenames.
HESML V1R5
- Builds an efficient in-memory taxonomy representation.
- Evaluates concept and word similarity measures based on WordNet and biomedical ontologies.
Biomedical NER libraries
Biomedical named entity recognizers (NER) for SNOMED-CT and MeSH ontologies
is a
METAMAP / METAMAP Lite
CTakes
runs
loads ontologies and evaluates ontology-based semantic methods
uses
Tensorflow Python wrapper
Input parameters:
- Pre-trained model filename
- Pre-processed sentences filename
Output:
- File with sentence vectors
uses
HESML-STS
(To be developed)
- Implements SWEM models
- Evaluates all sentence similarity methods and implements datasets.
loads and evaluates
Pre-trained FastText and other word embedding models
Collection of pre-trained models in fastText and word2vec format files.
uses
uses
uses
uses
TensorFlow library
loads
Pre-trained USE models
Collection of USE model files.
Sent2vec Python wrapper
Input:
- Pre-trained model filename
- Pre-processed sentences filename
Output:
- File with sentence vectors
uses
Sent2vec library
loads
Pre-trained sent2vec models
Collection of sent2vec model files.
Flair Python wrapper
Input:
- Pre-trained model filename
- Pre-processed sentences filename
Output:
- File with sentence vectors
uses
flairNLP framework
loads
Pre-trained Flair models
Collection of Flair model files.
Bert Python wrapper
(To be developed)
Input parameters:
- Pre-trained model filename
- Pre-processed sentences filename
Output:
- File with sentence vectors
uses
Bert-as-a-service library
loads
Pre-trained language model
Collection of BERT-based model files.

Fig 2. Concept map detailing the external software components that will be integrated in HESML-STS. Input data files are shown in green, whilst external software libraries are shown in orange, and software components that will be developed are shown in blue. All experiments will be specified into a single experiment file, which is executed by the HESMLSTSclient program.

https://doi.org/10.1371/journal.pone.0248663.g002

other hand, we will develop a parser for efficiently loading and evaluating FastText-based [23] and other word embedding models [22, 70, 71, 73, 114] in the HESML-STS library that will be specially developed for this work. Finally, we have developed all the necessary prototypes to confirm the feasibility of evaluating all the pre-trained models considered in our experiments.

5. *Licensing restrictions*. The licensing restrictions of third-party software components and resources, such as SNOMED-CT [103], MeSH [24] and MetaMap [107], require users to obtain previously a license from the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH. Users will be able to reproduce the experiments of this work by following two alternatives: (1) downloading the third-party software components and integrating them in the

HESML-STS framework as will be detailed in our reproducibility protocol; or (2) by downloading a Docker image file which will contain a pre-installed version of all the necessary software for reproducing our experiments. In the first case, we will publish all the necessary source code, binaries, data, and documentation in Github and Dataverse repositories, to allow the user to integrate restricted third-party software components into the HESML-STS framework. In the second case, users must send a copy of their NLM license to "eciencia@consorciomadrono.es" to obtain the password to decrypt the Docker file provided as supplementary material.

Detailed workflow of our experiments

Fig 3 shows the workflow for running the experiments that will be carried out for this work. Given an input dataset, such as BIOSSES [20], MedSTS [49], or CTR [50], the first step is to pre-process all of the sentences, as shown in Fig 4. For each sentence in the dataset (named S1 and S2), the preprocessing phase will be divided into four stages as follows: (1.a) named entity recognition of UMLS [120] concepts, using different state-of-the-art NER tools, such as MetaMap [107] or cTAKES [108]; (1.b) tokenize the sentence, using well-known tokenizers, such as the Stanford CoreNLP tokenizer [117], BioCNLPTokenizer [118], or WordPieceTokenizer [33] for BERT-based methods; (1.c) lower-case normalization; (1.d) character filtering, which

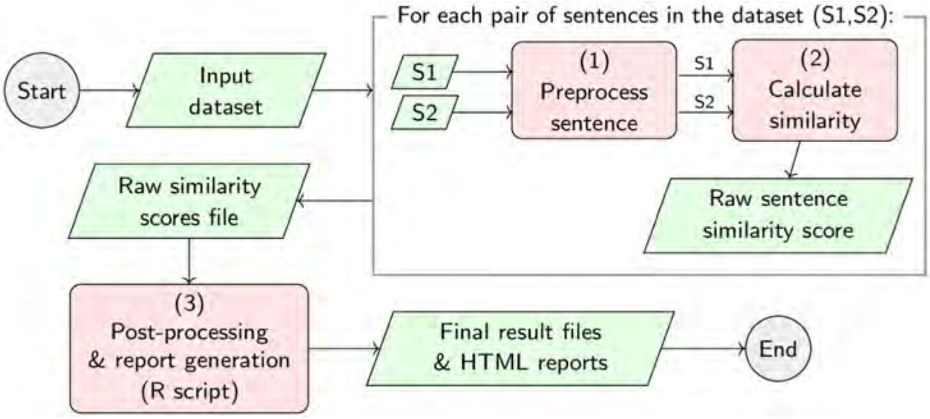

For each pair of sentences in the dataset (S1,S2):
Start
Input dataset
S1
S2
(1) Preprocess sentence
S1
S2
(2) Calculate similarity
Raw sentence similarity score
Raw similarity scores file
(3) Post-processing & report generation (R script)
Final result files & HTML reports
End

Fig 3. Detailed experimentation workflow which will be implemented by our experiments to preprocess, calculate the raw similarity scores, and post-process the results contained in the evaluation of the biomedical datasets. The workflow detailed below produces a collection of raw and processed data files.

https://doi.org/10.1371/journal.pone.0248663.g003

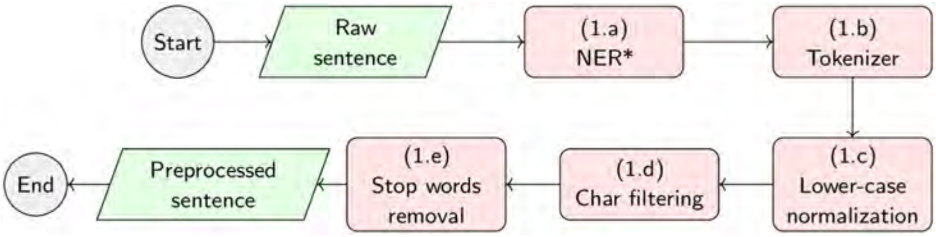


Fig 4. Detailed sentence preprocessing workflow that will be implemented in our experiments. The preprocessing stage takes an input sentence and produces a preprocessed sentence as output. (*) The named entity recognizer will be only evaluated in ontology-based methods.

https://doi.org/10.1371/journal.pone.0248663.g004

allows the removal of punctuation marks or special characters; and finally, (1.e) the removal of stop-words, following different approximations evaluated by other authors like Blagec et al. [28] or Sogancioglu et al. [20]. Once the dataset is pre-processed in step 1 detailed in Fig 3), the aim of step 2 is to calculate the similarity between each pair of sentences in the dataset to produce a raw output file containing all raw similarity scores, one score per sentence pair. Finally, a R-language script will be used in step 3 to process the raw similarity files and produce the final human-readable tables reporting the Pearson and Spearman correlation values detailed in Table 8, as well as the statistical significance of the results and any other supplementary data table required by our study on the impact of the pre-processing and NER tools.

Finally, we will also evaluate all the pre-processing combinations for each family of methods to study the impact of pre-processing methods on the performance of the sentence similarity methods results, with the only exception of the BERT-based methods. The pre-processing configurations of the BERT-based methods will only be evaluated in combination with the Word-Piece Tokenizer [33] because it is required by the current BERT implementations.

Evaluation metrics

The evaluation metrics used in this work are the Pearson correlation factor, denoted by r in Eq (1), and the Spearman rank correlation factor, denoted by ρ in Eq (2). The Pearson correlation is invariant regarding any scaling of the data, and it evaluates the linear relationship between two random samples, whilst the Spearman rank correlation is rank-invariant and evaluates the monotonic relationship between two random samples.

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{1}$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \qquad di = (x_i - y_i) \tag{2}$$

The use of the Pearson correlation to evaluate the task on sentence similarity can be traced back to the pioneering work of Dustin and Alfonsin [127]. On the other hand, both Pearson and Spearman correlation scores have been extensively used to compare the performance of the state-of-the-art methods on biomedical sentence similarity in most works in this line of research [20, 22, 28, 35]. Both aforementioned correlation metrics are also the standard metric for evaluating the task on word similarity [45]. For this reason, we use both aforementioned metrics to evaluate and compare the performance of the methods evaluated herein. However, Spearman's rank correlation has demonstrated to be more reliable in the evaluation of semantic similarity measures of sentences or words in different applications, because it is rank-invariant, and thus, it "provides an evaluation metric that is independent of such data-dependent transformations" [128].

We will use the well-known t-Student test to carry-out a statistical significance analysis of the results in the BIOSSES [20], MedSTS$_{full}$ [49], and CTR [50] datasets. In order to compare the performance of the semantic measures that will be evaluated in our experiments, we use the overall average values of the two aforementioned metrics in all datasets. The statistical significance of the results will be evaluated using the p-values resulting from the t-student test for the mean difference between the values reported by each pair of semantic measures in all datasets, or a subset of them relevant in the context of the discussion. The t-student test is used herein because it is a standard and widely-used hypothesis testing for small and independent data samples with the normal distribution. The p-values are computed using a one-sided t-student distribution on two paired random sample sets. Our null hypothesis, denoted by H_0, is that the difference in the average performance between each pair of compared sentence similarity methods is 0, whilst the alternative hypothesis, denoted by H_1, is that their average performance is different. For a 5% level of significance, it means that if the p-value is greater or equal than 0.05, we must accept the null hypothesis. Otherwise, we can reject H_0 with an error probability of less than the p-value. In this latter case, we will say that a first sentence similarity method obtains a statistically significantly higher value than the second one in a specific metric or that the former one significantly outperforms the second one.

Software implementation and development plan

Fig 5 shows a concept map detailing the planned experimental setup to run all experiments planned in this work, as detailed in Table 8. Our experiments will be based on our implementation and evaluation of all methods detailed in Tables 1–4 into a common and new Java software library called HESML-STS, which will be specifically developed for this work. HESML-STS will be based on an extension of the recent HESML V1R5 [122] semantic measures library for the biomedical domain.

All our experiments will be generated by running the *HESMLSTSclient* program shown in Fig 5 with a reproducible XML-based benchmark file, which will generate a raw output file in comma-separated file format (*.csv) for each dataset detailed in Table 5. The raw output files will contain the raw similarity values returned by each sentence similarity method in the evaluation of the degree of similarity between each sentence pair. The final results for the Pearson and Spearman correlation values planned in Table 8 will be automatically generated by running a R-language script file on the collection of raw similarity files using either R or RStudio statistical programs.

Table 6 shows the development plan schedule proposed for this work. We have decomposed the work into seven task groups, called Work Packages (WP), whose deliverables are as follows: (1) Python-based wrappers for the integration of the third-party software components (see Fig 2); (2) HESML-STS library beta 1 version integrated on top of HESML V1R5 (https://github.com/jjlastra/HESML) [122]; (3) HESML-STS beta 1 with an integrated end-to-end pipeline and the XML-based experiment engine; (4) collection of raw output data files generated by running the XML-based reproducible experiments; (5) detailed analysis of the results, including the identification of the main drawbacks and limitations of current methods; (6) reproducible protocol and dataset published in the Spanish Dataverse repository; and finally, (7) submission of the manuscript introducing the study that implements the protocol detailed herein, together with a companion data article introducing our reproducibility protocol and dataset.

Reproducing our benchmarks

For the sake of reproducibility, we will co-submit a companion data paper with the next work reporting the results of this study, which will introduce a publicly available reproducibility dataset, together with a detailed reproducibility protocol to allow the exact replication of all our experiments and results. Table 7 details the reproducibility software and data that will be published with our next work implementing this registered report. Our benchmarks will be implemented using Java and R languages and could be reproduced in any Java-complaint or Docker-complaint platforms, such as Windows, MacOS, or any Linux-based system. The available software and data will be published on the Spanish Dataverse Network.

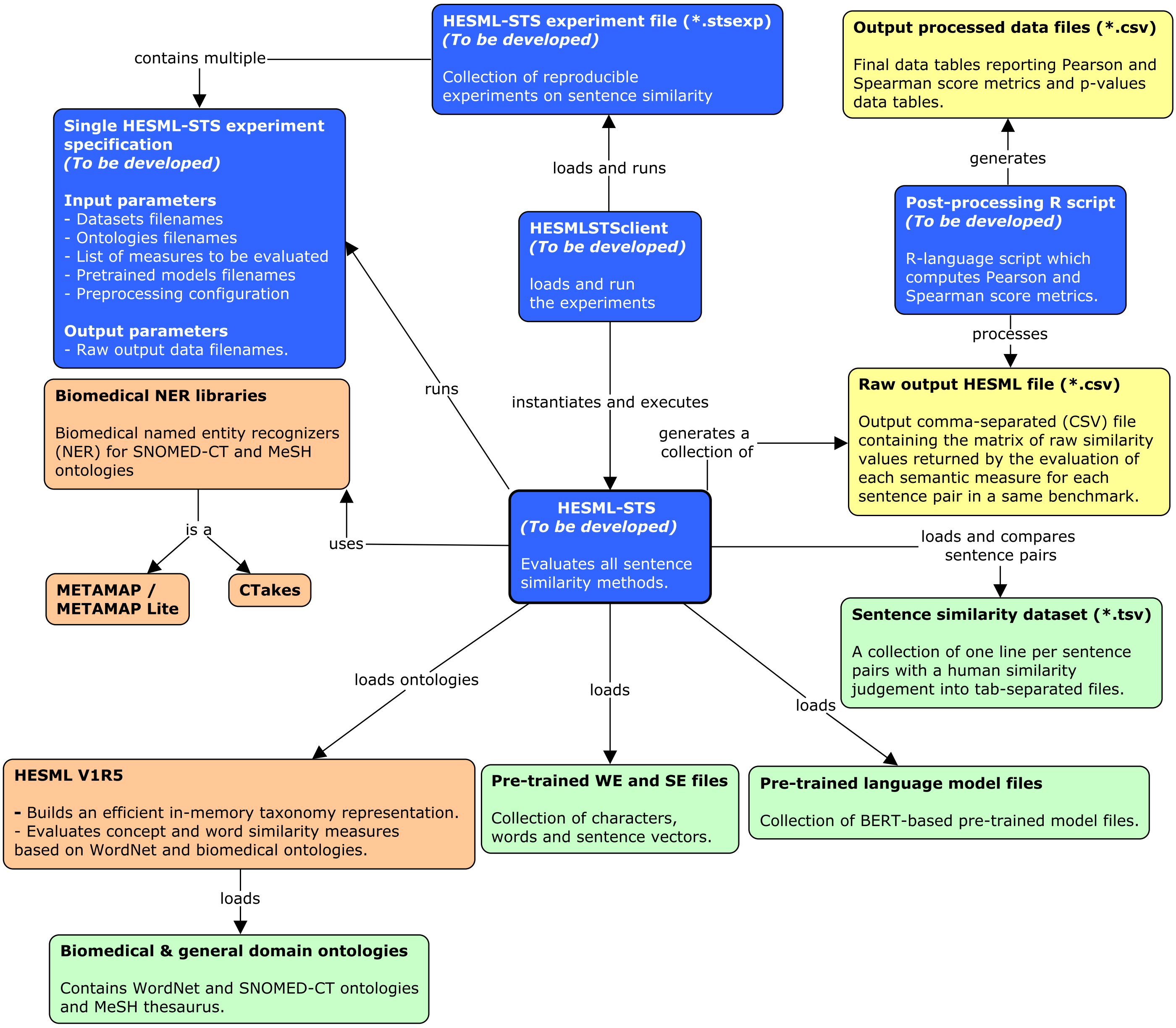

HESML-STS experiment file (*.stsexp)
(To be developed)
Collection of reproducible experiments on sentence similarity
Output processed data files (*.csv)
Final data tables reporting Pearson and Spearman score metrics and p-values data tables.
contains multiple
Single HESML-STS experiment specification
(To be developed)
Input parameters
- Datasets filenames
- Ontologies filenames
- List of measures to be evaluated
- Pretrained models filenames
- Preprocessing configuration
Output parameters
- Raw output data filenames.
loads and runs
HESMLSTSclient
(To be developed)
loads and run the experiments
generates
Post-processing R script
(To be developed)
R-language script which computes Pearson and Spearman score metrics.
processes
runs
instantiates and executes
Raw output HESML file (*.csv)
Output comma-separated (CSV) file containing the matrix of raw similarity values returned by the evaluation of each semantic measure for each sentence pair in a same benchmark.
generates a collection of
Biomedical NER libraries
Biomedical named entity recognizers (NER) for SNOMED-CT and MeSH ontologies
is a
uses
HESML-STS
(To be developed)
Evaluates all sentence similarity methods.
loads and compares sentence pairs
METAMAP / METAMAP Lite
CTakes
Sentence similarity dataset (*.tsv)
A collection of one line per sentence pairs with a human similarity judgement into tab-separated files.
loads ontologies
loads
loads
HESML V1R5
- Builds an efficient in-memory taxonomy representation.
- Evaluates concept and word similarity measures based on WordNet and biomedical ontologies.
Pre-trained WE and SE files
Collection of characters, words and sentence vectors.
Pre-trained language model files
Collection of BERT-based pre-trained model files.
loads
Biomedical & general domain ontologies
Contains WordNet and SNOMED-CT ontologies and MeSH thesaurus.

Fig 5. Concept map detailing the software architecture for our experimental setup. Input data files are shown in green, whilst output raw and processed data files are shown in yellow, external available software libraries in orange, and software components that will be developed are shown in blue. All experiments will be specified into a single experiment file, which is executed by the HESMLSTSclient program.

https://doi.org/10.1371/journal.pone.0248663.g005

Table 5. Benchmarks on biomedical sentence similarity evaluated in this work.

Dataset	#pairs	Corresponding file (*.tsv) in future HESML-STS distribution
BIOSSES [20]	100	BIOSSESNormalized.tsv
MedSTS [49]	1,068	CTRNormalized_averagedScore.tsv
CTR [50]	170	MedStsFullNormalized.tsv

https://doi.org/10.1371/journal.pone.0248663.t005

Table 6. Development plan proposed for this work.

Definition of the workpackages and tasks to be developed	Workload (weeks)
WP1—Implementation of Python wrappers for third-party components	
Task 1.1 Implementation of the BERT Python wrapper	1
Task 1.2 Implementation of the Sent2vec, Tensorflow, and Flair wrappers	1
WP2—Software implementation of methods	
Task 2.1 Implementation of all pre-processing methods shown in Fig 6	2
Task 2.2 Implementation of string-based methods detailed in Table 1	1
Task 2.3 Implementation of ontology-based methods detailed in Table 2	1
Task 2.4 Implementation of WE and SE methods detailed in Table 3	1
Task 2.5 Implementation of BERT-based methods detailed in Table 4	1
WP3—Implementation of the automatic reproducible experiments	
Task 3.1 Implementation of the benckmark objects and file parsers	1
Task 3.2 Preparation of the experiment files to evaluate the impact of the pre-processing configurations	1
Task 3.3 Preparation of the experiment files to evaluate the performance of the methods in the three biomedical sentence similarity datasets	1
WP4—Evaluation of the entire set of reproducible experiments	
Task 4.1 Execution of the pre-processing experiments to generate of all raw output data	4
Task 4.2 Execution of the method experiments and generation of all raw output data	2
WP5—Data analysis and results interpretation	
Task 5.1 Design and development of the post-processing scripts for the generation of tables and figures	2
Task 5.2 Data analysis and discussion	2
Task 5.3 Identification and analysis of the main drawbacks and limitations of current methods	3
WP6—Design and publication of the reproduciblity protocol and dataset	
Task 6.1 Design and validation of the reproducibility dataset	1
Task 6.2 Design of the reproducibility protocol	1
Task 6.3 Private publication and validation of the reproducibility dataset	1
Task 6.4 Software release of the first HESML-STS version	1
Task 6.5 Creation and validation of the Docker file	1
Task 6.6 Writing and testing of the reproducibility protocol	2
Task 6.7 Writing of the companion data article introducing our reproducibility protocol and dataset	2
WP8—Publishing the results	
Task 8.1 Writing and submission of the research article reporting the results of this study and co-submission of the companion data article	6
Overall estimated workload (weeks)	39

https://doi.org/10.1371/journal.pone.0248663.t006

Detailed results planned

Table 8 shows the methods and datasets that will be evaluated in this work, together with the detailed results which will be generated by our experiments. Finally, any further experimental results resulting from our study on the impact of the pre-processing and NER tools on the performance of the sentence similarity methods will also be reported in our next work, and they could also be reproduced using our aforementioned reproducibility resources.

Answering our research questions

Next, we explain how our experimental results will allow answering every of our research questions:

Table 7. Detailed planning of the supplementary reproducibility software and data that will be published with our future work implementing this registered report.

Material	Description
Reproducibility dataset	Contains all raw input and output data files, pre-trained model files, and a long-term reproducibility image based on ReproZip or Docker, which will be publicly available in the Spanish Dataverse Network.
Companion data article	Data and methods article introducing our reproducibility protocol and dataset to allow the independent replication of our experiments and results.
HESML-STS software library	Release of the new HESML-STS library. This library will be integrated into a forthcoming HESML version published both in Github and the Spanish Dataverse Network under CC By-NC-SA-4.0 license.
HESML-STS software paper	Software article introducing our sentence similarity library, called HESML-STS, which will be especially developed for this work.

https://doi.org/10.1371/journal.pone.0248663.t007

RQ1. Table 8 will report the Pearson and the Spearman rank correlation factors in the evaluation of the three datasets. Therefore, we will draw up our conclusions by comparing the performance of both metrics. However, we will set the best overall performing methods using the Spearman correlation results because of its better predictive nature in most extrinsic tasks, as pointed out in section "Evaluation Metrics".

RQ2. We will use a t-Student test between the Spearman correlation values obtained by each pair of methods in the evaluation of the three proposed datasets as a means to set the statistical significance of the results. Thus, we will say that a method significantly outperforms another one resulting p-values are less or equal than 0.05. The t-Student test will be based on the Spearman rank correlation value for the same reasons detailed above.

RQ3. Table 9 details the methods and biomedical NER tools that will be evaluated in this work. We will consider only ontology-based methods since word and sentence pre-trained models have been trained on raw texts and do not contain UMLS concepts. To make a fair comparison of the methods, we will evaluate them using the best pre-processing configuration defined by a selection of the tokenizer, lower-case normalization, char filtering, and stop words list. Our analysis and discussion of the results will be based on comparing the Pearson and Spearman correlation values reported for each method. However, we will set the best overall performing NER tool using the Spearman rank correlation results like the remaining research questions.

RQ4. Fig 6 details all the possible combinations of pre-processing configurations that will be evaluated in this work. String, word and sentence embedding, and ontology-based methods, will be evaluated using all the available configurations except the WordPiece-Tokenizer [33], which is specific to BERT-based methods. Thus, BERT-based methods will be evaluated using different char filtering, lower casing normalization, and stop words removal configurations. We will use the Pearson and Spearman's correlation values to determine the impact of the different pre-processing configurations on the evaluation results. However, we will set the best overall performing pre-processing configuration using the Spearman rank correlation results like the remaining research questions.

RQ5. Our methodology for identifying the main drawbacks and limitations is based on the following steps: (1) analyzing evaluated methods and tools; (2) identifying which methods do not perform well in the datasets; (3) searching and analyzing the sentence pairs

Table 8. Pearson (r) and Spearman (ρ) correlation values (0.xxx) which will be obtained in our experiments from the evaluation of all sentence similarity methods detailed below in the BIOSSES [20], MedSTS$_{full}$ [49], and CTR [50] datasets.

ID	Sentence similarity methods	BIOSSES		MedSTS$_{full}$		CTR	
		r	ρ	r	ρ	r	ρ
M1	Qgram	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M2	Jaccard	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M3	Block distance	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M4	Levenshtein distance [57]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M5	Overlap coefficient [60]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M6	WBSM-Rada [20, 111]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M7	WBSM-J&C [20, 112]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M8	WBSM-cosJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M9	WBSM-coswJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M10	WBSM-Cai [20, 110]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M11	UBSM-Rada [20, 111]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M12	UBSM-J&C [20, 112]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M13	UBSM-cosJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M14	UBSM-coswJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M15	UBSM-Cai [20, 110]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M16	COM [20]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M17	Flair [37, 77]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M18	Pyysalo et al. [73]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M19	BioConceptVec$_{word2vec\_sg}$	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M20	BioConceptVec$_{word2vec\_cbow}$	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M21	Newman-Griffis$_{word2vec\_sg}$ [70]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M22	Newman-Griffis$_{word2vec\_cbow}$ [70]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M23	Newman-Griffis$_{glove}$	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M24	BioConceptVec$_{glove}$ [71]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M25	BioWordVec$_{int}$ [22]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M26	BioWordVec$_{ext}$ [22]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M27	BioNLP2016$_{win2}$ [114]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M28	BioNLP2016$_{win30}$ [114]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M29	BioConceptVec$_{fastText}$	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M30	USE [115]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M31	BioSentVec (PubMed+MIMIC-III)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M32	FastText-SkGr-BioC (this work)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M33	BioBERT Base 1.0 (+ PubMed)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M34	BioBERT Base 1.0 (+ PMC)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M35	BioBERT Base 1.0 (+ PubMed + PMC)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M36	BioBERT Base 1.1 (+ PubMed)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M37	BioBERT Large 1.1 (+ PubMed)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M38	NCBI-BlueBERT Base PubMed	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M39	NCBI-BlueBERT Large PubMed	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M40	NCBI-BlueBERT Base PubMed + MIMIC-III	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M41	NCBI-BlueBERT Large PubMed + MIMIC-III	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M42	SciBERT	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M43	ClinicalBERT	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M44	PubMedBERT (abstracts)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M45	PubMedBERT (abstracts + full text)	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M46	ouBioBERT-Base, Uncased	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx

https://doi.org/10.1371/journal.pone.0248663.t008

Table 9. Pearson (r) and Spearman (ρ) correlation values (0.xxx) which will be obtained in our experiments from the evaluation of ontology similarity methods detailed below in the MedSTS$_{full}$ [49] dataset for each NER tool.

ID	Methods	MetaMap		MetaMap Lite		cTAKES	
		r	ρ	r	ρ	r	ρ
M11	UBSM-Rada [20, 111]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M12	UBSM-J&C [20, 112]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M13	UBSM-cosJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M14	UBSM-coswJ&C [20, 43, 109]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M15	UBSM-Cai [20, 110]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx
M16	COM [20]	.xxx	.xxx	.xxx	.xxx	.xxx	.xxx

https://doi.org/10.1371/journal.pone.0248663.t009

in which the methods report the largest differences from the gold standard; and finally, (4) analyzing and hypothesizing why the methods fail. We have already identified some of the drawbacks of several methods during our literature review and prototyping stage as follows. First, most methods reported in the literature neither consider the structure of the sentences nor the intrinsic relations between the parts that conform them. Second, BERT-based methods are trained for downstream tasks, using a supervised approach, and do not perform well in an unsupervised context. Finally, we expect to find drawbacks and limitations by analyzing and studying the results.

Conclusions and future work

We have introduced a detailed experimental setup to reproduce, evaluate, and compare the most extensive set of methods on biomedical sentence similarity reported in the literature,

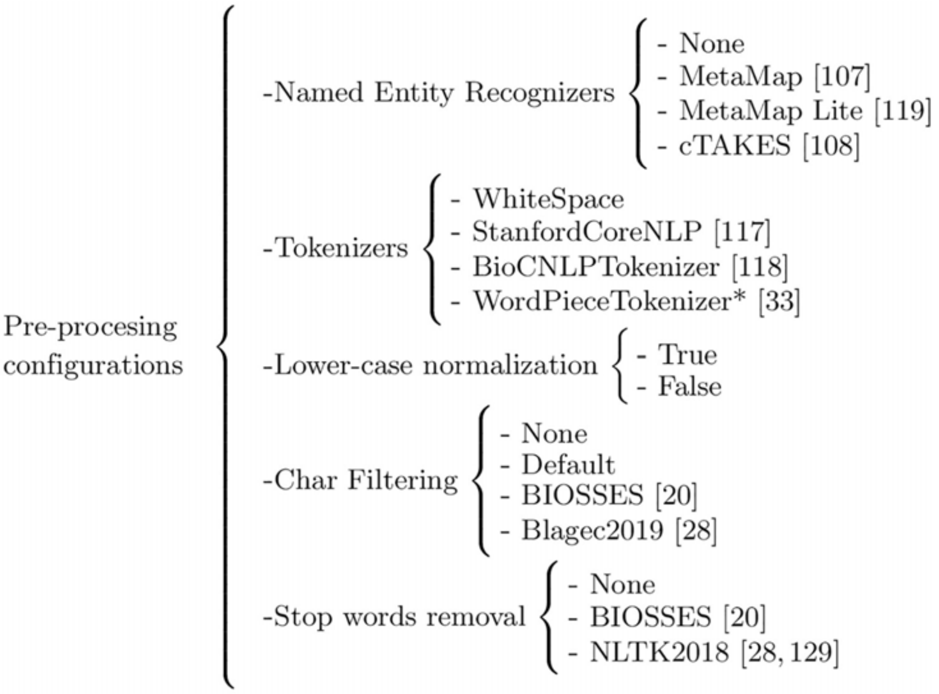


Fig 6. Details of the pre-processing configurations that will be evaluated in this work. (*) WordPieceTokenizer [33] will be used only for BERT-based methods. [20, 28, 33, 107, 108, 117–119, 129].

https://doi.org/10.1371/journal.pone.0248663.g006

with the following aims: (1) elucidating the state of the art on the problem, (2) studying the impact of different pre-processing configurations; (3) studying the impact of the NER tools; and (4) identifying the main drawbacks and limitations of the current methods to set new lines of research. Our work also introduces the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity based on the same software platform. In addition, we have proposed the evaluation of a new word embedding model based on FastText and trained on the full text of the articles in the PMC-BioC corpus [19], and the evaluation for the first time of the CTR [50] dataset.

All experiments introduced herein will be implemented into the same software library, called HESML-STS, which will be developed especially for this work. We will provide a detailed reproducibility protocol, together with a collection of software tools and a reproducibility dataset, to allow the exact replication of all our experiments, methods, and results. Thus, our reproducible experiments could be independently reproduced and extended by the research community, with the hope of becoming a de facto experimentation platform for this research line.

As forthcoming activities, we plan to evaluate the sentence similarity methods in an extrinsic task, such as semantic medical indexing [130] or summarization [131]. We also consider the evaluation of further pre-processing configurations, such as biomedical NER systems based on recent Deep Learning techniques [10], or extending our experiments and research to the multilingual scenario by integrating multilingual biomedical NER systems like Cimind [132]. Finally, we plan to evaluate some recent biomedical concept embeddings based on MeSH [133], which has not been evaluated in the sentence similarity task yet.

Acknowledgments

We are grateful to Gizem Sogancioglu and Kathrin Blagec for answering kindly our questions to replicate their methods and experiments. UMLS CUI codes, SNOMED-CT US ontology and MeSH thesaurus were used in our experiments by courtesy of the National Library of Medicine of the United States. Finally, we are grateful to the anonymous reviewers for their valuable comments to improve the quality of the paper.

Author Contributions

Conceptualization: Alicia Lara-Clares, Juan J. Lastra-Díaz, Ana Garcia-Serrano.

Formal analysis: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Funding acquisition: Ana Garcia-Serrano.

Investigation: Alicia Lara-Clares.

Methodology: Alicia Lara-Clares, Juan J. Lastra-Díaz, Ana Garcia-Serrano.

Resources: Alicia Lara-Clares.

Supervision: Juan J. Lastra-Díaz, Ana Garcia-Serrano.

Validation: Alicia Lara-Clares.

Visualization: Juan J. Lastra-Díaz.

Writing – original draft: Alicia Lara-Clares.

Writing – review & editing: Juan J. Lastra-Díaz, Ana Garcia-Serrano.

References

1. Tafti AP, Behravesh E, Assefi M, LaRose E, Badger J, Mayer J, et al. bigNN: An open-source big data toolkit focused on biomedical sentence classification. In: 2017 IEEE International Conference on Big Data (Big Data); 2017. p. 3888–3896.
2. Kim S, Kim W, Comeau D, Wilbur WJ. Classifying gene sentences in biomedical literature by combining high-precision gene identifiers. In: Proc. of the 2012 Workshop on Biomedical Natural Language Processing; 2012. p. 185–192.
3. Chen Q, Panyam NC, Elangovan A, Davis M, Verspoor K. Document triage and relation extraction for protein-protein interactions affected by mutations. In: Proc. of the BioCreative VI Workshop. vol. 6; 2017. p. 52–51.
4. Sarrouti M, Ouatik El Alaoui S. A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering. J Biomedical Informatics. 2017; 68:96–103. https://doi.org/10.1016/j.jbi.2017.03.001 PMID: 28286031
5. Kosorus H, Bögl A, Küng J. Semantic Similarity between Queries in QA System using a Domain-specific Taxonomy. In: ICEIS (1); 2012. p. 241–246.
6. Ravikumar KE, Rastegar-Mojarad M, Liu H. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database. 2017; 2017(1). https://doi.org/10.1093/database/baw156 PMID: 28365720
7. Rastegar-Mojarad M, Komandur Elayavilli R, Liu H. BELTracker: evidence sentence retrieval for BEL statements. Database. 2016; 2016. https://doi.org/10.1093/database/baw079 PMID: 27173525
8. Du J, Chen Q, Peng Y, Xiang Y, Tao C, Lu Z. ML-Net: multi-label classification of biomedical texts with deep neural networks. J Am Med Inform Assoc. 2019; 26(11):1279–1285. https://doi.org/10.1093/jamia/ocz085 PMID: 31233120
9. Liu H, Hunter L, Kešelj V, Verspoor K. Approximate subgraph matching-based literature mining for biomedical events and relations. PLoS One. 2013; 8(4):e60954. https://doi.org/10.1371/journal.pone.0060954 PMID: 23613763
10. Hahn U, Oleynik M. Medical Information Extraction in the Age of Deep Learning. Yearb Med Inform. 2020; 29(1):208–220. PMID: 32823318
11. Kim SN, Martinez D, Cavedon L, Yencken L. Automatic classification of sentences to support Evidence Based Medicine. BMC Bioinformatics. 2011; 12 Suppl 2:5. https://doi.org/10.1186/1471-2105-12-S2-S5 PMID: 21489224
12. Hassanzadeh H, Groza T, Nguyen A, Hunter J. A supervised approach to quantifying sentence similarity: with application to evidence based medicine. PLoS One. 2015; 10(6):e0129392. https://doi.org/10.1371/journal.pone.0129392 PMID: 26039310
13. Boyack KW, Newman D, Duhon RJ, Klavans R, Patek M, Biberstine JR, et al. Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. PLoS One. 2011; 6(3):e18029. https://doi.org/10.1371/journal.pone.0018029 PMID: 21437291
14. Dey S, Luo H, Fokoue A, Hu J, Zhang P. Predicting adverse drug reactions through interpretable deep learning framework. BMC Bioinformatics. 2018; 19(Suppl 21):476. https://doi.org/10.1186/s12859-018-2544-0 PMID: 30591036
15. Lamurias A, Ruas P, Couto FM. PPR-SSM: personalized PageRank and semantic similarity measures for entity linking. BMC Bioinformatics. 2019; 20(1):534. https://doi.org/10.1186/s12859-019-3157-y PMID: 31664891
16. Aliguliyev RM. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Syst Appl. 2009; 36(4):7764–7772. https://doi.org/10.1016/j.eswa.2008.11.022
17. Shang Y, Li Y, Lin H, Yang Z. Enhancing biomedical text summarization using semantic relation extraction. PLoS One. 2011; 6(8):e23862. https://doi.org/10.1371/journal.pone.0023862 PMID: 21887336
18. Allot A, Chen Q, Kim S, Vera Alvarez R, Comeau DC, Wilbur WJ, et al. LitSense: making sense of biomedical literature at sentence level. Nucleic Acids Res. 2019;. https://doi.org/10.1093/nar/gkz289 PMID: 31020319
19. Comeau DC, Wei CH, Islamaj Doğan R, Lu Z. PMC text mining subset in BioC: about three million full-text articles and growing. Bioinformatics. 2019;. https://doi.org/10.1093/bioinformatics/btz070 PMID: 30715220
20. Sogancioglu G, Öztürk H, Özgür A. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017; 33(14):49–58. https://doi.org/10.1093/bioinformatics/btx238 PMID: 28881973

21. Li Y, McLean D, Bandar ZA, James DO, Crockett K. Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Trans Knowl Data Eng. 2006; 18(8):1138–1150. https://doi.org/10.1109/TKDE.2006.130

22. Zhang Y, Chen Q, Yang Z, Lin H, Lu Z. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Sci Data. 2019; 6(1):52. https://doi.org/10.1038/s41597-019-0055-0 PMID: 31076572

23. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. MIT Press. 2017; 5:135–146.

24. Nelson SJ, Johnston WD, Humphreys BL. Relationships in Medical Subject Headings (MeSH). In: Bean CA, Green R, editors. Relationships in the Organization of Knowledge. Dordrecht: Springer Netherlands; 2001. p. 171–184.

25. Chen Q, Peng Y, Lu Z. BioSentVec: creating sentence embeddings for biomedical texts. In: 2019 IEEE International Conference on Healthcare Informatics (ICHI). IEEE; 2019. p. 1–5.

26. Pagliardini M, Gupta P, Jaggi M. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 528–540.

27. Johnson AEW, Pollard TJ, Shen L, Lehman LWH, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Sci Data. 2016; 3:160035. https://doi.org/10.1038/sdata.2016.35 PMID: 27219127

28. Blagec K, Xu H, Agibetov A, Samwald M. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. BMC Bioinformatics. 2019; 20(1):178. https://doi.org/10.1186/s12859-019-2789-2 PMID: 30975071

29. Le Q, Mikolov T. Distributed Representations of Sentences and Documents. In: International Conference on Machine Learning. Journal of Machine Learning Research; 2014. p. 1188–1196.

30. Kiros R, Zhu Y, Salakhutdinov RR, Zemel R, Urtasun R, Torralba A, et al. Skip-Thought Vectors. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems 28. Curran Associates; 2015. p. 3294–3302.

31. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2019; 36(4):1234–1240. https://doi.org/10.1093/bioinformatics/btz682

32. Peng Y, Yan S, Lu Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In: Proc. of the 18th BioNLP Workshop and Shared Task. Florence, Italy: Association for Computational Linguistics; 2019. p. 58–65.

33. Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein J, Doran C, Solorio T, editors. Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, (Long and Short Papers). Minneapolis, MN, USA: Association for Computational Linguistics; 2019. p. 4171–4186.

34. Kalyan KS, Sangeetha S. SECNLP: A survey of embeddings in clinical natural language processing. J Biomed Inform. 2020; 101:103323. https://doi.org/10.1016/j.jbi.2019.103323 PMID: 31711972

35. Khattak FK, Jeblee S, Pou-Prom C, Abdalla M, Meaney C, Rudzicz F. A survey of word embeddings for clinical text. Journal of Biomedical Informatics: X. 2019; 4:100057. https://doi.org/10.1016/j.yjbinx.2019.100057

36. Alsentzer E, Murphy J, Boag W, Weng WH, Jindi D, Naumann T, et al. Publicly Available Clinical BERT Embeddings. In: Proc. of the 2nd Clinical Natural Language Processing Workshop. Minneapolis, Minnesota, USA: Association for Computational Linguistics; 2019. p. 72–78.

37. Tawfik NS, Spruit MR. Evaluating Sentence Representations for Biomedical Text: Methods and Experimental Results. J Biomed Inform. 2020; p. 103396. https://doi.org/10.1016/j.jbi.2020.103396 PMID: 32147441

38. Chen Q, Du J, Kim S, Wilbur WJ, Lu Z. Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records. BMC Medical Informatics and Decision Making. 2020; 20(1):73. https://doi.org/10.1186/s12911-020-1044-0 PMID: 32349758

39. Breiman L. Random Forests. Machine Learning. 2001; 45(1):5–32. https://doi.org/10.1023/A:1010933404324

40. Lastra-Díaz JJ, Garcia-Serrano A, Batet M, Fernández M, Chirigati F. HESML: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems. 2017; 66:97–118. https://doi.org/10.1016/j.is.2017.02.002

41. Chirigati F, Rampin R, Shasha D, reire J. Reprozip: Computational reproducibility with ease. In: Proc. of the 2016 international conference on management of data. ACM Digital Libraries; 2016. p. 2085–2088.

42. Lastra-Díaz JJ, Garcia-Serrano A. A new family of information content models with an experimental survey on WordNet. Knowledge-Based Systems. 2015; 89:509–526. https://doi.org/10.1016/j.knosys.2015.08.019

43. Lastra-Díaz JJ, Garcia-Serrano A. A novel family of IC-based similarity measures with a detailed experimental survey on WordNet. Engineering Applications of Artificial Intelligence Journal. 2015; 46:140–153. https://doi.org/10.1016/j.engappai.2015.09.006

44. Lastra-Díaz JJ, Garcia-Serrano A. A refinement of the well-founded Information Content models with a very detailed experimental survey on WordNet. ETSI Informática. Universidad Nacional de Educación a Distancia (UNED). http://e-spacio.uned.es/fez/view/bibliuned:DptoLSI-ETSI-Informes-Jlastra-refinement; 2016. TR-2016-01.

45. Lastra-Diaz JJ, Goikoetxea J, Hadj Taieb MA, Garcia-Serrano A, Ben Aouicha M, Agirre E. A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence. 2019; 85:645–665. https://doi.org/10.1016/j.engappai.2019.07.010

46. Lastra-Díaz JJ, Garcia-Serrano A. WordNet-based word similarity reproducible experiments based on HESML V1R1 and ReproZip; 2016. Mendeley Data, v1. http://doi.org/10.17632/65pxgskhz9.1.

47. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb MA, Garcia-Serrano A, Aouicha MB, Agirre E. Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity. Data in Brief. 2019; 26:104432. https://doi.org/10.1016/j.dib.2019.104432 PMID: 31516953

48. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb M, Garcia-Serrano A, Ben Aouicha M, Agirre E, et al. A large reproducible benchmark of ontology-based methods and word embeddings for word similarity. Information Systems. 2021; 96:101636. https://doi.org/10.1016/j.is.2020.101636

49. Wang Y, Afzal N, Fu S, Wang L, Shen F, Rastegar-Mojarad M, et al. MedSTS: a resource for clinical semantic textual similarity. Language Resources and Evaluation. 2018; p. 1–16.

50. Lithgow-Serrano O, Gama-Castro S, Ishida-Gutiérrez C, Mejía-Almonte C, Tierrafría VH, Martínez-Luna S, et al. Similarity corpus on microbial transcriptional regulation. Journal of Biomedical Semantics. 2019; 10(1):8. https://doi.org/10.1186/s13326-019-0200-x PMID: 31118102

51. Lithgow-Serrano O, Gama-Castro S, Ishida-Gutiérrez C, Collado-Vides J. L-Regulon: A novel soft-curation approach supported by a semantic enriched reading for RegulonDB literature. bioRxiv. 2020;. https://doi.org/10.1101/2020.04.26.062745

52. Gerlach M, Shi H, Amaral LAN. A universal information theoretic approach to the identification of stopwords. Nature Machine Intelligence. 2019; 1(12):606–612. https://doi.org/10.1038/s42256-019-0112-6

53. Mishra MK, Viradiya J. Survey of Sentence Embedding Methods. International Journal of Applied Science and Computations. 2019; 6(3):592–592.

54. Babić K, Martinčić-Ipšić S, Meštrović A. Survey of Neural Text Representation Models. Information An International Interdisciplinary Journal. 2020; 11(11):511.

55. Jaccard P. Nouvelles recherches sur la distribution florale. Bull Soc Vaud sci nat. 1908; 44:223–270.

56. Manning CD, Manning CD, Schütze H. Foundations of Statistical Natural Language Processing. Online: MIT Press; 1999.

57. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10. Springer; 1966. p. 707–710.

58. Ukkonen E. Approximate string-matching with q-grams and maximal matches. Theor Comput Sci. 1992; 92(1):191–211. https://doi.org/10.1016/0304-3975(92)90143-4

59. Krause EF. Taxicab Geometry: An Adventure in Non-Euclidean Geometry. Online: Courier Corporation; 1986.

60. Lawlor LR. Overlap, Similarity, and Competition Coefficients. Ecology. 1980; 61(2):245–251. https://doi.org/10.2307/1935181

61. Jimenez S, Becerra C, Gelbukh A. Soft cardinality: A parameterized similarity function for text comparison. In: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proc. of the main conference and the shared task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). ACL; 2012. p. 449–453.

62. Wu H, Huang H, Lu W. Bit at semeval-2016 task 1: Sentence similarity based on alignments and vector with the weight of information content. In: Proc. of the 10th International Workshop on Semantic Evaluation (SemEval-2016). ACL; 2016. p. 686–690.

63. Wu H, Huang H, Jian P, Guo Y, Su C. BIT at SemEval-2017 Task 1: Using semantic information space to evaluate semantic textual similarity. In: Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017). ACL; 2017. p. 77–84.

64. Pawar A, Mago V. Challenging the Boundaries of Unsupervised Learning for Semantic Similarity. IEEE Access. 2019; 7:16291–16308. https://doi.org/10.1109/ACCESS.2019.2891692

65. Islam A, Inkpen D. Semantic Text Similarity Using Corpus-based Word Similarity and String Similarity. ACM Trans Knowl Discov Data. 2008; 2(2):10:1–10:25. https://doi.org/10.1145/1376815.1376819

66. Lee MC, Chang JW, Hsieh TC. A grammar-based semantic similarity algorithm for natural language sentences. ScientificWorldJournal. 2014; 2014:437162. https://doi.org/10.1155/2014/437162 PMID: 24982952

67. Shajalal M, Aono M. Semantic textual similarity between sentences using bilingual word semantics. Progress in Artificial Intelligence. 2019; 8(2):263–272. https://doi.org/10.1007/s13748-019-00180-4

68. Maharjan N, Banjade R, Gautam D, Tamang LJ, Rus V. DT_Team at SemEval-2017 Task 1: Semantic Similarity Using Alignments, Sentence-Level Embeddings and Gaussian Mixture Model Output. In: Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017). ACL; 2017. p. 120–124.

69. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP). ACL Web; 2014. p. 1532–1543.

70. Newman-Griffis D, Lai A, Fosler-Lussier E. Insights into Analogy Completion from the Biomedical Domain. In: BioNLP 2017. Vancouver, Canada,: Association for Computational Linguistics; 2017. p. 19–28.

71. Chen Q, Lee K, Yan S, Kim S, Wei CH, Lu Z. BioConceptVec: Creating and evaluating literature-based biomedical concept embeddings on a large scale. PLOS Computational Biology. 2020; 16(4):1–18. https://doi.org/10.1371/journal.pcbi.1007617 PMID: 32324731

72. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. arXiv. 2013;.

73. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. Distributional semantics resources for biomedical text processing. Proc of LBM. 2013; p. 39–44.

74. Kajiwara T, Bollegala D, Yoshida Y, Kawarabayashi KI. An iterative approach for the global estimation of sentence similarity. PLoS One. 2017; 12(9):e0180885. https://doi.org/10.1371/journal.pone.0180885 PMID: 28898242

75. Arora S, Liang Y, Ma T. A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations; 2017. p. 1–16.

76. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 2227–2237.

77. Akbik A, Blythe D, Vollgraf R. Contextual String Embeddings for Sequence Labeling. In: Proc. of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics; 2018. p. 1638–1649.

78. Ranasinghe T, Orasan C, Mitkov R. Enhancing Unsupervised Sentence Similarity Methods with Deep Contextualised Word Representations. In: Proc. of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). Varna, Bulgaria: INCOMA Ltd.; 2019. p. 994–1003.

79. Beltagy I, Lo K, Cohan A. SciBERT: A Pretrained Language Model for Scientific Text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019. p. 3615–3620.

80. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, et al. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. arXiv e-prints. 2020; p. arXiv:2007.15779.

81. Wada S, Takeda T, Manabe S, Konishi S, Kamohara J, Matsumura Y. A pre-training technique to localize medical BERT and to enhance biomedical BERT. arXiv e-prints. 2020; p. arXiv:2005.07202.

82. Oliva J, Serrano JI, del Castillo MD, Iglesias Á. SyMSS: A syntax-based measure for short-text semantic similarity. Data Knowl Eng. 2011; 70(4):390–405. https://doi.org/10.1016/j.datak.2011.01.002

83. Inan E. SimiT: A Text Similarity Method Using Lexicon and Dependency Representations. New Generation Computing. 2020; p. 1–22.

84. Bär D, Biemann C, Gurevych I, Zesch T. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In: Proc. of the First Joint Conference on Lexical and Computational Semantics—Volume 1: Proc. of the Main Conference and the Shared Task, and Volume 2: Proc.

of the Sixth International Workshop on Semantic Evaluation. SemEval'12. Stroudsburg, PA, USA: Association for Computational Linguistics; 2012. p. 435–440.

85. Šarić F, Glavaš G, Karan M, Šnajder J, Bašić BD. TakeLab: Systems for Measuring Semantic Text Similarity. In: Proc. of the First Joint Conference on Lexical and Computational Semantics—Volume 1: Proc. of the Main Conference and the Shared Task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation. SemEval'12. Stroudsburg, PA, USA: Association for Computational Linguistics; 2012. p. 441–448.

86. Chen Q, Du J, Kim S, Wilbur WJ, Lu Z. Combining rich features and deep learning for finding similar sentences in electronic medical records. Proceedings of the BioCreative/OHNLP Challenge. 2018; p. 5–8.

87. Rychalska B, Pakulska K, Chodorowska K, Walczak W, Andruszkiewicz P. Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity. In: Proc. of the 10th International Workshop on Semantic Evaluation (SemEval-2016). ACL; 2016. p. 602–608.

88. Al-Natsheh HT, Martinet L, Muhlenbach F, Zighed DA. UdL at SemEval-2017 Task 1: Semantic Textual Similarity Estimation of English Sentence Pairs Using Regression Model over Pairwise Features. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 115–119.

89. Farouk M. Sentence Semantic Similarity based on Word Embedding and WordNet. In: 2018 13th International Conference on Computer Engineering and Systems (ICCES). ieeexplore.ieee.org; 2018. p. 33–37.

90. Nguyen HT, Duong PH, Cambria E. Learning short-text semantic similarity with word embeddings and external knowledge sources. Elsevier. 2019; 182:104842.

91. Bounab Y, Seppnen J, Savusalo M, Mkynen R, Oussalah M. Sentence to Sentence Similarity. A Review. In: Conference of Open Innovations Association, FRUCT. elibrary.ru; 2019. p. 439–443.

92. Sultan MA, Bethard S, Sumner T. DLS @ CU: Sentence Similarity from Word Alignment. In: Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014). ACL; 2014. p. 241–246.

93. Sultan MA, Bethard S, Sumner T. DLS @ CU: Sentence Similarity from Word Alignment and Semantic Vector Composition. In: Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015). ACL; 2015. p. 148–153.

94. Agirre E, Cer D, Diab M, Gonzalez-Agirre A. Semeval-2012 task 6: A pilot on semantic textual similarity. In: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proc. of the main conference and the shared task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). ACL; 2012. p. 385–393.

95. Agirre E, Cer D, Diab M, Gonzalez-Agirre A, Guo W. * SEM 2013 shared task: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proc. of the Main Conference and the Shared Task: Semantic Textual Similarity. vol. 1. ACL; 2013. p. 32–43.

96. Agirre E, Banea C, Cardie C, Cer D, Diab M, Gonzalez-Agirre A, et al. Semeval-2014 task 10: Multilingual semantic textual similarity. In: Proc. of the 8th international workshop on semantic evaluation (SemEval 2014). ACL; 2014. p. 81–91.

97. Agirre E, Banea C, Cardie C, Cer D, Diab M, Gonzalez-Agirre A, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In: Proc. of the 9th international workshop on semantic evaluation (SemEval 2015). ACL; 2015. p. 252–263.

98. Agirre E, Banea C, Cer D, Diab M, others. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. 10th International Workshop on Semantic Evaluation (SemEval-2016). 2016;.

99. Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 1–14.

100. Wang Y, Afzal N, Liu S, Rastegar-Mojarad M, Wang L, Shen F, et al. Overview of the BioCreative/OHNLP Challenge 2018 Task 2: Clinical Semantic Textual Similarity. Proc of the BioCreative/OHNLP Challenge. 2018; 2018.

101. Han L, Kashyap AL, Finin T, Mayfield J, Weese J. UMBC_EBIQUITY-CORE: semantic textual similarity systems. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity. vol. 1. ACL; 2013. p. 44–52.

102. Sultan MA, Bethard S, Sumner T. Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). ACL; 2015. p. 148–153.

103. Donnelly K. SNOMED-CT: The advanced terminology and coding system for eHealth. Books Google. 2006; 121:279–290. PMID: 17095826

104. Miller GA. WordNet: A Lexical Database for English. ACM. 1995; 38(11):39–41. https://doi.org/10.1145/219717.219748

105. Harris Z. Distributional Hypothesis. Word World. 1954; 10(23):146–162.

106. Shen D, Wang G, Wang W, Min MR, Su Q, Zhang Y, et al. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 440–450.

107. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010; 17(3):229–236. https://doi.org/10.1136/jamia.2009.002733 PMID: 20442139

108. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010; 17(5):507–513. https://doi.org/10.1136/jamia.2009.001560 PMID: 20819853

109. Sánchez D, Batet M, Isern D. Ontology-based information content computation. Knowledge-Based Systems. 2011; 24(2):297–303. https://doi.org/10.1016/j.knosys.2010.10.001

110. Cai Y, Zhang Q, Lu W, Che X. A hybrid approach for measuring semantic similarity based on IC-weighted path distance in WordNet. Journal of intelligent information systems. 2017; p. 1–25.

111. Rada R, Mili H, Bicknell E, Blettner M. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics. 1989; 19(1):17–30. https://doi.org/10.1109/21.24528

112. Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of International Conference Research on Computational Linguistics (ROCLING X); 1997. p. 19–33.

113. Chapman S, Norton B, Ciravegna F. Armadillo: Integrating knowledge for the semantic web. In: Proceedings of the Dagstuhl Seminar in Machine Learning for the Semantic Web. Researchgate; 2005. p. 90.

114. Chiu B, Crichton G, Korhonen A, Pyysalo S. How to Train good Word Embeddings for Biomedical NLP. In: Proc. of the 15th Workshop on Biomedical Natural Language Processing. Berlin, Germany: Association for Computational Linguistics; 2016. p. 166–174.

115. Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, St John R, et al. Universal Sentence Encoder for English. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 169–174.

116. Huang K, Altosaar J, Ranganath R. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. arXiv e-prints. 2019; p. arXiv:1904.05342.

117. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP natural language processing toolkit. In: Proc. of 52nd annual meeting of the association for computational linguistics: system demonstrations. ACL; 2014. p. 55–60.

118. Comeau DC, Islamaj Doğan R, Ciccarese P, Cohen KB, Krallinger M, Leitner F, et al. BioC: a minimalist approach to interoperability for biomedical text processing. Database. 2013; 2013:bat064. https://doi.org/10.1093/database/bat064 PMID: 24048470

119. Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. J Am Med Inform Assoc. 2017; 24(4):841–844. https://doi.org/10.1093/jamia/ocw177 PMID: 28130331

120. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004; 32(Database issue):267–70. https://doi.org/10.1093/nar/gkh061 PMID: 14681409

121. Smith L, Rindflesch T, Wilbur WJ. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics. 2004; 20(14):2320–2321. https://doi.org/10.1093/bioinformatics/bth227 PMID: 15073016

122. Lastra-Díaz JJ, Lara-Clares A, Garcia-Serrano A. HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models; 2020. e-cienciaDatos, v1. https://doi.org/10.21950/1RRAWJ.

123. Reátegui R, Ratté S. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. BMC Med Inform Decis Mak. 2018; 18(Suppl 3):74. https://doi.org/10.1186/s12911-018-0654-2 PMID: 30255810

124. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology. 2007; 25(11):1251–1255. https://doi.org/10.1038/nbt1346 PMID: 17989687

125. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation OSDI 16). usenix.org; 2016. p. 265–283.

126. Xiao H. bert-as-service; 2018. https://github.com/hanxiao/bert-as-service.

127. Dustin DS, Alfonsin B. Similarity and liking. Psychon Sci. 1971; 22(2):119–119. https://doi.org/10.3758/BF03332524

128. Agirre E, Alfonseca E, Hall K, Kravalova J, Paşca M, Soroa A. A Study on Similarity and Relatedness Using Distributional and WordNet-Based Approaches. In: Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL'09. USA: Association for Computational Linguistics; 2009. p. 19–27.

129. Bird S, Klein E, Loper E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc.; 2009.

130. Couto FM, Krallinger M. Proposal of the First International Workshop on Semantic Indexing and Information Retrieval for Health from Heterogeneous Content Types and Languages (SIIRH). In: Advances in Information Retrieval. Springer International Publishing; 2020. p. 654–659.

131. Mishra R, Bian J, Fiszman M, Weir CR, Jonnalagadda S, Mostafa J, et al. Text summarization in the biomedical domain: a systematic review of recent research. J Biomed Inform. 2014; 52:457–467. https://doi.org/10.1016/j.jbi.2014.06.009 PMID: 25016293

132. Cabot C, Darmoni S, Soualmia LF. Cimind: A phonetic-based tool for multilingual named entity recognition in biomedical texts. J Biomed Inform. 2019; 94:103176. https://doi.org/10.1016/j.jbi.2019.103176 PMID: 30980962

133. Abdeddaïm S, Vimard S, Soualmia LF. The MeSH-Gram Neural Network Model: Extending Word Embedding Vectors with MeSH Concepts for Semantic Similarity. In: Ohno-Machado L, Séroussi B, editors. MEDINFO 2019: Health and Wellbeing e-Networks for All—Proceedings of the 17th World Congress on Medical and Health Informatics. vol. 264 of Studies in Health Technology and Informatics. IOS Press; 2019. p. 5–9.

Chapter 8

BMC Bioinformatics article

BMC Bioinformatics

SOFTWARE

Open Access

HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey

Juan J. Lastra-Díaz*, Alicia Lara-Clares and Ana Garcia-Serrano

*Correspondence:
jlastra@invi.uned.es
NLP & IR Research Group,
E.T.S.I. Informática,
Universidad Nacional de
Educación a Distancia
(UNED), C/Juan del Rosal 16,
28040 Madrid, Spain

Abstract

Background: Ontology-based semantic similarity measures based on SNOMED-CT, MeSH, and Gene Ontology are being extensively used in many applications in biomedical text mining and genomics respectively, which has encouraged the development of semantic measures libraries based on the aforementioned ontologies. However, current state-of-the-art semantic measures libraries have some performance and scalability drawbacks derived from their ontology representations based on relational databases, or naive in-memory graph representations. Likewise, a recent reproducible survey on word similarity shows that one hybrid IC-based measure which integrates a shortest-path computation sets the state of the art in the family of ontology-based semantic measures. However, the lack of an efficient shortest-path algorithm for their real-time computation prevents both their practical use in any application and the use of any other path-based semantic similarity measure.

Results: To bridge the two aforementioned gaps, this work introduces for the first time an updated version of the HESML Java software library especially designed for the biomedical domain, which implements the most efficient and scalable ontology representation reported in the literature, together with a new method for the approximation of the Dijkstra's algorithm for taxonomies, called Ancestors-based Shortest-Path Length (AncSPL), which allows the real-time computation of any path-based semantic similarity measure.

Conclusions: We introduce a set of reproducible benchmarks showing that HESML outperforms by several orders of magnitude the current state-of-the-art libraries in the three aforementioned biomedical ontologies, as well as the real-time performance and approximation quality of the new AncSPL shortest-path algorithm. Likewise, we show that AncSPL linearly scales regarding the dimension of the common ancestor subgraph regardless of the ontology size. Path-based measures based on the new AncSPL algorithm are up to six orders of magnitude faster than their exact implementation in large ontologies like SNOMED-CT and GO. Finally, we provide a detailed reproducibility protocol and dataset as supplementary material to allow the exact replication of all our experiments and results.

Keywords: HESML, Semantic measures library, Ontology-based semantic similarity measures, Information content models, SNOMED-CT, MeSH, Gene ontology, WordNet

Background

The development of the gene ontology (GO) [1, 2] has given rise to many significant applications in genomics and proteomics derived from some significant findings that show the correlation of GO-based semantic similarity between genes and proteins with some biological phenomena. For instance, the pioneering work of Lord et al. [3] shows that protein sequence similarity is highly correlated with their corresponding GO-based semantic similarity, which suggests that GO-based similarity measures could be used as protein function prediction tools. Likewise, Freudenberg and Propping [4] show that GO-based similarity measures can be used for the prediction of disease-relevant genes, whilst Sevilla et al. [5] show that gene expression is correlated with GO-based semantic similarity, and Couto et al. [6, 7] show that there is a correlation between the GO-based semantic similarity of proteins and their family similarity based on the Pfam database. As a consequence of these aforementioned findings, a plethora of GO-based semantic similarity measures have been proposed during the last two decades [8–11] which are commonly evaluated in multiple benchmarks [12, 13] using some protein similarity proxies based on their sequence, structure, or common metabolic pathways. Other significant applications of GO-based similarity measures are the prioritization of disease gene candidates [14–16], protein clustering [17], network alignment of protein interaction networks [18], protein functional similarity [19], prediction of the molecular function of genes [20], and characterization of human regulatory pathways [21]. For the reasons above, many software libraries and tools implementing GO-based similarity measures have been proposed in the literature, such as follows: (1) online web tools such as FuSSiMeg [7, 22], G-SESAME [23, 24], FunSimMat [25, 26], Proteinon [27], DaGO-Fun [28], GOssTo [29] and SemSim [30]; (2) R-packages such as GOSim [31] and GOSemSim [32] among others; (3) Python libraries such as FastSemSim [9] and A-DaGO-Fun [33]; and finally, (4) the Java software library called SML [34], which provides an unified and standalone implementation of the most significant ontologies, in addition to set significantly the state-of-the-art for the family of GO-based libraries in terms of performance [34, table 1].

On the other hand, ontology-based semantic similarity measures [35, 36] have been extensively used to estimate the degree of similarity between concepts as perceived by a human being in many text mining and information retrieval (IR) applications, both in the general language domain [35] and the biomedical domain [37, 38]. For instance, ontology-based similarity measures based on Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) ontology and the Medical Subject Headings (MeSH) thesaurus have been used in the definition or training of methods for biomedical sentence similarity [39–41], word sense disambiguation [42], estimating the semantic similarity between clinical terms [38] and concepts [43–46], inter-patient distance metrics [47], clinical text classification [48], classification of radiology reports [49], document clustering [50], retrieval of passage for biomedical question answering [51], and article screening [52] among many other applications based on the Unified Medical Language System (UMLS). In order to tackle all aforementioned applications, as well as the growing research interest on the topic, McInnes et al. [53] introduce the first UMLS-based semantic measure library reported in the literature, called UMLS::Similarity

(UMLS::Sim), which is implemented as a Perl library together with the standard MySQL database distribution of the UMLS [54] ontologies and vocabularies provided by courtesy of the NLM.[1]

Main motivation and hypotheses

The main motivation of this work is to overcome some performance and scalability drawbacks in current state-of-the-art semantic measures libraries for the biomedical domain in the fields of biomedical text mining and genomics. Despite the UMLS::Similarity has been extensively used in the literature, it has several significant drawbacks that prevent its use in high-throughput standalone applications, such as a poor performance in the evaluation of measures, as well as a tedious, complex, and long setup process to build several pre-calculated data structures and values stored into an auxiliary database called UMLS::Interface. UMLS::Similarity drawbacks are mainly derived from its use of a scripting programming language like Perl and an ontology representation based on a relational database, which strongly impacts its performance and software architecture. More recently, Harispe et al. [34] introduce the SML Java software library implementing for the first time the most significant ontologies into a single library, such as WordNet [55], SNOMED-CT, MeSH, the Gene Ontology and any others based on the OBO [56] and OWL file formats. However, SML has several significant performance and scalability drawbacks derived from the use of a naive in-memory graph representation based on hash tables and caching, which significantly impacts its overall performance, and very especially, its computation of path-based measures and scalability regarding the ontology size [57, Sect. 1.1.1]. To bridge the aforementioned drawbacks, Lastra-Diaz et al. [57] introduce the HESML Java software library based on WordNet, together with a very efficient and linearly scalable taxonomy representation called PosetHERep that allows the former library outperforms SML by several orders of magnitude [57]. However, the field of biomedical research has not benefited yet from these aforementioned advances because previous HESML versions implement none of the most significant biomedical ontologies, such as SNOMED-CT, MeSH, GO, and others based on the OBO file format. Our main hypothesis is that the efficient and scalable in-memory representation for ontologies provided by HESML should solve these aforementioned performance and scalability drawbacks, as detailed in hypothesis 1 below.

Hypothesis 1 (H1) A HESML implementation of the main biomedical ontologies should significantly outperform the state-of-the-art biomedical semantic measures libraries in the evaluation of ontology-based semantic similarity measures, such as previously shown for WordNet ontology [57].

The second motivation of our work is to overcome a significant performance and scalability drawback of all path-based semantic similarity measures, which prevents their use in high-throughput experiments, or any practical application demanding their real-time computation. This problem is especially relevant because a recent reproducible survey on word similarity [58–60] shows that one hybrid IC-based similarity measure [35, coswJ&C] sets the state of the art in the family of ontology-based measures for the

[1] https://www.nlm.nih.gov/.

general domain. However, their practical use in any application is limited because of the lack of an efficient shortest-path algorithm for their real-time computation. Path-based similarity measures require an efficient implementation of any shortest-path algorithm, such as Dijkstra's algorithm [61]; however, its computational complexity prevents its practical use in high-throughput applications based on large ontologies like SNOMED-CT, GO or WordNet. A common strategy followed by most of the software libraries and tools to tackle the aforementioned problem is to pre-calculate some auxiliary data structures, or all pairwise similarity scores, with the aim of speeding-up the subsequent evaluation of any path-based measure, such as done by UMLS::Similarity, whilst other libraries like SML compute the path-based measures on-the-fly, and store the resulting similarity scores into a cache. The caching of auxiliary data structures and values demands large quantities of memory and complex setup processes, which neither tackle nor solve the main practical problem on the real-time computation of path-based measures at interactive rates, and lead to a poor performance, long setup processes, and running out of memory on large ontologies when they are used on average workstations. Our hypothesis on the aforementioned problem of performance and scalability of path-based similarity measures is that a new approximated shortest-path algorithm, specifically designed for taxonomies, should overcome this problem, as detailed in hypothesis 2 below.

Hypothesis 2 (H2) A new approximated shortest-path algorithm specifically designed for taxonomies could provide an efficient and linearly scalable method for reformulating and evaluating any path-based semantic similarity measure at interactive rates, whose similarity values would show a high-correlation value as regards its implementation using any exact shortest-path algorithm.

And finally, a third motivation is to provide a larger and most updated set of ontology-based semantic similarity measures and Information Content (IC) models [58, 62] than those provided by UMLS::Similarity and SML libraries, as shown in Tables 2, 3, and 4 .

The aim of this work is to introduce an updated version of the HESML [57] library especially designed for the biomedical domain, called HESML V1R5 [63], together with a fast approximation of the Dijkstra's algorithm [64] for taxonomies based on a relaxed graph spanner called Ancestors-based Shortest-Path Length (AncSPL), which allows for the first time the real-time computation of any path-based similarity measure on large ontologies, such as SNOMED-CT, GO, and WordNet. HESML V1R5 implements most of the ontology-based similarity measures and IC models reported in the literature as shown in Tables 2, 3 and 4, as well as a very efficient and scalable in-memory representation of WordNet [55], SNOMED-CT, MeSH, GO [1], and other ontologies based on the OBO file format [56]. We introduce a set of reproducible benchmarks for testing our main hypothesis (H1) by comparing the performance of HESML with the UMLS::Similarity and SML libraries on the three most significant biomedical ontologies, as well as several experiments for testing our second hypothesis (H2) as regards the new AncSPL algorithm. Finally, we introduce a reproducibility dataset [65] together with a detailed reproducibility protocol, which is provided as supplementary material (see Additional file 1) to allow the exact replication of all our experiments and results.

Table 1 Ontologies and thesaurus implemented by the three main semantic measures libraries for the biomedical domain

Ontology	UMLS::Similarity	SML	HESML
MeSH	x	x	x
SNOMED	x	x	x
WordNet		x	x
OBO file format		x	x
Gene Ontology		x	x
OWL file format		x	
RDF triples files		x	

Related work

This section briefly reviews the literature on semantic measures libraries and tools for the biomedical domain, as well as the family of approximated shortest-path algorithms based on graph spanners [66–68], which are related with HESML and our AncSPL algorithm.

Biomedical semantic measures libraries

The main ontologies used for biomedical text mining and information retrieval applications in health sciences are SNOMED-CT and MeSH, although there are many other ontologies[2] based on the OBO file format [56]. Nowadays, there are only two semantic measures libraries based on the two aforementioned ontologies as follows: (1) the pioneering Perl software library and online web interface called UMLS::Similarity [53], and (2) the most recent Java software library called SML [34], which introduces several significant contributions, such as a portable and efficient object-oriented language programming, as well as a significant number of methods as shown in Tables 2, 3 and 4, and the implementation for the first time of the most significant biomedical ontologies and WordNet into a single software library, as shown in Table 1. However, both UMLS::Similarity and SML have several significant performance and scalability drawbacks previously detailed in the introduction which encourage our research in this work.

On the other hand, most early GO-based software libraries and tools have been implemented as online web tools, such as FuSSiMeg [7, 22], G-SESAME [23, 24], FunSimMat [25, 26], Proteinon [27], DaGO-Fun [28], GOssTo [29] and SemSim [30]. FuSSiMeg [22] introduces the first semantic similarity measure specifically designed for GO terms together with an online web tool for its evaluation, whilst Proteinon [27] provides the first online tool for evaluating GO-based protein semantic similarity. G-SESAME [23, 24] provides a large set of online tools for measuring the semantic similarity between GO terms and the GO-based functional similarity between genes and proteins. FunSimMat [25, 26] provides tools for GO-based protein functional similarity and disease gene prioritization. DaGO-Fun [28] web tool provides a rich set of GO-based similarity measures for GO terms, genes and proteins, as well as tools for the identification of gene and

[2] http://www.obofoundry.org.

Table 2 Pairwise ontology-based semantic similarity measures implemented by the three main publicly available software libraries for the biomedical domain

	UMLS::Similarity	SML	HESML
Gloss-based measures			
Banerjee and Pedersen [69]	x		
Patwardhan and Pedersen [70], context vector	x		
Path-based and taxonomy-based measures			
Rada et al. [71]	x	x	x*
Wu and Palmer [72]		x	x
Wu and Palmer [72] fast (depth-based approximation)	x		x
Leacock and Chodorow [73]	x	x	x*
Stojanovic et al. [74]		x	x*
Maedche and Staab [75]	x		
Zhong et al. [76]	x		
Pekar and Staab [77]	x	x	x*
Li et al. [78], strategy 3			x*
Li et al. [78], strategy 4			x*
Liu et al. [79], strategy 1			x*
Liu et al. [79], strategy 2			x*
Pedersen et al. [44], reciprocal Rada	x		x*
Al-Mubaid and Nguyen [80]	x		x*
Kyogoku et al. [81]		x	
Batet et al. [45]	x		
Hao et al. [82]			x*
Hadj Taieb et al. [83], sim1			x
Hadj Taieb et al. [83], sim2			x
McInnes et al. [84], U-path	x		
IC-based measures			
Resnik [85]	x	x	x
Jiang and Conrath [86]	x	x	x
Lin [87]	x	x	x
Schlicker et al. [88]		x	x
Pirró and Seco [89]			x
FaITH [90]	x		x
Garla and Brandt [91]			x
Meng and Gu [92]			x
Gao et al. [93], strategy 3			x
Lastra&García [35], cosJ&C			x
Cai et al. [94], strategy 2			x
Hybrid IC-based measures			
Li et al. [[78] strategy 9			x*
Zhou et al. [95]			x*
Meng et al. [96]			x*
Gao et al. [93], strategy 3			x*
Lastra and García [35], coswJ&C			x*
Lastra and García [35], weigthedJ&C			x*
Cai et al. [94], strategy 1			x*
Feature-based measures			
Sánchez et al. [97]	x		x

(*) Real-time reformulation of all path-based measures based on the AncSPL algorithm

Table 3 Groupwise ontology-based semantic similarity measures implemented by SML and HESML (this work), which are mainly used for genomics applications based on the GO ontology

Groupwise similarity measures	SML	HESML
Maximum [5, formula 2]		x
Average [104, formula 1]		x
Best-Match-Average (BMA) [104, formula 2]		x
SimUI [100]	x	x
SimLP [100]	x	x
SimGIC [105]	x	x
Ali and Deane [18]	x	
Lee et al. [106]	x	
Term Overlap (TO) [107]	x	
Normalized Term Overlap (NTO) [107]	x	
NTO_MAX [107]	x	

protein candidates for diseases, and tools for gene and protein clustering among others. GOssTo [29] is an online web tool for measuring GO-based similarity between organisms, which implements six similarity measures and it is also distributed as a standalone program based on Java together with an API for developers. SemSim [30] is a web tool which introduces several tools for measuring GO-based similarity between genes and organisms, as well as predicting gene and protein GO annotations, in addition to providing programmatic access to its functionality via Web services. We also find a standalone software called DynGO [98] and other standalone software libraries distributed as R-packages, such as GOSim [31], SemSim [99], GOStats [100], csbl.go [101], and GOSemSim [32]; Python libraries such as FastSemSim [9] and A-DaGO-Fun [33]; and finally, the aforementioned Java software library called SML [34] which sets the state-of-the-art for the family of GO-based libraries in terms of performance [34, Table 1]. Finally, Le [102] recently introduces a Cytospace [103] app called UFO, which implements a collection of semantic similarity measures and enrichment tools for biomedical ontologies based on the OBO file format.

Shortest-path algorithms based on graph spanners

Our new AncSPL shortest-path algorithm for taxonomies provides an approximated solution for the Single-Source Shortest-Path (SSSP) problem whose aim is to find the shortest-path from a single vertex to the rest of vertexes in a graph. The AncSPL algorithm belongs to the family of approximation methods based on sub-graphs, and it is closely related to the methods based on *graph spanners* whose core idea is to build a simplified version $G' = (V, E')$ of a weighted graph $G = (V, E)$ whose shortest-path distance function satisfies an upper error bound a priori. For this reason, this section focuses on graph spanners. For a comprehensive review of the literature on shortest-path algorithms, we refer the reader to the surveys by Sommer [122], Madkour et al. [123], and Zwick [124].

Table 4 Information Content models implemented by the main publicly available software libraries for the biomedical domain

IC models	UMLS ::Similarity	SML	HESML
Corpus-based IC models			
Resnik [85, 108]	x	x	x
CPCorpus [62], CPCorpus			x
CPRefCorpus [109],			x
Intrinsic IC models			
Seco et al. [110]	x	x	x
Blanchard et al. [111], *IC*$_g$			x
Zhou et al. [112]		x	x
Sebti and Barfroush [113]			x
Sánchez et al. [114]	x	x	x
Sánchez and Batet [115]			x
Meng et al. [116]			x
Harispe et al. [34]		x	x
Yuan et al. [117]			x
Hadj Taieb et al. [118]			x
Adhikari et al. [119]			x
Ben Aouicha and Hadj Taieb [120]			x
Ben Aouicha et al. [121]			x
CondProbHyponyms [62]			x
CondProbUniform [62]			x
CondProbLeaves [62]			x
CondProbCosine [62]			x
CondProbLogistic [62]			x
CondProbRefHyponyms [62]			x
CondProbRefUniform [62]			x
CondProbRefLeaves [62]			x
CondProbRefCosine [62]			x
CondProbRefLogistic [62]			x
CondProbCosineLeaves [62]			x
CondProbRefLogistic-Leaves [62]			x
CondProbRefLeaves-SubsumerRatio [62]			x

Graph spanners are pioneering by the works of Peleg and Schaffer [66] and Althofer et al. [67], whilst the current state-of-the-art spanner construction algorithm is introduced by Elkin and Solomon [68]. Given a graph $G = (V, E)$, a sub-graph $G' = (V, E')$ is a t-spanner if for every vertex pair $u, v \in V$ the distance in the sub-graph $d_{G'}(u, v)$ is at most t times longer than the distance $d_G(u, v)$ in G, such that $\forall u, v \in V, d_{G'}(u, v) \leq t \cdot d_G(u, v)$. Spanner-based algorithms are based on well-founded theoretical results in graph theory, in addition to be of great practical value in many scenarios. However, they have two drawbacks in the context of our problem as follows. On the one hand, graph spanners have a high complexity derived from the need for computing a spanning graph considering all graph vertexes, and on the other hand, they do not take advantage of the knowledge of the graph structure in special cases such as the single-root taxonomies considered herein. Elkin and Solomon [68] point that “the only algorithms for constructing sparse and light-weight spanners for general graphs admit high running times”. Precisely, we propose

AncSPL to take advantage of the intrinsic structure of the single-root taxonomies to provide an efficient approximation SSSP algorithm.

Implementation

This section is divided into two parts as follows. First part introduces the new semantic measures library for the biomedical domain, called HESML V1R5, whilst the second part introduces a real-time algorithm for the computation of the shortest-path between concepts in large ontologies, called AncSPL, whose performance and approximation quality are tested in our experiments.

The new semantic measures library

HESML V1R5 is a new version of the HESML [57] open-source Java software library that extends its applicability to the biomedical domain by implementing the SNOMED-CT, MeSH, GO [1, 2], and OBO file format ontologies [56], in addition to WordNet [55]. HESML V1R5 is a self-contained Java software library of pairwise and groupwise ontology-based semantic similarity measures, and information content (IC) models, which also supports the evaluation of pre-trained word embedding models in three different file formats. The core innovation of HESML is a very efficient and linearly scalable in-memory representation for taxonomies, called PosetHERep, which was introduced in the first version of HESML [57] based on WordNet. PosetHERep is mainly responsible for the real-time performance and scalability with low memory consumption shown by HESML. PosetHERep converts HESML V1R5 into the most efficient, scalable, and portable semantic measures library reported in the literature, as shown by the benchmarks based on WordNet and large synthetic ontologies reported in [57], and the benchmarks on biomedical ontologies evaluated in this work. For more information on the data structures and algorithms of the PosetHERep representation model, we refer the reader to [57, Sect. 3.2].

HESML V1R5 implements the largest set of pairwise ontology-based semantic measures and IC models reported in the literature, as shown in Tables 2 and 4 respectively. However, this first version of HESML for the biomedical domain does not include some specific GO-based pairwise and groupwise similarity measures which will be included in forthcoming versions. Likewise, HESML V1R5 provides for the first time real-time reformulations for most of the path-based and hybrid IC-based measures reported in the literature, which are based on the new AncSPL shortest-path algorithm introduced herein.

HESML V1R5 is a self-contained evaluation and experimentation platform on word and concept similarity and relatedness, which is especially well suited to run large experimental surveys by supporting the execution of automatic reproducible experiment files based on different XML-based file formats. Despite HESML V1R5 implements the most significant ontologies reported in the literature, it could also be easily extended to manage other ontology file formats, such as OWL or RDF files, by implementing the proper parsers as detailed in [57]. HESML V1R5 library has been completely developed in NetBeans 8 and Java 8, being distributed with three WordNet versions and GO. HESML V1R5 integrates some complementary Java console programs shown in turquoise blue boxes in Fig. 1, which use the HESML core library to

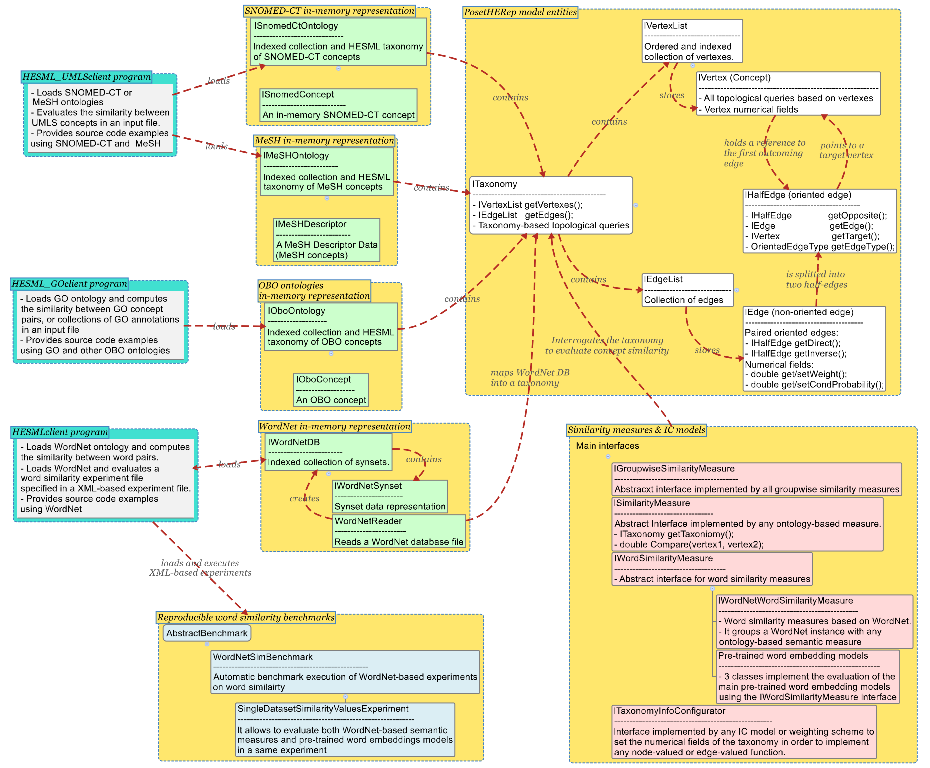

SNOMED-CT in-memory representation
ISnomedCtOntology
Indexed collection and HESML taxonomy of SNOMED-CT concepts
ISnomedConcept
An in-memory SNOMED-CT concept
HESML_UMLSclient program
- Loads SNOMED-CT or MeSH ontologies
- Evaluates the similarity between UMLS concepts in an input file.
- Provides source code examples using SNOMED-CT and MeSH
loads
loads
MeSH in-memory representation
IMeSHOntology
Indexed collection and HESML taxonomy of MeSH concepts
IMeSHDescriptor
A MeSH Descriptor Data (MeSH concepts)
contains
HESML_GOclient program
- Loads GO ontology and computes the similarity between GO concept pairs, or collections of GO annotations in an input file
- Provides source code examples using GO and other OBO ontologies
loads
OBO ontologies in-memory representation
IOboOntology
Indexed collection and HESML taxonomy of OBO concepts
IOboConcept
An OBO concept
contains
HESMLclient program
- Loads WordNet ontology and computes the similarity between word pairs.
- Loads WordNet and evaluates a word similarity experiment file specified in a XML-based experiment file.
- Provides source code examples using WordNet
loads
WordNet in-memory representation
IWordNetDB
Indexed collection of synsets.
contains
creates
IWordNetSynset
Synset data representation
WordNetReader
Reads a WordNet database file
loads and executes XML-based experiments
Reproducible word similarity benchmarks
AbstractBenchmark
WordNetSimBenchmark
Automatic benchmark execution of WordNet-based experiments on word similairty
SingleDatasetSimilarityValuesExperiment
It allows to evaluate both WordNet-based semantic measures and pre-trained word embeddings models in a same experiment
PosetHERep model entities
contains
IVertexList
Ordered and indexed collection of vertexes.
stores
contains
IVertex (Concept)
- All topological queries based on vertexes
- Vertex numerical fields
holds a reference to the first outcoming edge
points to a target vertex
ITaxonomy
- IVertexList getVertexes();
- IEdgeList getEdges();
- Taxonomy-based topological queries
IHalfEdge (oriented edge)
- IHalfEdge getOpposite();
- IEdge getEdge();
- IVertex getTarget();
- OrientedEdgeType getEdgeType();
contains
IEdgeList
Collection of edges
is splitted into two half-edges
contains
IEdge (non-oriented edge)
Paired oriented edges:
- IHalfEdge getDirect();
- IHalfEdge getInverse();
Numerical fields:
- double get/setWeight();
- double get/setCondProbability();
Interrogates the taxonomy to evaluate concept similarity
stores
maps WordNet DB into a taxonomy
Similarity measures & IC models
Main interfaces
IGroupwiseSimilarityMeasure
Abstracxt interface implemented by all groupwise similarity measures
ISimilarityMeasure
Abstract Interface implemented by any ontology-based measure.
- ITaxonomy getTaxoniomy();
- double Compare(vertex1, vertex2);
IWordSimilarityMeasure
- Abstract interface for word similarity measures
IWordNetWordSimilarityMeasure
- Word similarity measures based on WordNet.
- It groups a WordNet instance with any ontology-based semantic measure
Pre-trained word embedding models
- 3 classes implement the evaluation of the main pre-trained word embedding models using the IWordSimilarityMeasure interface
ITaxonomyInfoConfigurator
Interface implemented by any IC model or weighting scheme to set the numerical fields of the taxonomy in order to implement any node-valued or edge-valued function.

Fig. 1 HESML V1R5 architecture showing the main functional blocks and abstract interfaces. Boxes in yellow show main abstract objects and interfaces contained in the HESML library, whilst boxes in turquoise blue show main HESML client programs, whose aim is to evaluate semantic similarity measures implemented in HESML on the SNOMED-CT, MeSH, GO, and WordNet ontologies

run reproducible experiments and evaluate the semantic similarity between words, UMLS concepts, or GO terms and GO annotation sets (genes and proteins) which are based on WordNet, SNOMED-CT or MeSH, and GO.

HESML Software Architecture. Figure 1 shows a concept map detailing the HESML V1R5 architecture. The core HESML component is the half-edge taxonomy representation (PosetHERep) defined by the yellow entities within the largest box in yellow. Red entities in the block entitled 'Similarity measures & IC models' represent the interfaces that should be implemented to define new methods, such as general groupwise (IGroupwiseSimilarityMeasure) or pairwise (ISimilarityMeasure) similarity measures, word similarity measures (IWordSimilarityMeasure) including pre-trained word embedding models, or new IC models (ITaxonomyInfoConfigurator). Every type of ontology is implemented by a specific collection of Java classes and interfaces which holds a ITaxonomy object to represent its corresponding ontology, such as the ISnomedCtOntology, IMeSHOntology, IOboOntology and IWordNetDB interfaces shown in Fig. 1. All the HESML objects are provided as Java interfaces, being instanced by factory objects not represented in the figure above. For a detailed

Table 5 Collection of pre-trained word embedding (WE and WEC) models and ontology-based vector models (OVM) evaluated in a previous series of experiments [58–60] by using the Java classes implementing their evaluation

WN	Family	Word embedding model
Yes	WEC	Attract-repel [127]
No	WE	FastText [128]
No	WE	GloVe [129]
No	WE	CBOW [130]
Yes	WEC	SymPatterns (SP-500d) [131]
No	WEC	Paragram-ws [132]
No	WEC	Paragram-sl [132]
Yes	WEC	Counter-fitting (CF) [133]
Yes	OVM	WN-RandomWalks [134]
Yes	OVM	WN-UKB [125]
Yes	OVM	Nasari [126]

First column details which methods use WordNet during their training

introduction to the software architecture, PosetHERep, and main algorithms of HESML, we refer the reader to its introductory paper [57], and the HESML web page.[3]

Current methods implemented by HESML. Table 1 shows the ontologies and ontology-based file formats implemented by the three main semantic measures libraries for the biomedical domain evaluated herein, whilst Tables 2, 3, and 4 shows the pairwise and groupwise ontology-based semantic similarity measures, and the IC models, implemented by the aforementioned software libraries respectively. Finally, Table 5 shows a collection of pre-trained word embedding models which were evaluated in a large benchmark [58] on word similarity using three new HESML classes called EMBWordEmbeddingModel, UKBppvWordEmbeddingModel and NasariWordEmbeddingModel respectively, which implement the evaluation of the (*.emb), (*.ppv) UKB [125] and Nasari [126] word vector file formats. Thus, HESML is able to evaluate both semantic similarity measures based on any ontology shown in Table 1 and recent word embedding models in a common software platform.

Extending the HESML functionality. HESML can be extended in different directions by developing new features as follows: (1) further pairwise or groupwise semantic similarity measures; (2) further IC models; (3) further ontology parsers for unimplemented ontology file formats; (4) further evaluators for unimplemented pre-trained word embedding models or file formats; (5) further client programs dealing with specific ontologies; and (6) further new tools based on ontology-based semantic similarity measures, such as gene clustering and other gene enrichment tools, or sentence similarity measures among many other text mining applications. For instance, in order to develop any new similarity measure, you should develop a class, which implements the appropriate interface, by following any of the multiple source code examples in the library, then the reader should include its creation in its corresponding factory function in the class *MeasureFactory*. In order to develop any new IC model, the reader should develop

[3] http://hesml.lsi.uned.es.

a class implementing the *ITaxonomyInfoConfigurator* by deriving from *AbstractICmodel* class. Finally, HESML source code is clear and well documented, thus the readers will find a lot of source code examples to learn the HESML basics on its use and extension. In addition, the readers can subscribe to the HESML community forum, or contact the authors, as detailed in the availability section.

The new shortest-path algorithm for taxonomies

Our new shortest-path algorithm for taxonomies, called ancestors-based shortest-path length (AncSPL), is a fast approximation of the Dijkstra's algorithm that is based on a min-priority queue implementation [61] constrained to a sub-graph derived from the ancestor sets of the source and target concepts. AncSPL uses an exact shortest-path algorithm that runs on the sub-graph derived from the ancestor sets by ignoring those edges connecting to any node not belonging to the sub-graph; thus, AncSPL does not require any graph transformation or auxiliary data structure. Implementation of the Dijkstra's algorithm in HESML is very efficient because PosetHERep [57] allows traversing any taxonomy in linear time as regards the number of edges. In addition, the AncSPL algorithm is easy to implement, all topological queries required are efficiently computed by HESML and it does not require any complex auxiliary data structure or preprocessing as required by the most of approximated SSSP methods for general graphs.

Given a single-root taxonomy $\mathcal{C} = (C, \leq_C, \Gamma)$, where $(C, \leq_C)$ is a partially ordered set, and $\Gamma \in C$ is a distinguished supreme element called the root, such that $\forall c_i \in C \rightarrow c_i \leq_C \Gamma$. The core idea and underlying hypothesis of our AncSPL algorithm is that given two randomly selected taxonomy nodes $c_i, c_j \in C$, most of the shortest paths between them will be contained in a set defined by the union of their ancestor sets. Our aforementioned underlying hypothesis is always true on any tree-like taxonomy, such as MeSH, in whose case we can use a direct, exact, and linearly scalable formula (line 5, Algorithm 1) to compute the length of the shortest path. However, this later formula is not exact for general taxonomies with multiple inheritance, such as WordNet, SNOMED-CT, and GO.

Our new AncSPL algorithm is detailed in Algorithm 1 box. PosetHERep representation [57] implemented by HESML allows that all topological queries involved in the implementation of AncSPL can be efficiently computed in linear time as regards each node depth value, such as the computation of the lowest common subsumer (LCS) concept, concept depth, and ancestor sets. For this reason, the combination of fast topological queries provided by HESML together with a large graph reduction based on the ancestor sets allows getting a very efficient approximation of the exact value for the length of the shortest path between concepts in any non-tree-like taxonomy. Finally, we refer the reader to the *Vertex.getFastShortestPathDistanceTo()* method in HESML V1R5 [63] to see our current implementation of AncSPL. Likewise, we provide the definition of the LCS function used in step 5 of AncSPL, and the HESML min-priority queue implementation of the Dijkstra's algorithm in Algorithm 2 and 3 boxes, respectively.

Approximation error of AncSPL. The shortest-path length estimated by AncSPL is always greater or equal than the exact value, it means that let be $spl(c_1, c_2)$ the exact length value between concepts c_1 and c_2, then $AncSPL(c_1, c_2) \geq spl(c_1, c_2)$ for any concept pairs in any 'is-a' taxonomy, as shown in Fig. 2 for SNOMED-CT, GO, and WordNet ontologies, respectively. Consequently, the AncSPL reformulation of any path-based similarity measure will always return a less or equal similarity value than their corresponding exact version. On the other hand, $AncSPL(c_1, c_2)$ will be equal to $spl(c_1, c_2)$ when either the shortest path between both concepts is contained in the common ancestor set or the taxonomy is a tree. Thus, any AncSPL reformulation will return the same value that the original path-based measure in these latter cases, and for tree-like taxonomies as MeSH, any AncSPL reformulation will be exact for any concept pair by definition.

Algorithm 1 Ancestors-based Shortest-Path Length (AncSPL). AncSPL algorithm uses the following functions: (1) depth(c_i) function returns the number of edges from c_i concept to the root; (2) isTree(C) returns true if the taxonomy $\mathcal{C}$ is a single-root tree; (3) LCS(c_i, c_j) function returns the lowest common subsumer concept of c_i, c_j concepts; (4) predicate $(a \leq_C b)$ is true if a is descendant from b; (5) AncSet(c_i) function returns the ancestor set of c_i concept; (6) minPriorityQueueDijkstra($c_i, G \subset C, w$) algorithm computes the minimum distance from c_i concept node to all concepts in $(G \subset C, \leq_C)$ by counting edges $(w = false)$ or using the edge weights $(w = true)$; and finally, (7) getMinDistance(c_j) returns the distance from c_i to c_j.

```
 1: procedure $\textsc{AncSPL}(c_i, c_j, w)$   ▷ $c_i, c_j \in C, w \in \{true, false\}$
 2:     if $c_i == c_j$ then
 3:         dist ← 0
 4:     else if (isTree(C) and !w) then
 5:         dist ← $depth(c_i) + depth(c_j) - 2 \cdot depth(LCS(c_i, c_j))$
 6:     else
 7:         if $c_j \leq_C c_i$ then
 8:             minPriorityQueueDijkstra($c_i$, AncSet($c_j$), $w$)
 9:         else if $c_i \leq_C c_j$ then
10:             minPriorityQueueDijkstra($c_i$, AncSet($c_i$), $w$)
11:         else
12:             mergedAncestors ← AncSet($c_i$) ∪ AncSet($c_j$)
13:             minPriorityQueueDijkstra($c_i$, mergedAncestors, $w$)
14:         end if
15:         dist ← getMinDistance($c_j$)
16:     end if
17: end procedure
```

Time complexity of the AncSPL algorithm

AncSPL uses two different methods to compute the length of the shortest path between concepts as follows: (1) an exact method for tree-like taxonomies defined in step 5 of Algorithm 1, which is based on the LCS function detailed in Algorithm 2; and (2) a min-priority queue implementation of the Dijkstra's algorithm constrained to the ancestors-based subgraph defined in steps 7–14 of Algorithm 1, which is based on the efficient PosetHERep representation introduced by HESML [57] and a Java PriorityQueue object, as detailed in Algorithm 3.

Algorithm 2 LCS function returns the lowest common subsumer concept between concepts $c_i, c_j \in C$.

```
function LCS($c_i, c_j$)                              ▷ $c_i, c_j \in C$
    IVertex lcs ← $\Gamma$                             ▷ $\Gamma$ is the root
    $\delta_{max} \leftarrow 0$                        ▷ $\delta_{max} \in \mathbb{Z}^+$
    $A \leftarrow AncSet(c_i)$                         ▷ $AncSet$ is a HashSet
    $B \leftarrow AncSet(c_j)$
    for $a \in A$ do
        if $((depth(a) > \delta_{max}) \wedge (a \in B))$ then
            $\delta_{max} \leftarrow depth(a)$
            lcs ← $a$
        end if
    end for
    return lcs
end function
```

The Java PriorityQueue class uses a priority heap whose time complexity is $O(log(n))$ for the insertion (add) and poll operations, and $O(n)$ for the remove operation, as pointed out in its user's documentation.[4] Thus, the time complexity of the AncSPL algorithm detailed in Algorithm 1 box can be elucidated by directly inspecting the auxiliary function and procedure detailed in Algorithm 2 and 3 boxes, respectively.

Algorithm 3 minPriorityQueueDijkstra(c_i, G_{ij}, w) algorithm computes the minimum distance from c_i concept node to all concepts in the subgraph associated to the ancestor set $(G_{ij} \subset C, \le_C)$ by counting edges $(w = false)$ or using the edge weights $(w = true)$, and sets the minimum distance value of each concept node (IVertex).

```
procedure minPriorityQueueDijkstra($c_i, G_{ij}, w$)              ▷
        $c_i \in G_{ij}, (G_{ij} \subset C, \le_C), w \in \{true, false\}$
    for $c_j \in G_{ij}$ do                            ▷ $G_{ij}$ is a HashSet<IVertex>
        setMinDistance($c_j, \infty$)
    end for
    setMinDistance($c_i$, 0)
    PriorityQueue<IVertex> p ← new PriorityQueue<>()
    p.add($c_i$)
    while !p.isEmpty() do
        IVertex $c_j$ ← p.poll()
        IHalfEdge fOutEdge ← $c_j$.getFirstOutcomingEdge()
        IHalfEdge loop ← fOutEdge
        repeat                                         ▷ visiting all neighbours of $c_j$
            IVertex $a$ ← loop.getTarget()
            if $a \in G_{ij}$ then
                weight ← w ? loop.getEdge().getWeight() : 1.0
                newDist ← $c_j$.getMinDistance() + weight
                if newDist < $a$.getMinDistance() then
                    setMinDistance($a$, newDist)
                    p.remove($a$)
                    p.add($a$)
                end if
            end if
            loop ← loop.getOpposite().getNext()
        until loop = fOutEdge
    end while
end procedure
```

Theorem 1 *Let be a single-root taxonomy $\mathcal{C} = (C, \le_C, \Gamma)$, where $(C, \le_C)$ is a partially ordered set, and $\Gamma \in C$ is a distinguished supreme element called the root, such that $\forall c_i \in C \to c_i \le_C \Gamma$, and let be $(G_{ij} \subset C, \le_C, \Gamma)$ a sub-taxonomy of $\mathcal{C}$ made up by*

[4] https://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html.

the common ancestor set of concepts $c_i, cj \in C$, such that $G_{ij} = AncSet(c_i) \bigcup AncSet(c_j)$, where $AncSet(x) = \{c \in C, x \leq_C c\}$. Then, the time complexity of the AncSPL algorithm is linear in the dimension of the sub-taxonomy with $O(N)$, being $N = |G_{ij}|$ the dimension of the common ancestor-based sub-taxonomy G_{ij}.

Proof There are two cases and workflows for the execution of AncSPL depending on the input taxonomy is tree-like (case 1) or not (case 2). Thus, time complexity of AncSPL denoted by TC_{AncSPL} will be equal to the time complexity of the Algorithm 2 (TC_2) or the Algorithm 3 (TC_3) as proven below.

(Case 1) For tree-like taxonomies processed in step 5, AncSPL evaluates the sorthest-path length by computing the distance to the Lowest Common Subsummer (LCS) using the Algorithm 2 whose time complexity can be computed as follows:

1: Steps 2–3 takes 2 operations in constant time k_1.
2: Ancestor sets in steps 4–5 can be obtained either in 2 operations in constant time k_2 if they are cached, or $O(k_3 N)$ otherwise by retrieving the ancestor nodes using PosetHERep [57], where $N = |AncSet(c_i)| \leq |G_{ij}|$ is the number of ancestors of c_i.
3: Loop in steps 6–11 is executed N times.
3.1: Step 7 takes 3 operations in constant time k_4.
3.2: Steps 8–9 take 2 operations in constant time k_5.

Thus, summing the overall time consumed by all steps detailed above, and considering that the ancestor sets can be cached, time complexity of Algorithm 2 is as follows:

$$TC_2 = \begin{cases} O(k_1 + k_2 + (k_4 + k_5)N) = O(kN), \text{if cached} \\ O(k_1 + (2k_3 + k_4 + k_5)N) = O(kN), \text{otherwise} \end{cases}$$

(Case 2) For non-tree-like taxonomies processed by the else-branch in step 6, AncSPL computes the shortest-path length using the Algorithm 3 with the sub-taxonomy G_{ij} as input. Thus, let be $N = |G_{ij}|$ the number of common ancestor nodes, then its time complexity can be computed as follows:

1: Steps 2–5 takes exactly $N + 1$ operations in constant time k_1, it means $O(k_1(N + 1))$ time.
2: Steps 6–7 takes 2 operations in constant time k_2
3: Traversing loop in steps 8–25 is executed N times.
3.1: Step 9 requires $O(log(n))$ time, being n the current item count stored within the priority queue. However, in step 9, the queue mainly stores the adjacent nodes of the last visited node in each iteration. Thus, the time will be $O(k_3 log(\bar{E}_{G_{ij}})$ in average, where $\bar{E}_{G_{ij}}$ is the average number of adjacent nodes per ancestor for each node $c_i \in G_{ij}$.
3.2: Loop in steps 12–24 is executed $E^j_{G_{ij}}$ times $\forall c_j \in G_{ij}$, where $E^j_{G_{ij}}$ is the number of adjacent nodes of c_j contained in the sub-taxonomy G_{ij}.

3.2.1: Step 14 takes 1 operation in constant time k_4.
3.2.2: Steps 15–18 takes constant time k_5.
3.2.3: Step 19 takes $O(n)$ time for removing the visited node a, being n the current item count stored within the queue. However, using the same argument provided in step 3.1 above, the time will be $O(k_6\bar{E}_{G_{ij}})$ in average.
3.2.4: Step 20 requires $O(log(n))$ time for inserting the visited node a, but using the same argument above, the time will be $O(k_7 log(\bar{E}_{G_{ij}}))$ in average.
3.2.5: Step 23 takes 2 operations in constant time k_8

Thus, summing the overall time consumed by all steps of Algorithm 3 detailed above, its time complexity (TC_3) is:

$$\begin{aligned} TC_3 &= O(k_1(N+1) + k_2 + N(k_3 log(\bar{E}_{G_{ij}}) \\ &\quad + \bar{E}_{G_{ij}}(k_4 + k_5 + k_6\bar{E}_{G_{ij}} + k_7 log(\bar{E}_{G_{ij}}) + k_8))) \\ &= O(k_1(N+1) + k_2 + N(k_3 log(\bar{E}_{G_{ij}}) \\ &\quad + k_9\bar{E}_{G_{ij}} + k_6\bar{E}^2_{G_{ij}} + k_7\bar{E}_{G_{ij}} log(\bar{E}_{G_{ij}}))) \end{aligned}$$

because $\forall x \geq 2 \Rightarrow x^2 >> xlog(x) > log(x)$ we can approximate TC_3 as follows:

$$\begin{aligned} TC_3 &= O((k_1 + k\bar{E}^2_{G_{ij}})N + k_1 + k_2) \\ &= O((k_1 + k\bar{E}^2_{G_{ij}})N + k') \\ &= O(k\bar{E}^2_{G_{ij}}N) \end{aligned}$$

□

Corollary 1 *Let be a single-root taxonomy* $\mathcal{C} = (C, \leq_C, \Gamma)$ *as defined in theorem above,* $c_i, c_j \in C$ *two arbitrary distinct concepts,* $\bar{E}_C$ *is the average number of adjacent nodes* $\forall c \in C$*, and* N_{max} *is the maximum number of ancestor nodes for any concept* $c_i \in C$*. Then, the time complexity* (TC_{AncSPL}) *is upper bounded as follows:*

$$TC_{AncSPL} \leq \begin{cases} kN_{max}, & \text{C is tree-like} \\ k\bar{E}^2_C N_{max}, & \text{otherwise} \end{cases}$$

Proof The proof of the corollary follows directly from the proof of the theorem above.

□

The dimensions of the largest ancestor sets (N_{max}) for the ontologies evaluated herein are as follows: $N^{SND}_{max} = 129$, $N^{GO}_{max} = 98$, $N^{MSH}_{max} = 14$, and $N^{WN}_{max} = 35$. The performance of AncSPL is much higher on MeSH than the remaining ontologies because, on the one hand, its N_{max} value is significantly lower than the corresponding value of the remaining ontologies, and on the other hand, the AncSPL time complexity is much lower for tree-like ontologies than for non-tree-like ones because TC_2 linearly depends on kN, whilst TC_3 depends on $k\bar{E}^2_{G_{ij}}N$. Thus, the intrinsic feature $\bar{E}^2_{G_{ij}}$ scales the time complexity of AncSPL on non-tree-like ontologies, as shown in Fig. 3.

Table 6 Average speed in CUI concept pairs per second (pairs/s) for the evaluation of random CUI pairs with three representative ontology-based similarity measures based on the SNOMED-CT US 2019AB ontology (357,406 nodes) implemented by the three UMLS-based semantic measures libraries reported in the literature

Similarity measure	UMLS::Similarity	SML	HESML
	Avg. speed (pairs/s)	Avg. speed (pairs/s)	Avg. speed (pairs/s)
Rada [71]	**0.122** (15)	xxx	0.041 (15)
AncSPL-Rada (this work)	–	–	**30110** (10^7)
Lin-Seco [87, 110]	0.744 (500)	202160 (10^7)	**491942** (10^7)
Wu-Palmer$_{fast}$ [72]	0.035 (15)	–	**435252** (10^7)

Best performing values are shown in bold. Non-implemented methods (–) or more than 1 h/pair (xxx). UMLS::Similarity uses caching for the shortest path computations. The number of random CUI pairs evaluated to measure each value is shown between parentheses

Reformulating any path-based similarity measure

Any path-based semantic similarity or distance measure can be reformulated using the AncSPL algorithm by substituting the call to the function *spl* computing the exact length of the shortest path between concepts by a call to the *AncSPL* function. For example, formulas (1–2) show the AncSPL reformulation of the reciprocal Rada et al. distance [71], called sim_{path} [44], whilst formulas (3–4) show the reformulation of the Leacock-Chodorow [73] similarity measure.

$$sim_{path}(c_1, c_2) = \frac{1}{1 + spl(c_1, c_2)} \tag{1}$$

$$sim_{AncSPL-path}(c_1, c_2) = \frac{1}{1 + AncSPL(c_1, c_2)} \tag{2}$$

$$sim_{L\&C}(c_1, c_2) = -log\left(\frac{1 + spl(c_1, c_2)}{2 \times maxDepth}\right) \tag{3}$$

$$sim_{AncSPL-L\&C}(c_1, c_2) = -log\left(\frac{1 + AncSPL(c_1, c_2)}{2 \times maxDepth}\right) \tag{4}$$

Results

This section introduces a series of reproducible experiments whose main goals are as follows: (1) to test our main hypothesis H1 by evaluating and comparing the performance of the new HESML V1R5 library with the state-of-the-art biomedical semantic measure libraries based on the main biomedical ontologies; and (2) to test our second hypothesis H2 on the new AncSPL shortest-path algorithm introduced in this work. All experiments reported herein were implemented in an Ubuntu 20.04 desktop based on one AMD Ryzen 7 5800x CPU (16 cores) with 64 Gb RAM and 2TB Gb SSD disk. Likewise, we provide a very detailed reproducibility protocol and dataset as

Table 7 Average speed in CUI concept pairs per second (pairs/s) for the evaluation of random CUI pairs with three representative ontology-based similarity measures based on the MeSH ontology (Nov, 2019. 59,747 nodes) implemented by the three UMLS-based semantic measures libraries reported in the literature

Similarity measure	UMLS::Similarity	SML	HESML
	Avg. speed (pairs/s)	Avg. speed (pairs/s)	Avg. speed (pairs/s)
Rada [71]	30.43 (15)	0.096 (15)	**644729**(10^7)
AncSPL-Rada (this work)	–	–	**705189**(10^7)
Lin-Seco [87, 110]	140.82 (500)	532913(10^7)	**824307**(10^7)
Wu-Palmer$_{fast}$ [72]	21.34 (15)	–	**717535**(10^7)

Best performing values are shown in bold. Non-implemented methods (–). The number of random CUI pairs evaluated to measure each value is shown between parentheses

Table 8 Average speed in GO concept pairs per second (pairs/s) for the evaluation of two representative ontology-based similarity measures based on the Gene Ontology [1, 2] (2020-05-02 version, 44509 nodes)) implemented by state-of-the-art SML [34] library and HESML

Similarity measure	Measure type	SML	HESML
		Avg. speed (pairs/s)	Avg. speed (pairs/s)
Rada [71]	Edge-counting	0.077 (20)	**3.217** (20)
AncSPL-Rada (this work)	Edge-counting	–	**140422** (10^7)
Lin-Seco [87, 110] IC model	IC-based	372140 (10^7)	**1063219** (10^7)

Best performing values are shown in bold. The number of random GO concept pairs evaluated to measure each value is shown between parentheses

Table 9 Average speed in sentence pairs per second (sent/s) and CUI pairs per second (CUIs/s) for the evaluation of the UBSM [39] sentence similarity measure combined with three representative ontology-based similarity measures based on MeSH (Nov, 2019) in 30 sentence pairs extracted from the MedSTS [135] sentence similarity dataset, and 1 million sentence pairs extracted from BioC corpus [136]

Pairwise sentence comparison based on MeSH	UMLS::Sim (30 pairs)		SML (30 pairs)		HESML (30 pairs)		*HESML* (10^6 *pairs*)	
Similarity measure	Avg. speed (sent/s)	Avg. speed (CUIs/s)	Avg. speed (sent/s)	Avg. speed (CUIs/s)	Avg. speed (sent/s)	Avg. speed (CUIs/s)	Avg. speed (sent/s)	Avg. speed (CUIs/s)
Rada et al. [71]	0.441	36.63	0.126	10.478	**2830.189**	235000	7982.222	337843.826
AncSPL-Rada (this work)	–	–	–	–	**2542.373**	211101.695	7958.742	336850.041
Lin-Seco [87, 110]	0.782	64.956	2586.207	214741.379	**3125**	259479.167	8166.185	345629.98
Wu-Palmer$_{fast}$ [72]	0.181	15.067	–	–	**3125**	259479.167	7892.959	334065.805

We provide the average evaluation in normalized CUI pairs per second to allow a fair and unbiased comparison of the results reported for 30 and 1 million sentence pairs. The dataset with 30 sentence pairs requires 2491 pairwise CUI comparisons, whilst the 1 million sentence pairs dataset requires 42324534 pairwise CUI comparisons. Best performing values are shown in bold. Non-implemented methods (–)

supplementary material to allow the exact replication of all experiments and results introduced herein (see Aditional file 1).

Evaluation of HESML performance. We compare the performance of HESML V1R5 with UMLS::Similarity 1.47 and SML 0.9 libraries, which are the only publicly available semantic measures libraries for SNOMED-CT and MeSH, whilst SML is also the best performing semantic measures library based on GO (see [34, Table 1]). First, we evaluate the average speed of each library, measured in concepts by second, in the evaluation of the semantic similarity of a sequence of randomly generated pairs of UMLS or GO concepts using the SNOMED-CT, MeSH, and GO ontologies as shown in Tables 6, 7 and 8 respectively. Next, we evaluate the average speed of each library, measured in sentences by second as shown in Table 9, in the evaluation of the similarity of a subset of 30 sentence pairs extracted from the MedSTS [135] sentence similarity benchmark, and 1 million sentence pairs extracted from the BioC corpus [136], by implementing the UBSM [39] sentence similarity measure in combination with some ontology-based semantic similarity measures based on MeSH. Table 9 also reports the average speed measured in UMLS Concept Unique Identifier (CUI) pairs per second to compare the results reported for the evaluation of either 30 sentence pairs or 1 million.

Selection of ontology-based similarity measures. We use the Rada et al. [71], Lin [87] and Wu and Palmer [72] similarity measures as a common representative sample to evaluate the performance of the three aforementioned libraries in all our experiments. However, we exclude the evaluation of the Wu-Palmer measure for the SML library because it does not provide the same depth-based version implementation than HESML or UMLS::Similarity. We selected these three similarity measures mentioned above because of several reasons. Firstly, they are implemented by the three libraries analyzed herein, as shown in Table 2. Secondly, Rada et al. measure is a good representative for the family of path-based similarity measures, whilst Lin and Wu-Palmer measures are good representatives for the families of similarity measures based on IC models and taxonomic features, respectively. Third, these three later measures allow evaluating the HESML performance in three graph-based algorithms used by most of ontology-based similarity measures as follows: (1) the computation of the length of the shortest path between concepts; (2) the computation of the Most Informative Common Ancestor (MICA) concept; and (3) the Lowest Common Subsumer (LCS) concept. Fourth, IC-based measures based on a single computation of the MICA concept will exhibit the same performance, such as the measures by Resnik [85], Lin [87], and Jiang-Conrath [86], whilst all path-based using a single computation of the length of the shortest path between concepts will also share the same performance. Finally, current authors showed theoretically [109, Table 3] and experimentally that many ontology-based similarity measures reported in the literature are based on monotone transformations or reformulations of other path-based or IC-based measures. For all the reasons above, the performance results reported herein could be extrapolated to other similar measures based on the same set of graph-based algorithms.

Experimental setup. All our experiments were generated by running a Java console program called HESML_UMLS_benchmark on a Docker container based on UBUNTU 20.04, as detailed in Appendix A (see Additional file 1), which is provided as supplementary material [65] to allow the exact replication of all experiments and results introduced

Table 10 This table shows the Pearson (r) and Spearman (ρ) correlation values between the similarity values returned by a set of path-based similarity measures and those values returned by their reformulation based on the new AncSPL algorithm for a sequence of 1000 random CUI pairs in SNOMED-CT 2019AB, GO (2020-05-02), and WordNet 3.0

Base measure	AncSPL reformulation	50 samples		100 samples		200 samples		1000 samples	
		r	ρ	r	ρ	r	ρ	r	ρ
Correlation values in SNOMED-CT (tree-like$_\sigma = 0.425$)									
Rada [71]	AnsSPL-Rada	0.9214	0.9412	0.9413	0.9444	0.9357	0.9352	0.9231	0.9217
Leacock and Chodorow [73]	AnsSPL-Leacock	0.9409	0.9412	0.9479	0.9444	0.9422	0.9352	0.9217	0.9217
coswJ&C [35]	AnsSPL-coswJ&C	0.9136	0.9506	0.9583	0.9747	0.9761	0.9775	0.941	0.9714
Correlation values in GO (tree-like$_\sigma = 0.446$)									
Rada [71]	AnsSPL-Rada	0.8571	0.8277	0.9133	0.9085	0.8883	0.8868	0.9074	0.8947
Leacock and Chodorow [73]	AnsSPL-Leacock	0.8542	0.8277	0.9109	0.9085	0.9007	0.8868	0.9191	0.8947
coswJ&C [35]	AnsSPL-coswJ&C	0.9679	0.9848	0.9372	0.9894	0.9654	0.9888	0.9533	0.977
Correlation values in WordNet (tree-like$_\sigma = 0.0269$)									
Rada [71]	AnsSPL-Rada	0.9072	0.8882	0.9151	0.8855	0.9225	0.8994	0.9168	0.9038
Leacock and Chodorow [73]	AnsSPL-Leacock	0.9354	0.8882	0.9375	0.8855	0.937	0.8994	0.9345	0.9038
coswJ&C [35]	AnsSPL-coswJ&C	0.9993	0.9906	0.998	0.9916	0.9644	0.9859	0.9815	0.9807

We show the results obtained in the evaluation of the first 50, 100, 200, and 1000 random CUI pairs. All similarity measures are implemented in HESML V1R5 [63]. CoswJ&C [35] sets the current state-of-the-art in the family of ontology-based semantic similarity measures based on WordNet [58]. We define the tree-like deviation (tree-like$_\sigma$) below as the ratio of nodes with multiple parents regarding the overall number of ontology nodes. The tree-like deviation is 0 for MeSH, whilst it is (2213/82115) for WordNet 3.0, (151916/357406) for SNOMED-CT, and (19680/44509) for GO

herein. Because there are large differences in the average speed of each library, especially UMLS::Similarity, we used a different number of concept pairs (samples) per library from the same randomly-generated sequence of UMLS concept (CUI) pairs. Our reproducibility dataset [65] also provides the raw data files obtained in three runs of our experiments. All experiments reported herein are based on HESML V1R5.0.2 release, which is publicly available at HESML GitHub repository[5] and its permanent dataset [63].

Testing our hypothesis for the AncSPL algorithm. Concerning the new AncSPL algorithm, we include the evaluation of the AncSPL-Rada reformulation of the Rada et al. [71] measure in Tables 6, 7, 8 and 9 to compare the performance of the AncSPL-based measures with that obtained by their exact implementations. Finally, to test the second part of our hypothesis H2 on the approximation quality of our AncSPL algorithm, we evaluate the Pearson and Spearman correlation values between the similarity values returned by a set of path-based similarity measures for 50, 100, 200, and 1000 random CUI pairs in SNOMED-CT, GO, and WordNet non-tree-like ontologies and those values returned by their reformulation based on the AncSPL algorithm, as shown in Table 10.

Approximation error of AncSPL. To analyze the absolute approximation error made by AncSPL in the estimation of the exact shortest-path length on non-tree-like ontologies, Fig. 2 shows the cumulative distribution function (CDF) for a set of random samples of

[5] https://github.com/jjlastra/HESML.

Table 11 Overall running time in seconds (s) and average speed in protein pairs per second (prot. pairs/s) obtained by four groupwise GO-based similarity measures (GO, 2020-05-02 version) implemented by HESML in the evaluation of the pairwise protein similarity between the Homo Sapiens and Canis lupus familiaris organisms

Pairwise protein comparison between two large organisms

Measure	**Type**	**HESML Time (s)**	**Avg. speed (prot. pairs/s)**
SimLP [100]	Common ancestors ratio	28243	12038
SimUI [100]	Common ancestor max depth	31922	10651
SimGIC-Seco [105, 110]	IC-based	30754	11055
BMA-Lin-Seco [87, 104, 110]	IC-based	7981	42604

We used the 542193 and 120720 GO annotations for both organisms provided by the "goa_human.gaf" and "go_dog.gaf" files, respectively. Approximately 340 million protein pairs and 33.5×10^9 GO-annotation pairs are compared

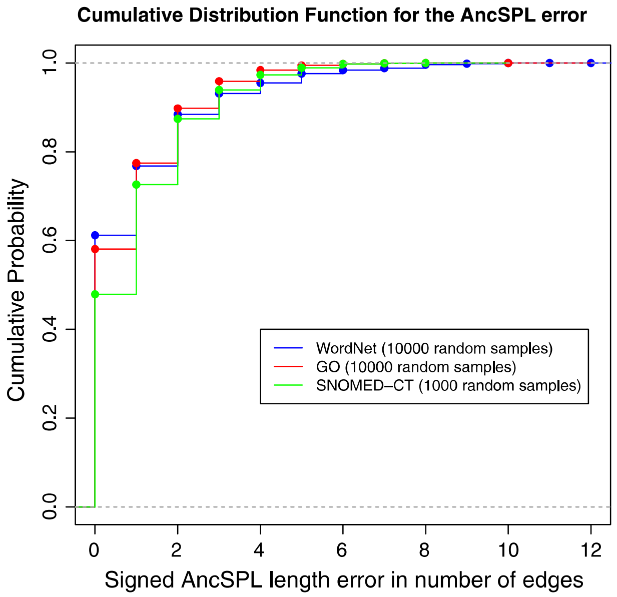


Fig. 2 This figure shows the cumulative distribution function (CDF) of the signed AncSPL length error function $E(c_i, c_j) = AncSPL(c_i, c_j) - spl(c_i, c_j)$, where $spl(c_i, c_j)$ is the exact length of the shortest path between concepts c_i and c_j in SNOMED-CT, GO, and WordNet ontologies

the signed shortest-path length error measured in number of edges in SNOMED-CT, GO, and WordNet.

Testing the AncSPL time complexity. To test experimentally the time complexity of AncSPL, Fig. 3 reports the average running time obtained in evaluating the AncSPL-Rada similarity measure on groups of random concept pairs grouped by the dimension of their corresponding ancestor-based subgraph in SNOMED-CT, GO, and WordNet ontologies, respectively. These experiments evaluate the time complexity of the AncSPL algorithm on non-tree-like taxonomies based on the min-priority queue

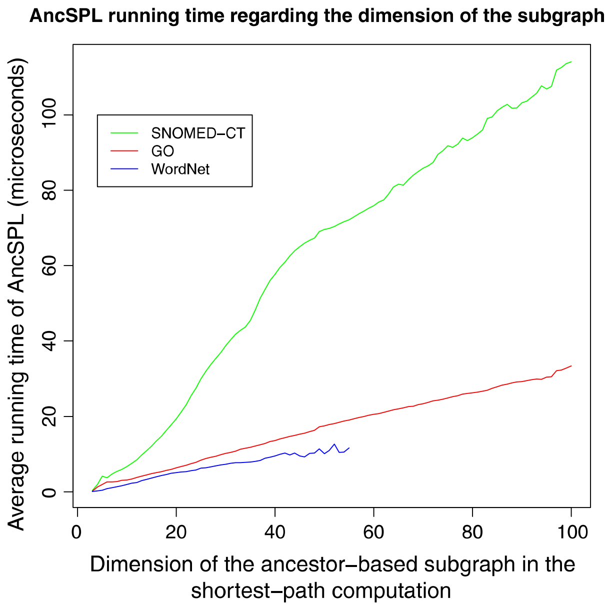


Fig. 3 This figure shows the average running time in micro seconds (μs) obtained in evaluating the AncSPL-Rada similarity measure for groups of at least 10^6 random concept pairs in SNOMED-CT and GO, and at least 10^7 random pairs in WordNet, which are grouped by the dimension of their corresponding ancestor-based subgraph

Table 12 Experimental confirmation of the $k\bar{E}_C^2$ factor impacting the linear scalability of AncSPL for non-tree-like ontologies (TC_3) shown in Fig. 3

Ontology	$\bar{E}_C$	$\widehat{k\bar{E}_C^2}$ (μs)	$\bar{E}_C^2/\bar{E}_{WN}^2$	$\widehat{\bar{E}_C^2/\bar{E}_{WN}^2}$
SNOMED-CT	72.02	1.191	7.79	5.39
GO	31.14	0.3277	1.46	1.48
WordNet (WN)	25.80	0.2210	1	1

First column shows the average number of adjacent nodes per ancestor set for each node in ontology C, denoted by $\bar{E}_C$. Second column shows the estimated value for the factor $\widehat{k\bar{E}_C^2}$ in TC_3 obtained by fitting the scalability plot shown in Fig. 3 to the line $t_{\mu s} = \alpha + (k\bar{E}_C^2)N$. Then, third and fourth columns compare the theoretical and experimental expected ratios between the time complexity (slope) of two different ontologies using WordNet (WN) as baseline

implementation of the Djikstra's algorithm 3 using the PosetHERep taxonomy representation [57], when the input graph is constrained to the corresponding ancestor-based subgraph defined by the AncSPL algorithm 1. Every running time value is measured by evaluating at least 10^6 random concept pairs per group in SNOMED-CT and GO, and at least 10^7 pairs per group in WordNet. Likewise, to test experimentally the impact of the intrinsic scaling factor $k\bar{E}^2_{G_{ij}}$, which scales the linear time complexity of AncSPL in non-tree-like ontologies as defined by TC_3, Table 12 compares the theoretical and experimental values for the expected running-time ratios between ontologies derived from the average number of adjacent nodes per ancestor set $\bar{E}_C$ measured on the ontologies.

Large GO-based similarity evaluation. To show the performance of HESML in a large high-demanding GO-based similarity task, Table 11 shows the performance of four groupwise GO-based similarity measures in the evaluation of the pairwise protein similarity between all proteins of the Homo Sapiens and Canis lupus familaris organisms, using their corresponding protein[6] files in GO annotation file (GAF) file format.

Evaluating HESML real-time capabilities. The performance of real-time applications is measured as the time in which an application should answer to a pre-defined event. The main functionality provided by HESML is the capability to evaluate on-the-fly the semantic similarity between ontology concepts at very high rates measured in concept pairs per second without costly auxiliary data structures, as shown in Tables 6, 7 and 8. This later functionality can be used in other ontology-based semantic similarity tasks, such as the evaluation of biomedical sentence similarity reported in Table 9, or the evaluation of GO-based protein similarity reported in Table 11, among others. Thus, HESML allows the proposal of new real-time biomedical applications demanding either a large number of ontology-based semantic similarity evaluations in a pre-defined fraction of a second or the capability to process large ontology-based annotated data files in a pre-defined time as a measure of their quality of service.

Discussion

HESML outperforms by four orders of magnitude the implementation of the Rada et al. [71] path-based measure of UMLS::Similarity in the MeSH ontology as shown in Tables 7 and 9 . However, UMLS::Similarity implementation of the Rada et al. [71] measure based on caching is roughly three times faster than the HESML real-time implementation in the large SNOMED-CT ontology, as shown in Table 6. On the other hand, HESML outperforms by six and three orders of magnitude the implementation of the Lin [87] IC-based measure of UMLS::Similarity in the SNOMED-CT and MeSH ontologies respectively, as shown in Tables 6, 7 and 9. Finally, HESML outperforms by seven and four orders of magnitude the implementation of the depth-based approximation of the Wu and Palmer [72] measure of UMLS::Similarity in the SNOMED-CT and MeSH ontologies respectively, as shown in Tables 6, 7, and 9 .

HESML outperforms by six, two, and four orders of magnitude the implementation of the Rada et al. [71] path-based measure of SML in the MeSH and GO ontologies as shown in Tables 7, 8 and 9 respectively. In addition, SML is unable to provide a practical implementation of the Rada et al. [71] measure on the large SNOMED-CT ontology, as shown in Table 6. On the other hand, HESML implementation of the Lin [87] IC-based measure is roughly 2.43 times faster than the implementation of SML based on SNOMED-CT as shown in Table 6, as well as a roughly 1.55 times faster on MeSH as shown in Tables 7 and 9 , and roughly 2.86 times faster on GO as shown in Table 8.

The conclusions detailed in the two paragraphs above positively confirms our main hypothesis H1 on the outperformance of HESML on the state-of-the-art semantic measures libraries for the biomedical domain.

Path-based measures based on the new AncSPL algorithm are six and five orders of magnitude faster than their exact implementation in large ontologies with multiple inheritance, such as SNOMED-CT and GO, as shown in Tables 6 and 8 respectively,

[6] http://current.geneontology.org/products/pages/downloads.html.

whilst AncSPL obtains similar performance to the exact implementation on tree-like ontologies like MeSH, as shown in Tables 7 and 9 , because both implementations are identical by definition. On the other hand, the results reported in Table 10 show that the reformulation of any path-based measure using AncSPL is highly correlated both in Pearson and Spearman correlation metrics with their corresponding exact implementations. High Spearman rank correlation values guarantee that any ontology-based task using ranking selection will get similar or almost identical results when AncSPL-based measures are used. Thus, this conclusion endorses the reformulation of any path-based similarity measure using AncSPL to obtain real-time approximations of any path-based measure on large ontologies with multiple inheritance, such as SNOMED-CT, GO, or WordNet. We note that in a very well-known replication of the MC30 [137] similarity benchmark carried-out by Resnik [85, Sect. 3.2], the inter-annotator Pearson correlation was 0.8848 for 30 word pairs, whilst in the most recent building of the SimLex-999 benchmark [138, Sect. 4.1] the inter-annotator Spearman correlation was 0.67 for 999 word pairs. Thus, these two later values are currently considered as reliable upper bounds of any practical estimation method for the semantic similarity between word and concepts, or like Resnik says “This value represents an upper bound on what one should expect from a computational attempt to perform the same task” [85, Sect. 3.2]. For this reason, looking at the values reported in Table 10, we can conclude that there is a high correlation between the exact path-based measures and their AncSPL reformulations.

Finally, the significant performance gain shown in Tables 6, 7, 8 and 9, together with the high-correlation values shown in Table 10, allow to confirm positively our hypothesis H2 on the performance, scalability, and approximation quality of the new AncSPL algorithm.

Groupwise similarity measures based on GO implemented by HESML provide a high average speed in the evaluation of the pairwise protein similarity between two large organisms in a large-scale experiment, as shown in Table 11. Thus, HESML can significantly contribute to improving the performance of any application using GO-based semantic similarity measures. Likewise, HESML opens the possibility of processing large-scale GO annotated data at high computation rates, which could encourage new applications like the similarity-based search of proteins in large GO-annotated databases, among others.

The shortest-path length estimated by AncSPL is always greater or equal to the exact value, as shown in Fig. 2 by the empirical Cumulative Distribution Function (CDF) for SNOMED-CT, GO, and WordNet ontologies, respectively. The signed length error of AncSPL is 0 with a probability of 0.479, 0.581, and 0.612, on SNOMED-CT, GO, and WordNet, respectively. On the other hand, the signed length error of AncSPL is less or equal to 2 with a probability of 0.874, 0.898, and 0.8841, on the three aforementioned ontologies, respectively. Thus, the AncSPL-based reformulations of any path-based similarity measure on non-tree-like ontologies always return a less or equal value than their corresponding base measures evaluated using an exact shortest-path algorithm.

The signed length error of AncSPL decreases with the tree-like deviation (tree-like$_\sigma$), as shown in Fig. 2. It means that lower is the number of concepts with multiple parents, higher is the probability of obtaining an AncSPL length error equals to 0. However, looking at the correlation values reported in Table 10, we can observe that correlation values obtained by the AncSPL-based reformulations in WordNet are not significantly higher than the values obtained in SNOMED-CT and GO as would be expected, with the only exception of the IC-based weighted AncSPL-coswJ&C measure, despite WordNet is close to being a tree-like ontology (tree-like$_\sigma = 0.0269$). The AncSPL-coswJ&C measure obtains the higher correlation values in all ontologies and random samples, as shown in Table 10, with the only exception of the Pearson correlation for 50 concept pairs in SNOMED-CT. We conjecture that AncSPL-coswJ&C is more immune to the AncSPL approximation error than the edge-counting measures because it is defined by the length of the IC-based weighted shortest path between concepts.

The average running time of the AncSPL algorithm is linear regarding the dimension of the ancestor-based subgraph, as predicted by Theorem 1 and shown experimentally in Fig. 3 for SNOMED-CT, GO, and WordNet ontologies, respectively. As pointed out above, the performance of AncSPL depends on the dimension of the common ancestor-based subgraph and the average number of adjacent nodes for the nodes in the common ancestor-based subgraph, and not other factors as the distance between concepts, their depth in the taxonomy, or the ontology size. Likewise, the values in the third and fourth columns of Table 12 confirm that the linear time complexity of AncSPL regarding the dimension of the ancestor-based subgraph is scaled by the factor $\bar{E}^2_{G_{ij}}$. Looking at the third and fourth columns of Table 12, we can see that the ratio between the running-times of GO and WordNet is 1.48, whilst the expected theoretical value is 1.46, and the ratio between SNOMED and WordNet is 5.39, whilst the expected theoretical value is 7.79. These minor differences between the theoretical and experimental values for the scaling factor of TC_3 can be attributed to measurement noise and the removal of non-quadratic factors of $\bar{E}_{G_{ij}}$ to approximate its time complexity. Likewise, we conjecture that the difference is higher for SNOMED than GO, because its scalability plot is noisier, as shown in Fig. 3.

Next developments planned for HESML. As forthcoming activities, we plan to implement further tools and functionality as follows: (1) a R-package to make the HESML functionality accessible from the R program; (2) further GO-based semantic similarity measures; (3) support of further pre-trained word embeddings models for the biomedical domain; and (4) gene clustering methods among others.

Conclusions

We have introduced a new semantic measures library for the biomedical domain called HESML V1R5, which implements the largest set of ontology-based semantic similarity measures and IC models for the SNOMED-CT, MeSH, GO, WordNet and OBO-based ontologies, as well as a new approximated shortest-path algorithm called AncSPL which provides a real-time and highly-correlated reformulation of any path-based semantic similarity measure. Our reproducible experiments show that HESML significantly outperforms current state-of-the-art semantic measures libraries in the real-time evaluation of semantic similarity measures. Likewise, our new aforementioned AncSPL algorithm allows for the first time the real-time evaluation of any path-based semantic measures, such as the large set of measures based on AncSPL which are implemented by HESML V1R5. In addition, we show that AncSPL linearly scales regarding the dimension of the common ancestor subgraph regardless of the ontology size, and the AncSPL reformulations of path-based measures are up to six and five orders of magnitude faster than their exact implementation in SNOMED-CT and GO ontologies, respectively.

The main features of HESML V1R5 are as follows: (1) the implementation of a very large set of semantic similarity methods, IC models, biomedical ontologies, and WordNet, into a single software library; (2) a real-time performance and linear scalability as regards the ontology size; (3) an open and easily extensible architecture based on abstract Java interfaces; and finally, (4) its implementation based on a portable and first-class object-oriented programming language like Java. For this reason, HESML V1R5 is a valuable resource with a huge potential for the development of high-throughput experiments and data-intensive applications in the fields of genomics and biomedical text mining.

As forthcoming activities, we plan to develop a library of sentence similarity measures for a biomedical survey [41], and Python and R interfaces for HESML.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04539-0.

Additional file 1: We provide the Appendix A entitled "The reproducible benchmarks of biomedical semantic measures libraries" as supplementary material in one additional file. Appendix A introduces a detailed experimental setup, which is based on a publicly available reproducibility dataset [65] provided as supplementary material to allow the exact replication of all the experiments and results reported herein, as well as providing the source code of our benchmarks.

Acknowledgements

We are grateful of Fernando González and Juan Corrales for setting up our reproducibility dataset, Hongfang Liu and Yanshan Wang for providing us the MedSTS dataset, Sebastian Hennig for testing our reproducibility protocol. UMLS CUI codes, SNOMED-CT US and MeSH ontologies were used in our experiments by courtesy of the National Library of Medicine of the United States. Finally, we express our most since gratitude to the anonymous reviewers for their valuable suggestions to improve the quality of the paper. This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018). This work introduces new original methods and a significant extension of the methods and software resources previously introduced in the PhD thesis of the first author [139].

Authors' contributions

JLD devised this study and developed the HESML V1R5 library and the new shortest-path method, together with a large part of the experiments, the reproducibility protocol and dataset, and finally, he wrote the manuscript and supervised this study. ALC developed a large part of the experiments, the reproducibility protocol and dataset, and finally, she developed the HESML web site. AGS contributed to the funding and supervision of this study. All authors have read and approved the manuscript.

Funding

This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018). UNED played no role in the design of the study or in the collection, analysis of data, and writing of the manuscript.

Availability of data and materials

In addition to the distribution of the HESML software library detailed below, we also provide a self-contained reproducibility dataset [65], together with a detailed reproducibility protocol introduced in Appendix A (see Additional file 1) to allow the exact replication of all our experiment and results. Project name: HESML. Project home page: http://hesml.lsi.uned.es/, https://github.com/jjlastra/HESML. Community forum: hesml+subscribe@googlegroups.com, hesml+unsubscribe@googlegroups.com. Current version (this work): HESML V1R5 [63]. Operating system(s): Platform independent. Programming language: Java. Other requirements: Java 1.8. License: CC By-NC-SA-4.0. Any restrictions to use by non-academics: no restrictions for non-commercial use. For commercial use of the software, it is needed to contact the authors and/or the UNED technology transfer office.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 21 September 2020 Accepted: 15 December 2021
Published online: 06 January 2022

References

1. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Michael Cherry J, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene Ontology: tool for the unification of biology. Nat Genet. 2000;25(1):25–9.
2. The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. Nucleic Acids Res. 2019;47(D1):330–8.
3. Lord PW, Stevens RD, Brass A, Goble CA. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics. 2003;19(10):1275–83.
4. Freudenberg J, Propping P. A similarity-based method for genome-wide prediction of disease-relevant human genes. Bioinformatics. 2002;18(Suppl 2):110–5.
5. Sevilla JL, Segura V, Podhorski A, Guruceaga E, Mato JM, Martínez-Cruz LA, Corrales FJ, Rubio A. Correlation between gene expression and GO semantic similarity. IEEE/ACM Trans Comput Biol Bioinform. 2005;2(4):330–8.
6. Couto FM, Silva MJ, Coutinho PM. Semantic similarity over the gene ontology: family correlation and selecting disjunctive ancestors. In: Proceedings of the 14th ACM international conference on information and knowledge management. CIKM '05. New York: ACM; 2005. pp. 343–344.
7. Couto FM, Silva MJ, Coutinho PM. Measuring semantic similarity between Gene Ontology terms. Data Knowl Eng. 2007;61(1):137–52.
8. Pesquita C, Faria D, Falcao AO, Lord P, Couto FM. Semantic similarity in biomedical ontologies. PLoS Comput Biol. 2009;5(7):1000443.
9. Guzzi PH, Mina M, Guerra C, Cannataro M. Semantic similarity analysis of protein data: assessment with biological features and issues. Brief Bioinform. 2012;13(5):569–85.
10. Mazandu GK, Chimusa ER, Mulder NJ. Gene Ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. Brief Bioinform. 2016;18(5):886–901.
11. Pesquita C. Semantic similarity in the gene ontology. In: Dessimoz C, Škunca N, editors. Chap. 12. The gene ontology handbook. methods in molecular biology. Cham: Springer; 2017. p. 161–73.
12. Pesquita C, Pessoa D, Faria D, Couto F. CESSM: collaborative evaluation of semantic similarity measures. JB2009: Challenges in Bioinformatics 2009; 157, 190.
13. Cardoso C, Sousa RT, Köhler S, Pesquita C. A collection of benchmark data sets for knowledge graph-based similarity in the biomedical domain. In: Proceedings of the 17th extended semantic web conference (ESWC). 2020.
14. Adie EA, Adams RR, Evans KL, Porteous DJ, Pickard BS. SUSPECTS: enabling fast and effective prioritization of positional candidates. Bioinformatics. 2006;22(6):773–4.
15. Schlicker A, Lengauer T, Albrecht M. Improving disease gene prioritization using the semantic similarity of Gene Ontology terms. Bioinformatics. 2010;26(18):561–7.
16. Börnigen D, Tranchevent L-C, Bonachela-Capdevila F, Devriendt K, De Moor B, De Causmaecker P, Moreau Y. An unbiased evaluation of gene prioritization tools. Bioinformatics. 2012;28(23):3081–8.
17. Bastos H, Faria D, Pesquita C, et al. Using GO terms to evaluate protein clustering. In: Proceedings of the 10th annual bio-ontologies meeting at ISMB/ECCB—15th annual international conference on intelligent systems for molecular biology (ISMB). 2007.
18. Ali W, Deane CM. Functionally guided alignment of protein interaction networks for module detection. Bioinformatics. 2009;25(23):3166–73.
19. Yu H, Jansen R, Stolovitzky G, Gerstein M. Total ancestry measure: quantifying the similarity in tree-like classification, with genomic applications. Bioinformatics. 2007;23(16):2163–73.
20. Tao Y, Sam L, Li J, Friedman C, Lussier YA. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. Bioinformatics. 2007;23(13):529–38.
21. Guo X, Liu R, Shriver CD, Hu H, Liebman MN. Assessing semantic similarity measures for the characterization of human regulatory pathways. Bioinformatics. 2006;22(8):967–73.
22. Couto FM, Silva MJ, Coutinho PM. Implementation of a functional semantic similarity measure between gene-products. Technical Report TR–03–29, Department of Informatics, University of Lisbon. 2003.
23. Wang JZ, Du Z, Payattakool R, Yu PS, Chen C-F. A new method to measure the semantic similarity of GO terms. Bioinformatics. 2007;23(10):1274–81.
24. Du Z, Li L, Chen C-F, Yu PS, Wang JZ. G-SESAME: web tools for GO-term-based gene similarity analysis and knowledge discovery. Nucleic Acids Res. 2009;37(2):345–9.
25. Schlicker A, Albrecht M. FunSimMat: a comprehensive functional similarity database. Nucleic Acids Res. 2008;36(Database issue):434–9.
26. Schlicker A, Albrecht M. FunSimMat update: new features for exploring functional similarity. Nucleic Acids Res. 2010;38(Database issue):244–8.
27. Faria D, Pesquita C, Couto FM, Falcão A. Proteinon: a web tool for protein semantic similarity. Technical Report TR–07–6. Department of Computer Science. Univeristy of Lisbon. 2007.
28. Mazandu GK, Mulder NJ. DaGO-Fun: tool for Gene Ontology-based functional analysis using term information content measures. BMC Bioinform. 2013;14:284.

29. Caniza H, Romero AE, Heron S, Yang H, Devoto A, Frasca M, Mesiti M, Valentini G, Paccanaro A. GOssTo: a standalone application and a web tool for calculating semantic similarities on the Gene Ontology. Bioinformatics. 2014;30(15):2235–6.
30. Chicco D, Masseroli M. Software suite for gene and protein annotation prediction and similarity search. IEEE/ACM Trans Comput Biol Bioinform. 2015;12(4):837–43.
31. Fröhlich H, Speer N, Poustka A, Beissbarth T. GOSim-an R-package for computation of information theoretic GO similarities between terms and gene products. BMC Bioinform. 2007;8:166.
32. Yu G, Li F, Qin Y, Bo X, Wu Y, Wang S. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics. 2010;26(7):976–8.
33. Mazandu GK, Chimusa ER, Mbiyavanga M, Mulder NJ. A-DaGO-Fun: an adaptable gene ontology semantic similarity-based functional analysis tool. Bioinformatics. 2016;32(3):477–9.
34. Harispe S, Ranwez S, Janaqi S, Montmain J. The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. Bioinformatics. 2014;30(5):740–2.
35. Lastra-Díaz JJ, García-Serrano A. A novel family of IC-based similarity measures with a detailed experimental survey on WordNet. Eng Appl Artif Intell. 2015;46:140–53.
36. Harispe S, Ranwez S, Janaqi S, Montmain J. Semantic similarity from natural language and ontology analysis. Synthesis lectures on HLT, vol. 8. San Rafael: Morgan & Claypool publishing; 2015.
37. Pakhomov S, McInnes B, Adam T, Liu Y, Pedersen T, Melton GB. Semantic similarity and relatedness between clinical terms: an experimental study. Proc Annu Symp AMIA. 2010;2010:572–6.
38. McInnes BT, Pedersen T. Evaluating semantic similarity and relatedness over the semantic grouping of clinical term pairs. J Biomed Inform. 2015;54:329–36.
39. Sogancioglu G, Öztürk H, Özgür A. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017;33(14):49–58.
40. Blagec K, Xu H, Agibetov A, Samwald M. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. BMC Bioinform. 2019;20:178.
41. Lara-Clares A, Lastra-Díaz JJ, García-Serrano A. Protocol for a reproducible experimental survey on biomedical sentence similarity. PLoS ONE. 2021;16(3):0248663.
42. McInnes BT, Pedersen T. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. J Biomed Inform. 2013;46(6):1116–24.
43. Caviedes JE, Cimino JJ. Towards the development of a conceptual distance metric for the UMLS. J Biomed Inform. 2004;37(2):77–85.
44. Pedersen T, Pakhomov SVS, Patwardhan S, Chute CG. Measures of semantic similarity and relatedness in the biomedical domain. J Biomed Inform. 2007;40(3):288–99.
45. Batet M, Sánchez D, Valls A. An ontology-based measure to compute semantic similarity in biomedicine. J Biomed Inform. 2011;44(1):118–25.
46. Sánchez D, Batet M. Semantic similarity estimation in the biomedical domain: an ontology-based information-theoretic perspective. J Biomed Inform. 2011;44(5):749–59.
47. Melton GB, Parsons S, Morrison FP, Rothschild AS, Markatou M, Hripcsak G. Inter-patient distance metrics using SNOMED CT defining relationships. J Biomed Inform. 2006;39(6):697–705.
48. Garla VN, Brandt C. Ontology-guided feature engineering for clinical text classification. J Biomed Inform. 2012;45(5):992–8.
49. Mabotuwana T, Lee MC, Cohen-Solal EV. An ontology-based similarity measure for biomedical data—application to radiology reports. J Biomed Inform. 2013;46(5):857–68.
50. Zhu S, Zeng J, Mamitsuka H. Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity. Bioinformatics. 2009;25(15):1944–51.
51. Sarrouti M, Ouatik El Alaoui S. A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering. J Biomed Inform. 2017;68:96–103.
52. Ji X, Ritter A, Yen P-Y. Using ontology-based semantic similarity to facilitate the article screening process for systematic reviews. J Biomed Inform. 2017;69:33–42.
53. McInnes BT. Pedersen T, Pakhomov SVS. UMLS-interface and UMLS-similarity: open source software for measuring paths and semantic similarity. In: Proceedings of the annual symposium of AMIA, vol. 2009. San Francisco, CA; 2009. pp. 431–5.
54. Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004;32(Database issue):267–70.
55. Miller GA. WordNet: a lexical database for English. Commun ACM. 1995;38(11):39–41.
56. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ, OBI Consortium Leontis N, Rocca-Serra P, Ruttenberg A, Sansone S-A, Scheuermann RH, Shah N, Whetzel PL, Lewis S. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol. 2007;25(11):1251–5.
57. Lastra-Díaz JJ, García-Serrano A, Batet M, Fernández M, Chirigati F. HESML: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Inf Syst. 2017;66:97–118.
58. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb MA, García-Serrano A, Ben Aouicha M, Agirre E. A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Eng Appl Artif Intell. 2019;85:645–65.
59. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb MA, García-Serrano A, Ben Aouicha M, Agirre E. Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity. Data Brief. 2019;26:104432.
60. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb M, García-Serrano A, Ben Aouicha M, Agirre E, Sánchez D. A large reproducible benchmark of ontology-based methods and word embeddings for word similarity. Inf Syst. 2021;96:101636.

61. Chen M, Chowdhury RA, Ramachandran V, Roche DL, Tong L. Priority queues and Dijkstra's algorithm. Technical Report TR-07-54, Computer Science Department, University of Texas at Austin. 2007.
62. Lastra-Díaz JJ, García-Serrano A. A new family of information content models with an experimental survey on WordNet. Knowl-Based Syst. 2015;89:509–26.
63. Lastra-Díaz JJ, Lara-Clares A, García-Serrano A. HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models. e-cienciaDatos, v2. 2020. https://doi.org/10.21950/1RRAWJ.
64. Dijkstra EW. A note on two problems in connexion with graphs. Numer Math. 1959;1(1):269–71.
65. Lastra-Díaz JJ, Lara-Clares A, García-Serrano A. Reproducibility dataset for a benchmark of biomedical semantic measures libraries. e-cienciaDatos. 2020. https://doi.org/10.21950/OTDA4Z.
66. Peleg D, Schäffer AA. Graph spanners. J Graph Theory. 1989;13(1):99–116.
67. Althöfer I, Das G, Dobkin D, Joseph D, Soares J. On sparse spanners of weighted graphs. Discrete Comput Geom. 1993;9(1):81–100.
68. Elkin M, Solomon S. Fast constructions of lightweight spanners for general graphs. ACM Trans Algorithms. 2016;12(3):1–21.
69. Banerjee S, Pedersen T. An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Computational linguistics and intelligent text processing. lecture notes in computer science. Springer; 2002. pp. 136–45.
70. Patwardhan S, Pedersen T. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 workshop making sense of sense-bringing computational linguistics and psycholinguistics together. 2006;1501, pp. 1–8.
71. Rada R, Mili H, Bicknell E, Blettner M. Development and application of a metric on semantic nets. IEEE Trans Syst Man Cybern. 1989;19(1):17–30.
72. Wu Z, Palmer M. Verbs semantics and lexical selection. In: Proceedings of the annual meeting of ACL. ACL; 1994. pp. 133–138.
73. Leacock C, Chodorow M. Combining local context and WordNet similarity for word sense identification. In: WordNet: an electronic lexical database, Chap. 11. MIT Press; 1998. pp. 265–283.
74. Stojanovic N, Maedche A, Staab S, Studer R, Sure Y. SEAL: a framework for developing SEmantic PortALs. In: Proceedings of the 1st international conference on knowledge capture (K-CAP). ACM; 2001. pp. 155–62.
75. Maedche A, Staab S. Comparing ontologies-similarity measures and a comparison study. Technical Report 408, Institute AIFB, University of Karlsruhe. 2001.
76. Zhong J, Zhu H, Li J, Yu Y. Conceptual graph matching for semantic search. In: Conceptual structures: integration and interfaces. Springer; 2002. pp. 92–106.
77. Pekar V, Staab S. Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In: Proceedings of COLING, vol. 1. Stroudsburg: ACL; 2002. pp. 1–7.
78. Li Y, Bandar ZA, McLean D. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans Knowl Data Eng. 2003;15(4):871–82.
79. Liu XY, Zhou YM, Zheng RS. Measuring semantic similarity in wordnet. In: Proceedings of the 2007 international conference on machine learning and cybernetics, vol. 6. IEEE; 2007. pp. 3431–3435.
80. Al-Mubaid H, Nguyen HA. Measuring semantic similarity between biomedical concepts within multiple ontologies. IEEE Trans Syst Man Cybern. 2009;39(4):389–98.
81. Kyogoku R, Fujimoto R, Ozaki T, Ohkawa T. A method for supporting retrieval of articles on protein structure analysis considering users' intention. BMC Bioinform. 2011;12 Suppl 1:42.
82. Hao D, Zuo W, Peng T, He F. An approach for calculating semantic similarity between words using WordNet. In: Proceedings of the international conference on digital manufacturing automation. IEEE; 2011. pp. 177–180.
83. Hadj Taieb MA, Ben Aouicha M, Ben Hamadou A. Ontology-based approach for measuring semantic similarity. Eng Appl Artif Intell. 2014;36:238–61.
84. McInnes BT, Pedersen T, Liu Y, Melton GB, Pakhomov SV. U-path: an undirected path-based measure of semantic similarity. In: AMIA ... annual symposium proceedings/AMIA symposium, vol. 2014. AMIA Symposium; 2014. pp. 882–891.
85. Resnik P. Using information content to evaluate semantic similarity in a taxonomy. Proc IJCAI. 1995;1:448–53.
86. Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of international conference research on computational linguistics (ROCLING X); 1997. pp. 19–33.
87. Lin D. An information-theoretic definition of similarity. In: Proceedings of of ICML, vol. 98. Madison, WI; 1998. pp. 296–304.
88. Schlicker A, Domingues FS, Rahnenführer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinform. 2006;7:302.
89. Pirró G, Seco N. Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content. In: On the move to meaningful internet systems: OTM 2008. LNCS, vol. 5332. Springer; 2008. pp. 1271–1288.
90. Pirró G, Euzenat J. A feature and information theoretic framework for semantic similarity and relatedness. In: Proceedings of ISWC. LNCS, vol. 6496. Shangai: Springer; 2010. pp. 615–630.
91. Garla VN, Brandt C. Semantic similarity in the biomedical domain: an evaluation across knowledge sources. BMC Bioinform. 2012;13:261.
92. Meng L, Gu J. A new model for measuring word sense similarity in WordNet. In: Proceedings of the ASTL 4th international conference on advanced communication and networking, vol. 14; 2012. pp. 18–23.
93. Gao JB, Zhang BW, Chen XH. A WordNet-based semantic similarity measurement combining edge-counting and information content theory. Eng Appl Artif Intell. 2015;39:80–8.
94. Cai Y, Zhang Q, Lu W, Che X. A hybrid approach for measuring semantic similarity based on IC-weighted path distance in WordNet. J Intell Inf Syst. 2017;51:1–25.
95. Zhou Z, Wang Y, Gu J. New model of semantic similarity measuring in WordNet. In: Proceedings of the 3rd international conference on intelligent system and knowledge engineering, vol. 1. IEEE; 2008. pp. 256–261.

96. Meng L, Huang R, Gu J. Measuring semantic similarity of word pairs using path and information content. Int J Fut Gener Commun Netw. 2014;7(3):183–94.
97. Sánchez D, Batet M, Isern D, Valls A. Ontology-based semantic similarity: a new feature-based approach. Expert Syst Appl. 2012;39(9):7718–28.
98. Liu H, Hu Z-Z, Wu CH. DynGO: a tool for visualizing and mining of Gene Ontology and its associations. BMC Bioinform. 2005;6:201.
99. Guo X. SemSim. 2008. http://www.bioconductor.org/packages/2.2/bioc/html/SemSim.html.
100. Gentleman R, Falcon S. GOstats. 2009. http://bioconductor.org/packages/2.3/bioc/html/GOstats.html.
101. Ovaska K, Laakso M, Hautaniemi S. Fast gene ontology based clustering for microarray experiments. BioData Min. 2008;1(1):11.
102. Le D-H. UFO: a tool for unifying biomedical ontology-based semantic similarity calculation, enrichment analysis and visualization. PLoS ONE. 2020;15(7):0235670.
103. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13(11):2498–504.
104. Azuaje F, Wang H, Bodenreider O. Ontology-driven similarity approaches to supporting gene functional assessment. In: Proceedings of the ISMB'2005 SIG meeting on bio-ontologies. academia.edu; 2005. pp. 9–10.
105. Pesquita C, Faria D, Bastos H, Falcão A, Couto F. Evaluating GO-based semantic similarity measures. In: Proceedings of 10th annual bio-ontologies meeting, vol. 37; 2007. p. 38.
106. Lee HK, Hsu AK, Sajdak J, Qin J, Pavlidis P. Coexpression analysis of human genes across many microarray data sets. Genome Res. 2004;14(6):1085–94.
107. Mistry M, Pavlidis P. Gene ontology term overlap as a measure of gene functional similarity. BMC Bioinform. 2008;9:327.
108. Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. J Artif Intell Res. 1999;11:95–130.
109. Lastra-Díaz JJ. García-Serrano A. A refinement of the well-founded Information Content models with a very detailed experimental survey on WordNet. Technical Report TR-2016-01, UNED. 2016. http://e-spacio.uned.es/fez/view/bibliuned:DptoLSI-ETSI-Informes-Jlastra-refinement.
110. Seco N, Veale T, Hayes J. An intrinsic information content metric for semantic similarity in WordNet. In: Proceedings of ECAI, vol. 16. Valencia: IOS Press; 2004. pp. 1089–1094.
111. Blanchard E, Harzallah M, Kuntz P. A generic framework for comparing semantic similarities on a subsumption hierarchy. In: Proceedings of ECAI. IOS Press; 2008. pp. 20–24.
112. Zhou Z, Wang Y, Gu J. A new model of information content for semantic similarity in WordNet. In: Proceedings of the second international conference on future generation communication and networking symposia (FGCNS'08), vol. 3. IEEE; 2008. pp. 85–89.
113. Sebti A, Barfroush AA. A new word sense similarity measure in WordNet. In: Proceedings of the international multiconference on computer science and information technology. IEEE; 2008. pp. 369–373.
114. Sánchez D, Batet M, Isern D. Ontology-based information content computation. Knowl-Based Syst. 2011;24(2):297–303.
115. Sánchez D, Batet M. A new model to compute the information content of concepts from taxonomic knowledge. Int J Seman Web Inf Syst (ISWIS). 2012;8(2):34–50.
116. Meng L, Gu J, Zhou Z. A new model of information content based on concept's topology for measuring semantic similarity in WordNet. Int J Grid Distrib Comput. 2012;5(3):81–93.
117. Yuan Q, Yu Z, Wang K. A new model of information content for measuring the semantic similarity between concepts. In: Proceedings of the intlernational conference on cloud computing and big data (CloudCom-Asia 2013). IEEE Computer Society; 2013. pp. 141–146.
118. Hadj Taieb MA, Ben Aouicha M, Ben Hamadou A. A new semantic relatedness measurement using WordNet features. Knowl Inf Syst. 2014;41(2):467–97.
119. Adhikari A, Singh S, Dutta A, Dutta B. A novel information theoretic approach for finding semantic similarity in WordNet. In: Proceedings of IEEE international technical conference. Macau: IEEE; 2015. pp. 1–6.
120. Ben Aouicha M, Hadj Taieb MA. Computing semantic similarity between biomedical concepts using new information content approach. J Biomed Inform. 2016;59:258–75.
121. Ben Aouicha M, Hadj Taieb MA, Ben Hamadou A. Taxonomy-based information content and wordnet-wiktionary-wikipedia glosses for semantic relatedness. Appl Intell. 2016;45:1–37.
122. Sommer C. Shortest-path queries in static networks. ACM Comput Surv. 2014;46(4):1–31.
123. Madkour A, Aref WG, Rehman FU, Rahman MA, Basalamah S. A survey of shortest-path algorithms. 2017. arXiv:1705.02044.
124. Zwick U. Exact and approximate distances in graphs—a survey. In: Proceedings of the European symposium on algorithms. LNCS, vol. 1261. Berlin: Springer; 2001. pp. 33–48.
125. Agirre E, Soroa A. Personalizing pagerank for word sense disambiguation. In: Proceedings of the EACL. Stroudsburg: ACL; 2009. pp. 33–41.
126. Camacho-Collados J, Pilehvar MT, Navigli R. Nasari: integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artif Intell. 2016;240:36–64.
127. Mrkšić N, Vulić I, Séaghdha DÓ, Leviant I, Reichart R, Gašić M, Korhonen A, Young S. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. Trans ACL. 2017;5:309–24.
128. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. 2016. arXiv:1607.04606.
129. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Proc EMNLP. 2014;12:1532–43.
130. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 2013. arXiv:1301.3781.

131. Schwartz R, Reichart R, Rappoport A. Symmetric pattern based word embeddings for improved word similarity prediction. In: Proceedings of the conference on computational natural language learning; 2015. pp. 258–267.
132. Wieting J, Bansal M, Gimpel K, Livescu K, Roth D. From paraphrase database to compositional paraphrase model and back. Trans ACL. 2015;3:345–58.
133. Mrkšić N, Ó Séaghdha D, Thomson B, Gašić M, Rojas-Barahona L, Su P-H, Vandyke D, Wen T-H, Young S. Counter-fitting word vectors to linguistic constraints. In: Proceedimgs of HLT-NAACL. 2016.
134. Goikoetxe, J, Soroa A, Agirre E. Random walks and neural network language models on knowledge bases. In: Proceedings of HLT-NAACL; 2015. pp. 1434–1439.
135. Wang Y, Afzal N, Fu S, Wang L, Shen F, Rastegar-Mojarad M, Liu H. MedSTS: a resource for clinical semantic textual similarity. Lang Resour Eval. 2018;1–16.
136. Comeau DC, Wei C-H, Islamaj Dogan R, Lu Z. PMC text mining subset in BioC: about 3 million full text articles and growing. Bioinformatics. 2019.
137. Miller GA, Charles WG. Contextual correlates of semantic similarity. Lang Cognit Process. 1991;6(1):1–28.
138. Hill F, Reichart R, Korhonen A. SimLex-999: evaluating semantic models with (genuine) similarity estimation. Comput Linguist. 2015;41(4):665–95.
139. Lastra-Díaz JJ. Recent advances in ontology-based semantic similarity measures and information content models based on WordNet. Universidad Nacional de Educación a Distancia (UNED). 2017. http://e-spacio.uned.es/fez/view/tesisuned:ED-Pg-SisInt-Jjlastra.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chapter 9

Plos One Registered Report article

PLOS ONE

RESEARCH ARTICLE

A reproducible experimental survey on biomedical sentence similarity: A string-based method sets the state of the art

Alicia Lara-Clares*, Juan J. Lastra-Díaz, Ana Garcia-Serrano

NLP & IR Research Group, E.T.S.I. Informática, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

* alara@lsi.uned.es

OPEN ACCESS

Citation: Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A (2022) A reproducible experimental survey on biomedical sentence similarity: A string-based method sets the state of the art. PLoS ONE 17(11): e0276539. https://doi.org/10.1371/journal.pone.0276539

Editor: Usman Qamar, National University of Sciences and Technology (NUST), PAKISTAN

Received: February 21, 2022

Accepted: October 8, 2022

Published: November 21, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0276539

Abstract

This registered report introduces the largest, and for the first time, reproducible experimental survey on biomedical sentence similarity with the following aims: (1) to elucidate the state of the art of the problem; (2) to solve some reproducibility problems preventing the evaluation of most current methods; (3) to evaluate several unexplored sentence similarity methods; (4) to evaluate for the first time an unexplored benchmark, called Corpus-Transcriptional-Regulation (CTR); (5) to carry out a study on the impact of the pre-processing stages and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods; and finally, (6) to bridge the lack of software and data reproducibility resources for methods and experiments in this line of research. Our reproducible experimental survey is based on a single software platform, which is provided with a detailed reproducibility protocol and dataset as supplementary material to allow the exact replication of all our experiments and results. In addition, we introduce a new aggregated string-based sentence similarity method, called LiBlock, together with eight variants of current ontology-based methods, and a new pre-trained word embedding model trained on the full-text articles in the PMC-BioC corpus. Our experiments show that our novel string-based measure establishes the new state of the art in sentence similarity analysis in the biomedical domain and significantly outperforms all the methods evaluated herein, with the only exception of one ontology-based method. Likewise, our experiments confirm that the pre-processing stages, and the choice of the NER tool for ontology-based methods, have a very significant impact on the performance of the sentence similarity methods. We also detail some drawbacks and limitations of current methods, and highlight the need to refine the current benchmarks. Finally, a notable finding is that our new string-based method significantly outperforms all state-of-the-art Machine Learning (ML) models evaluated herein.

Introduction

Measuring semantic similarity between sentences is an important task in the fields of Natural Language Processing (NLP), Information Retrieval (IR), and biomedical text mining, among

Data Availability Statement: All the necessary information for reproducing the experiments is available in the Spanish Dataverse Network permanent repositories [1] and [2]. The pre-installed software for executing the experiments is published as a Docker image in DockerHub: https://hub.docker.com/r/alicialara/hesml_v2r1, which is also publicly available in [1]. The DockerHub repository is offered as an alternative to obtain the pre-installed software, but it is not mandatory for reproducing the results reported in this work. The source code of HESML-V2R1 is also published in Github: https://github.com/jjlastra/HESML/releases/tag/Release_HESML_V2R1. We have also created a permanent branch with all the necessary code for reproducing the results (https://github.com/jjlastra/HESML/tree/HESML_STS_paper_experiments). Also, these files have been uploaded to the permanent repository [1]. Finally, the protocol for setting up and running our experiments is published on [3] https://www.protocols.io/view/a-reproducibility-protocol-and-dataset-on-the-biomb5ckq2uw. [1] Lara-Clares, Alicia; Lastra-Díaz, Juan J.; Garcia-Serrano, Ana, 2021, "Reproducible experiments on word and sentence similarity measures for the biomedical domain", https://doi.org/10.21950/EPNXTR, e-cienciaDatos, V2 [2] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A., 2022, "HESML V2R1 Java software library of semantic similarity measures for the biomedical domain", https://doi.org/10.21950/AQLSMV, e-cienciaDatos, V2 [3] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. A reproducibility protocol and dataset on the biomedical sentence similarity; 2022. Protocols.io, v1. https://www.protocols.io/view/a-reproducibility-protocol-and-dataset-on-the-biom-b5ckq2uw. Special licensing restrictions: We provide a reproducibility dataset that contains all the source code, pre-trained models, pre-installed software, and raw input and output data files together with a reproducibility protocol published at protocols.io. However, to reproduce the experiments evaluated in this work, it is necessary to obtain a license from the National Library of Medicine (NLM) of the United States, which allows the use of the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies and the NER tools. For this reason, any reader has full access to all our software and data, but they should go to the NLM license page (https://uts.nlm.nih.gov/license.html) to obtain a license to use the software and data resources mentioned above. Once the readers have obtained the license from the NLM, they should write to eciencia@consorciomadrono.es to obtain the password to decrypt the file, or they can download all the required libraries from the NLM website and

others. For instance, the estimation of the degree of semantic similarity between sentences is used in text classification [1–3], question answering [4, 5], evidence sentence retrieval to extract biological expression language statements [6, 7], biomedical document labeling [8], biomedical event extraction [9], named entity recognition [10], evidence-based medicine [11, 12], biomedical document clustering [13], prediction of adverse drug reactions [14], entity linking [15], document summarization [16, 17] and sentence-driven search of biomedical literature [18], among other applications. In the question answering task, Sarrouti and El Alaomi [4] build a ranking of plausible answers by computing the similarity scores between each biomedical question and the candidate sentences extracted from a knowledge corpus. Allot et al. [18] introduce a system to retrieve the most similar sentences in the BioC biomedical corpus [19] called Litsense [18], which is based on the comparison of the user query with all sentences in the aforementioned corpus. Likewise, the relevance of the research in this area is endorsed by the proposal of recent conference series, such as SemEval [20–25] and BioCreative/OHNLP [26], and studies based on sentence similarity measures, such as the work of Aliguliyev [16] in automatic document summarization, which shows that the performance of these applications depends significantly on the sentence similarity measures used.

The aim of any semantic similarity method is to estimate the degree of similarity between two textual semantic units as perceived by a human being, such as words, phrases, sentences, short texts, or documents. Unlike sentences from the language in general use whose vocabulary and syntax is limited both in extension and complexity, most sentences in the biomedical domain are comprised of a huge specialized vocabulary made up of all sorts of biological and clinical terms, in addition to innumerable acronyms, which are combined in complex lexical and syntactical forms.

Currently, there are several papers in the literature that experimentally evaluate multiple methods on biomedical sentence similarity. However, they are either theoretical or have a limited scope and cannot be reproduced. For instance, Kalyan et al. [27], Khattak et al. [28], and Alsentzer et al. [29] introduce theoretical surveys on biomedical word and sentence embeddings with a limited scope. On the other hand, the experimental surveys introduced by Sogancioglu et al. [30], Blagec et al. [31], Peng et al. [32], and Chen et al. [33] among other authors, cannot be reproduced because of the lack of source code and data to replicate both methods and experiments, or the lack of a detailed definition of their experimental setups. For instance, Sogancioglu et al. [30] provide the BIOSSES evaluation dataset evaluated in this work, as well as a Demo application and the source code used in their biomedical sentence similarity dataset (https://tabilab.cmpe.boun.edu.tr/BIOSSES/About.html); however, they do provide neither the MetaMap [34] annotation tool and UMLS ontology subsets MeSH [35] and OMIM [36] versions to reproduce the ontology-based measures nor the Open Access Subset of PubMed Central (http://www.ncbi.nlm.nih.gov/pmc/) dataset used in their training stage. Blagec et al. [31] introduce a comprehensive experimental survey for biomedical sentence similarity measures, providing the detailed hyper-parameters used for training the models, as well as several code and data to allow the training and evaluation of their methods (https://github.com/kathrinblagec/neural-sentence-embedding-models-for-biomedical-applications); however, they provide neither the post-processed biomedical dataset used in their training phase, nor the pre-trained models. Peng et al. [32] provide the pre-trained models and pre-processed dataset used to train the models (https://github.com/ncbi-nlp/BLUE_Benchmark), but they do not provide detailed information about the pre-processing of the dataset. Finally, Chen et al. [33] provide the pre-trained models (https://github.com/ncbi-nlp/BioSentVec) but provide neither the detailed information about the data used for training the models nor the information on the pre-processing stage. Therefore, it is not possible to evaluate their results in our experiments. Likewise, there are other recent works whose results need to be confirmed. For

add all the external libraries manually. Likewise, they should obtain and sign a Data User Agreement from the Mayo Clinic (https://n2c2.dbmi.hms.harvard.edu/data-use-agreement) to use the MedSTS dataset by sending the Data User Agreement form to the authors of MedSTS.

Funding: This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018) and the CLARA-HD (PID2020-116001RB-C32) project. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study."

Competing interests: NO authors have competing interests.

instance, Tawfik and Spruit [37] experimentally evaluate a set of pre-trained language models, whilst Chen et al. [38] propose a system to study the impact of a set of similarity measures on a Deep Learning ensemble model, which is based on a Random Forest model [39].

The main aim of this work is to introduce a comprehensive and very detailed reproducible experimental survey of methods on biomedical sentence similarity to elucidate the state of the problem by implementing our previous registered report protocol [40]. Our experiments are based on our software implementation and evaluation of all methods analyzed herein into a common and new software platform based on an extension of the Half-Edge Semantic Measures Library (HESML) [41, 42], called HESML (http://hesml.lsi.uned.es) for Semantic Textual Similarity (HESML-STS). All our experiments have been recorded into a Docker virtualization image that is provided as supplementary material together with our software [43] and a detailed reproducibility protocol [44] and dataset [43] to allow the easy replication of all our methods, experiments, and results. This work is based on our previous experience developing reproducible research in a series of publications in the area, such as the experimental surveys on word similarity introduced in [45–48], whose reproducibility protocols and datasets [49, 50] are detailed and independently confirmed in two companion reproducible papers [41, 51], and a reproducible benchmark on semantic measures libraries for the biomedical domain [42]. Finally, we refer the reader to our previous work [40] for a very detailed review of the literature on sentence similarity measures, which is omitted here because of the lack of space and to avoid repetition.

Main motivations and research questions

Our main motivation is the lack of a comprehensive and reproducible experimental survey on biomedical sentence similarity that allows state of the problem to be set out in a sound and reproducible way, as detailed in our previous registered report protocol [40]. Our main research questions are as follows:

RQ1 Which methods get the best results on biomedical sentence similarity?

RQ2 Is there a statistically significant difference between the best-performing methods and the remaining ones?

RQ3 What is the impact of the biomedical Named Entity Recognition (NER) tools on the performance of the methods on biomedical sentence similarity?

RQ4 What is the impact of the pre-processing stage on the performance of the methods on biomedical sentence similarity?

RQ5 What are the main drawbacks and limitations of current methods on biomedical sentence similarity?

A second motivation is implementing a set of unexplored methods based on adaptations from other methods proposed for the general language domain. A third motivation is the evaluation in the same software platform of the three known benchmarks on biomedical sentence similarity reported in the literature as follows: the Biomedical Semantic Similarity Estimation System (BIOSSES) [30] and Medical Semantic Textual Similarity (MedSTS) [52] datasets, as well as the evaluation for the first time of the Microbial Transcriptional Regulation (CTR) [53] dataset in a sentence similarity task, despite it having been previously evaluated in other related tasks, such as the curation of gene expressions from scientific publications [54]. A fourth motivation is a study on the impact of the pre-processing stage and NER tools on the performance of the sentence similarity methods, such as that done by Gerlach et al. [55] for stop-words in a topic modeling task. And finally, our fifth motivation is the lack of reproducibility software and data resources on this task, which allow an easy replication and confirmation of previous methods, experiments, and results in this line of research, as well as encouraging the development and evaluation of new sentence similarity methods.

Definition of the problem and contributions

The two main research problems tackled in this work are the design and implementation of a large and reproducible experimental survey on sentence similarity measures for the biomedical domain, and the evaluation of a set of unexplored methods based on adaptations from previous methods used in the general language domain. Our main contributions are as follows: (1) the largest, and for the first time, reproducible experimental survey on biomedical sentence similarity; (2) the first collection of self-contained and reproducible benchmarks on biomedical sentence similarity; (3) the evaluation of a set of previously unexplored methods, such as a new string-based sentence similarity method, based on Li et al. [56] and Block distance [57], eight variants of the current ontology-based methods from the literature based on the work of Sogancioglu et al. [30], and a new pre-trained Word Embedding (WE) model based on FastText [58] and trained on the full-text of articles in the PMC-BioC corpus [19]; (4) the evaluation for the first time of an unexplored benchmark, called CTR [53]; (5) the study on the impact of the pre-processing stage and Named Entity Recognition (NER) tools on the performance of the sentence similarity methods; (6) the integration for the first time of most sentence similarity methods for the biomedical domain into the same software library, called HESML-STS, which is available both on Github (https://github.com/jjlastra/HESML) and in a reproducible dataset [43]; (7) a detailed reproducibility protocol together with a collection of software tools and datasets provided as supplementary material to allow the exact replication of all our experiments and results; and finally, (8) an analysis of the drawbacks and limitations of the current state-of-the-art methods.

The rest of the paper is structured as follows. First, we introduce a collection of new sentence similarity methods evaluated here for the first time. Next, we describe a detailed experimental setup for our experiments on biomedical sentence similarity and introduce our experimental results. Then, we discuss our results and answer the research questions detailed above. Subsequently, we introduce our conclusions and future work. Finally, we introduce three appendices with supplementary material as follows. S1 Appendix introduces all statistical significance results of our experiments, whilst S2 Appendix introduces all data tables reporting the performance of all methods with all pre-processing configurations evaluated herein, and the S3 Appendix introduces a reproducibility protocol detailing a set of step-by-step instructions to allow the exact replication of all our experiments, which is published at protocols.io [44].

The new sentence similarity methods

This section introduces a new string-based sentence similarity method based on the aggregation of the Li et al. [56] similarity and Block distance [57] measures, called LiBlock, as well as eight new variants of the ontology-based methods proposed by Sogancioglu et al. [30], and a new pre-trained word embedding model based on FastText [58] and trained on the full-text of the articles in the PMC-BioC corpus [19].

The new LiBlock string-based method

Two key advantages of the family of string-based methods are as follows. Firstly, they can be very efficiently computed because they do not require the use of external knowledge or pre-trained models, and secondly, they obtain competitive results as shown in Table 8. However, the string-based methods do not capture the semantics of the words in the sentence, which prevent them from recognizing semantic relationships between words, such as synonymy and meronymy among others. In contrast, the family of ontology-based methods capture the semantic relationships between words in a sentence pair and obtain state-of-the-art results in the sentence similarity task for the biomedical domain, as shown in Table 8. However, the effectiveness of ontology-based methods depends on the lexical coverage of the ontologies and the ability to recognize automatically the underlying concepts in sentences by using Named Entity Recognition (NER) and Word Sense Desambiguation (WSD) tools, whose coverage and performance could be limited in several application domains. In fact, the NER task is still an open problem [59] in the biomedical domain because of the vast biomedical vocabulary and the complex lexical and syntactic forms found in the biomedical literature. In comparison, the methods based on pre-trained word embedding models provide a broader lexical coverage than the ontology-based ones and obtain better results. However, the methods based on word embedding do not significantly outperform all ontology-based measures in a word similarity task [48] in addition to requiring a large corpus for training, a complex training phase, and more computational resources than the families of string-based and ontology-based methods.

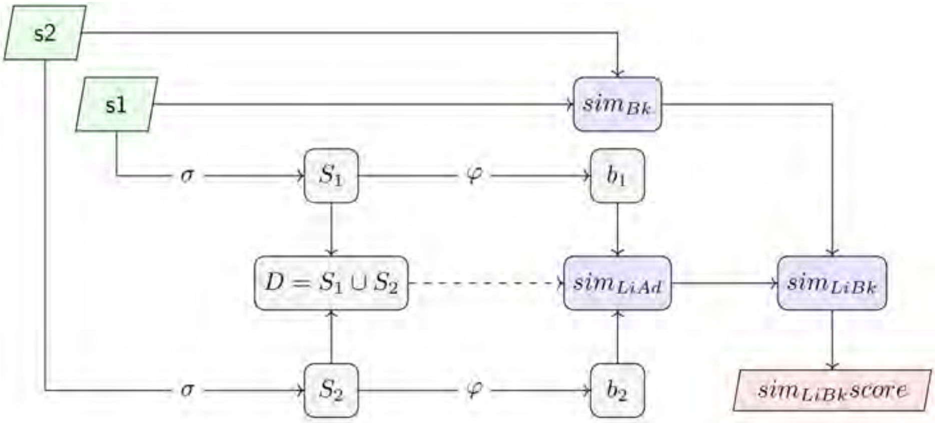
To overcome the drawbacks and limitations of the string-based and ontology-based methods detailed above, we propose here a new aggregated string-based measure called LiBlock and denoted by sim_{LiBk} henceforth, which is based on the combination of a similarity measure derived from the Block Distance [57] and an adaptation from the ontology-based similarity measure introduced by Li et al. [56] that removes the use of ontologies, such as WordNet [60] or Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [61]. The LiBlock similarity measure obtains the best results in combination with the cTAKES NER tool [62], which allows the detection of synonyms of CUI concepts. Nevertheless, the LiBlock method obtains competitive results regarding the state-of-the-art methods with no use, either implicitly or explicitly, of an ontology, as detailed in Table 12.

The sim_{LiBk} method detailed in Eq (1) is defined by the linear aggregation of an adaptation of the Li et al. [56] measure, called sim_{LiAd} (Eq (3)), and a similarity measure derived from the Block Distance measure [57], called sim_{Bk} (Eq (2)). Let be L_Σ the set of word sequences in a universal unseen alphabet Σ, the sim_{LiBk} function returns a value between 0 and 1 which indicates the similarity score between two input sentences, as defined in Eq 1. The sim_{Bk} function is based on the computation of the word frequencies $fr(w_i, s_j)$ for each input sentence s_1 and s_2 and their concatenation s_1+ s_2, as detailed in equation (Eq (2)). The auxiliary function $fr(w_i, s_j)$ returns the frequency of a word w_i in the word sequence s_j, whilst the function $fr(w_i, s_1+ s_2)$ returns the number of occurrences of the word w_i in the concatenation of the two word sequences, denoted by $s_1 + s_2$. On the other hand, the sim_{LiAd} function takes two word sets obtained by invoking the σ function (Eq (5)) with the sentences s_1 and s_2, and then it computes the cosine similarity of the two binary semantic vectors corresponding to invoke the $\varphi(S_1)$ function (Eq (4)) with the $\sigma(s_1)$ and $\sigma(s_2)$ word sets. Finally, the sim_{LiBk} score is defined by either the linear combination of sim_{Bk} and sim_{LiAd}, as detailed in Eq (1), or sim_{Bk} if sim_{LiAd} is 0.

A walk-through example. Algorithm 1 details the step-by-step procedure to compute the sim_{LiBk} function, whilst Fig 1 shows the pipeline for calculating the LiBlock similarity score defined in Eq 1, as well as an example for illustrating an end-to-end calculation of the sim_{LiBk} similarity score of two sentences.

Algorithm 1 LiBlock sentence similarity measure for two input pre-processed sentences.

```
function: simLiBlock ($s_1$, $s_2$)          ▷ being $s_1$, $s_2$ word sequences ∈ $L_\Sigma$
    $S_1 \leftarrow \sigma(s_1)$          ▷ word set sentence 1
```



Input : Raw $s_1 \leftarrow$ "Lung tumour formation in mice by oncogenic KRAS requires formation Craf, but not Braf."
Raw $s_2 \leftarrow$ "The oncogenic activity of mutant Kras appears dependent" on functional Craf but not on Braf."

step 1: $s_1 \leftarrow$ {c0280089, formation, mice, oncogenic, c1537502, requires, formation, craf, c0812241}
$s_2 \leftarrow$ {oncogenic, activity, mutant, c1537502, appears, dependent, functional, craf, c0812241}

step 2: $S_1 \leftarrow$ {c0280089, formation, mice, oncogenic, c1537502, requires, craf, c0812241}

step 3: $S_2 \leftarrow$ {oncogenic, activity, mutant, c1537502, appears, dependent, functional, craf, c0812241}

step 4: $D \leftarrow$ {c0280089, formation, mice, oncogenic, c1537502, requires, craf, c0812241, activity, mutant, appears, dependent, functional}

step 5: $b_1 \leftarrow \{1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0\}$

step 6: $b_2 \leftarrow \{0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1\}$

step 7: $sim_{LiAd} \leftarrow 0.471$

step 8: $sim_{Bk} \leftarrow 0.444$

step 9: $sim_{LiBk} \leftarrow 0.458$

Fig 1. This figure details the workflow for computing the new LiBlock measure and an example illustrating a use case of the workflow following the steps defined in algorithm 1.

https://doi.org/10.1371/journal.pone.0276539.g001

```
3:  $S_2 \leftarrow \sigma(s_2)$                 ▷ word set sentence 2
4:  $D \leftarrow S_1 \cup S_2$                ▷ construct the dictionary $D$
5:  $b_1 \leftarrow \varphi(S_1)$               ▷ construct the semantic binary vector $b_1$
6:  $b_2 \leftarrow \varphi(S_2)$               ▷ construct the semantic binary vector $b_2$
7:  $score_{LiAd} \leftarrow sim_{LiAd}(b_1, b_2)$        ▷ compute LiAdapted similarity
8:  $score_{Bk} \leftarrow sim_{Bk}(s_1, s_2)$          ▷ compute Block Distance similarity
9:  $score_{LiBk} \leftarrow sim_{LiBk}(score_{LiAd}, score_{Bk})$     ▷ compute LiBlock similarity
10:   return $score_{LiBk}$
```

```
end function
```

$$\begin{aligned} & & \text{(LiBlock similarity)} \\ sim_{LiBk} &: L_\Sigma \times L_\Sigma \rightarrow [0,1] \subset \mathbb{R},\ L_\Sigma = \{\text{word sequences in alphabet } \Sigma\} \\ sim_{LiBk}(s_1, s_2) &= \begin{cases} sim_{Bk}(s_1, s_2), & \text{if } sim_{LiAd}(\sigma(s_1), \sigma(s_2)) = 0 \\ \frac{1}{2} sim_{Bk}(s_1, s_2) + \frac{1}{2} sim_{LiAd}(\sigma(s_1), \sigma(s_2)), & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

$$\begin{aligned} sim_{Bk} &: L_\Sigma \times L_\Sigma \rightarrow [0,1] \subset \mathbb{R}, \qquad \text{(Block distance)} \\ sim_{Bk}(s_1, s_2) &= 1 - \frac{\sum_{i=1}^{|D|} |fr(w_i, s_1) - fr(w_i, s_2)|}{\sum_{i=1}^{|D|} fr(w_i, s_1 + s_2)}, \quad D = \sigma(s_1) \cup \sigma(s_2) \in \mathcal{P}(\Sigma) \end{aligned} \tag{2}$$

$$\begin{aligned} sim_{LiAd} &: \mathcal{P}(D) \times \mathcal{P}(D) \rightarrow [0,1] \subset \mathbb{R}, \qquad \text{(Li's score adaptation)} \\ sim_{LiAd}(S_1, S_2) &= \frac{\varphi(S_1) \cdot \varphi(S_2)}{||\varphi(S_1)|| * ||\varphi(S_2)||} \end{aligned} \tag{3}$$

$$\begin{aligned} \varphi &: \mathcal{P}(D) \rightarrow \{0,1\}^{|D|}, \qquad \text{(binary vector constructor)} \\ \varphi(S) &= (b_1, b_2, \ldots, b_{|D|}), \quad b_i = \begin{cases} 1, w_i \in D \\ 0, w_i \notin D \end{cases} \end{aligned} \tag{4}$$

$$\begin{aligned} \sigma &: L_\Sigma \rightarrow \mathcal{P}(\Sigma), \qquad \text{(word set generator)} \\ \sigma(s) &= \{w \in \Sigma : \exists k \in [1, \text{length}(s)] \text{ such that } s_k = w\} \end{aligned} \tag{5}$$

The eight new variants of current ontology-based methods

The current family of ontology-based methods for biomedical sentence similarity proposed by Sogancioglu et al. [30] is based on the ontology-based semantic similarity between word and concepts within the sentences to be compared. Thus, this later family of methods defines a framework in which we can design new variants by exploring other word similarity measures. For this reason, we propose here the evaluation of a set of new ontology-based sentence similarity measures based on two different unexplored notions as follows: (1) the evaluation of state-of-the-art word similarity measures from the general domain [48] not evaluated in the biomedical domain yet; and (2) the evaluation of several ontology-based word similarity measures based on a recent and very efficient shortest-path algorithm, called Ancestors-based Shortest-Path Length (AncSPL) [42], which is a fast approximation of the Dijkstra's algorithm [63] for taxonomies that is introduced with the first HESML version for the biomedical domain [42].

Thus, we propose here the evaluation based on the combination of WBSM and UBSM methods with the path-based word similarity methods as follows: WBSM-Rada (M7); WBSM-cosJ&C (M9); WBSM-coswJ&C (M10); WBSM-Cai (M11); UBSM-Rada (M12); UBSM-cosJ&C (M14); UBSM-coswJ&C (M15); and UBSM-Cai (M16). The detailed information about this later method is shown in Table 3.

The new pre-trained word embedding model

Current sentence similarity methods based on the evaluation of pre-trained embedding models are mostly trained using PubMed Central (PMC) Open Access dataset (https://www.ncbi.nlm.nih.gov/labs/pmc/), or Medical Information Mart for Intensive Care (MIMIC-III) clinical notes [64]. However, as far as we know, there are no models in the literature trained on the full text of the articles in the PMC-BioC corpus [19]. Therefore, we propose evaluating a new FastText [58] word embedding model trained on the aforementioned BioC corpus. FastText overcomes one significant limitation of other methods, such as word2vec [65] and GloVe [66], which ignore the morphology of words by assigning a vector to each word in the vocabulary. For a more detailed review of the family of word embedding methods, we refer the authors to the recent reproducible survey by Lastra-Díaz et al. [48]. The configuration parameters for training this model are detailed in Table 4, and all the necessary information and resources for evaluating it are available in our reproducibility dataset [43], as detailed in Table 6.

The reproducible experimental survey

This section introduces a detailed experimental setup to evaluate and compare all the sentence similarity methods for the biomedical domain proposed in our primary work [40], together with the new methods introduced herein. The main aims of our experiments are as follows: (1) the evaluation of most of known methods for biomedical sentence similarity on the three biomedical datasets shown in Table 1, and implemented on the same software platform; (2) the evaluation of a set of new sentence similarity methods adapted from their definitions for the general-language domain; (3) the evaluation of a new sentence method called LiBlock introduced in this work, eight variants of the current ontology-based methods from the literature based on the work of Sogancioglu et al. [30], and a new word embedding model based on FastText and trained on the full-text of articles in the PMC-BioC corpus [19]; (4) the setting out of the state of the art of the problem in a sound and reproducible way; (5) the replication and independent confirmation of previously reported methods and results; (6) a

Table 1. Benchmarks on biomedical sentence similarity evaluated in this work.

Dataset	#pairs	Corresponding file (*.tsv) in HESML-STS distribution
BIOSSES [30]	100	BIOSSESNormalized.tsv
MedSTS [52]	1,068	CTRNormalized_averagedScore.tsv
CTR [53]	170	MedStsFullNormalized.tsv

https://doi.org/10.1371/journal.pone.0276539.t001

study on the impact of different pre-processing configurations on the performance of the sentence similarity methods; (7) a study on the impact of different Name Entity Recognition (NER) tools, such as MetaMap [34] and clinical Text Analysis and Knowledge Extraction System (cTAKES) [62], on the performance of the sentence similarity methods; (8) the evaluation for the first time of the CTR [53] dataset; (9) the identification of the main drawbacks and limitations of current methods; and finally, (10) a detailed statistical significance analysis of the results.

Selection of methods

The criteria for the selection of the sentence similarity methods evaluated herein is as follows: (a) all the methods that have been evaluated in BIOSSES and MedSTS datasets; (b) a selection of methods that have not been evaluated in the biomedical domain yet; (c) a collection of new variants or adaptations of methods previously proposed for the general or biomedical domain, which are evaluated for the first time in this work, such as the WBSM-cosJ&C [30, 42, 46, 67], WBSM-coswJ&C [30, 42, 46, 67], WBSM-Cai [30, 42, 68], UBSM-cosJ&C [30, 42, 46, 67], UBSM-coswJ&C [30, 42, 46, 67], and UBSM-Cai [30, 42, 68] methods detailed in Tables 3 and 4; and (d) a new string-based method based on Li et al. [56] introduced in this work. For a more detailed description of the selection criteria of the methods, we refer the reader to our registered report protocol [40].

Tables 2 and 3 detail the configuration of the string-based measures and ontology-based measures that are evaluated here, respectively. Both WBSM and UBSM methods are evaluated in combination with the following word and concept similarity measures: Rada et al. [69], Jiang&Conrath [70], and three state-of-the-art unexplored measures, called cosJ&C [42, 46], coswJ&C [42, 46], and Cai et al. [42, 68]. The word similarity measure which reports the best results is used to evaluate the COM method [30, 69]. Table 4 details the sentence similarity methods based on the evaluation of pre-trained character, word, and Sentence Embedding (SE) models that are evaluated in this work. Finally, Table 5 details the pre-trained language models that are evaluated in our experiments.

Table 2. Detailed setup for the string-based sentence similarity measures which are evaluated in this work. All the string-based measures follow the implementation of Sogancioglu et al. [30], who use the Simmetrics library [71]. The LiBlock method proposed herein is an adaptation from Li et al. [56] combined with a string-based measure, as detailed in the previous section.

ID	Method	Detailed setup of each method
M1	Qgram [72]	$sim(a,b) = \frac{2 \times \lvert q{-}grams(a) \cup q{-}grams(b) \rvert}{\lvert q{-}grams(a) \rvert + \lvert q{-}grams(b) \rvert}$, being *a* and *b* sets of q words, and with q = 3.
M2	Jaccard [73, 74]	$sim(a,b) = \frac{\lvert a \cup b \rvert}{\lvert a \cap b \rvert}$, being *a* and *b* sets of words of the first and second sentence respectively.
M3	Block distance [57]	$sim(s_1, s_2) = 1 - \frac{\sum_{i=1}^{\lvert D \rvert} \lvert fr(w_i, s_1) - fr(w_i, s_2) \rvert}{\sum_{i=1}^{\lvert D \rvert} fr(w_i, s_1 + s_2)}$, as detailed in equation Eq 2.
M4	LiBlock (this work)	LiBlock method (see Eq (1) annotated with CUI concepts and using cTAKES combined with the Block Distance [57] method using its best pre-processing configuration.
M5	Levenshtein distance [75]	Measures the minimal cost number of insertions, deletions and replacements needed for transforming the first into the second sentence. Insert, delete and substitution cost set to 1.
M6	Overlap coefficient [76]	$sim(a,b) = \frac{\lvert a \cap b \rvert}{\lvert Min(\lvert a \rvert, \lvert b \rvert) \rvert}$, being *a* and *b* sets of words of the first and second sentence respectively.

https://doi.org/10.1371/journal.pone.0276539.t002

Table 3. Detailed setup for the ontology-based sentence similarity measures evaluated in this work. The evaluation of the methods using Rada [69], coswJ&C [46], and Cai [68] word similarity measures use a reformulation of the original path-based measures based on the new Ancestors-based Shortest-Path Length (AncSPL) algorithm [42].

ID	Sentence similarity method	Detailed setup of each method
M7	WBSM-Rada [30, 42, 69]	WBSM [30] combined with Rada [69] measure using the AncSPL algorithm [42]
M8	WBSM-J&C [30, 67, 70]	WBSM [30] combined with J&C [70]measure and Sanchez et al. [67] IC model
M9	WBSM-cosJ&C [30, 42, 46] (this work)	WBSM [30] with cosJ&C [46] measure and Sanchez et al. [67] IC model using the AncSPL algorithm [42]
M10	WBSM-coswJ&C [30, 42, 46, 67] (this work)	WBSM [30] with coswJ&C [46] measure and Sanchez et al. [67] IC model using the AncSPL algorithm [42]
M11	WBSM-Cai [30, 42, 68]	WBSM [30] combined with Cai et al. [68] measure and Cai et al. [68] IC model using the AncSPL algorithm [42]
M12	UBSM-Rada [30, 42, 69]	UBSM [30] with Rada et al. [69] measure using the AncSPL algorithm [42]
M13	UBSM-J&C [30, 67, 70]	UBSM [30] combined with J&C [70] measure and Sanchez et al. [67] IC model
M14	UBSM-cosJ&C [30, 47, 67] (this work)	UBSM [30] with cosJ&C [46] measure and Sanchez et al. [67] IC model using the AncSPL algorithm [42]
M15	UBSM-coswJ&C [30, 42, 46, 67] (this work)	UBSM [30] with coswJ&C [46] measure and Sanchez et al. [67] IC model using the AncSPL algorithm [42]
M16	UBSM-Cai [30, 42, 68]	UBSM [30] combined with Cai et al. [68] measure and Cai et al. [68] IC model using the AncSPL algorithm [42]
M17	COM [30, 69]	$\lambda \cdot$ WBSM-Rada + $(1 - \lambda) \cdot$ UBSM-Rada with $\lambda = 0.5$

https://doi.org/10.1371/journal.pone.0276539.t003

Table 4. Detailed setup for the sentence similarity methods based on pre-trained character, word (WE) and sentence (SE) embedding models evaluated herein.

ID	Sentence similarity method	Detailed setup of each method
M18	Flair [77]	Contextual string embeddings trained on PubMed
M19	Pyysalo et al. [78]	Skip-gram trained on PubMed + PMC
M20	BioConceptVec [79]	Skip-gram WE model trained on PubMed using word2vec program
M21	BioConceptVec [79]	CBOW WE model trained on PubMed using word2vec program
M22	Newman-Griffis et al. [80]	Skip-gram WE model trained on PubMed using word2vec program
M23	Newman-Griffis et al. [80]	CBOW WE model trained on PubMed using word2vec program
M24	Newman-Griffis et al. [80]	GloVe WE model trained on PubMed
M25	BioConceptVec$_{GloVe}$ [79]	GloVe We model trained on PubMed
M26	BioWordVec$_{int}$ [81]	FastText [58] WE model trained on PubMed + MeSH
M27	BioWordVec$_{ext}$ [81]	FastText [58] trained on PubMed + MeSH
M28	BioNLP2016$_{win2}$ [82]	FastText [58] WE model based on skip-gram and trained on PubMed with training setup detailed in [82]
M29	BioNLP2016$_{win30}$ [82]	FastText [58] WE model based on skip-gram and trained on PubMed with training setup detailed in [82]
M30	BioConceptVec$_{fastText}$ [79]	FastText [58] WE model trained on PubMed
M31	Universal Sentence Encoder (USE) [83]	USE SE pre-trained model of Cer et al. [83]
M32	BioSentVec [33]	sent2vec [84] SE model trained on PubMed + MIMIC-III
M33	FastText-Skipgram-BioC (this work)	FastText [58] WE model based on Skip-gram and trained on PMC-BioC corpus (05,09,2019) with the following setup: vector dim. = 200, learning rate = 0.05, sampling thres. = 1e-4, and negative examples = 10

https://doi.org/10.1371/journal.pone.0276539.t004

Table 5. Detailed setup for the sentence similarity methods based on pre-trained language models evaluated in this work.

ID	Sentence similarity method	Detailed setup of each method
M34	BioBERT Base 1.0 [85] (+ PubMed)	BERT [86] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M35	BioBERT Base 1.0 [85] (+ PMC)	BERT [86] trained on English Wikipedia + BooksCorpus + PMC full-text articles
M36	BioBERT Base 1.0 [85] (+ PubMed + PMC)	BERT [86] trained on English Wikipedia + BooksCorpus + PubMed abstracts + PMC full-text articles
M37	BioBERT Base 1.1 [85] (+ PubMed)	BERT [86] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M38	BioBERT Large 1.1 [85] (+ PubMed)	BERT [86] trained on English Wikipedia + BooksCorpus + PubMed abstracts
M39	NCBI-BlueBERT Base [32] PubMed	BERT [86] trained on PubMed abstracts
M40	NCBI-BlueBERT Large [32] PubMed	BERT [86] trained on PubMed abstracts
M41	NCBI-BlueBERT Base [32] PubMed + MIMIC-III	BERT [86] trained on PubMed abstracts + MIMIC-III
M42	NCBI-BlueBERT Large [32] PubMed + MIMIC-III	BERT [86] trained on PubMed abstracts + MIMIC-III
M43	SciBERT [87]	BERT [86] trained on PubMed abstracts
M44	ClinicalBERT [88]	BERT [86] trained on PubMed abstracts
M45	PubMedBERT [89] (abstracts)	BERT [86] trained on PubMed abstracts
M46	PubMedBERT [89] (abstracts + full text)	BERT [86] trained on PubMed abstracts + full text
M47	ouBioBERT-Base [90] (Uncased)	BERT [86] trained on PubMed abstracts
M48	BioClinicalBERT [29]	BERT [86] trained on MIMIC-III
M49	BioDischargesummaryBERT [29]	BERT [86] trained on MIMIC-III summaries
M50	DischargesummaryBERT [29]	BERT [86] trained on MIMIC-III summaries

https://doi.org/10.1371/journal.pone.0276539.t005

Pre-processing methods evaluated in this study

The pre-processing stage aims to ensure a fair comparison of the methods that are evaluated in a single end-to-end pipeline. To achieve this goal, the pre-processing stage normalizes and decomposes the sentences into a series of components that evaluate the same sequence of words applied to all the methods simultaneously. The selection criteria of the pre-processing components have been conditioned by the following constraints: (a) the pre-processing methods and tools used by state-of-the-art methods; and (b) the availability of resources and software tools. Fig 2 details all the possible combinations of pre-processing configurations that are

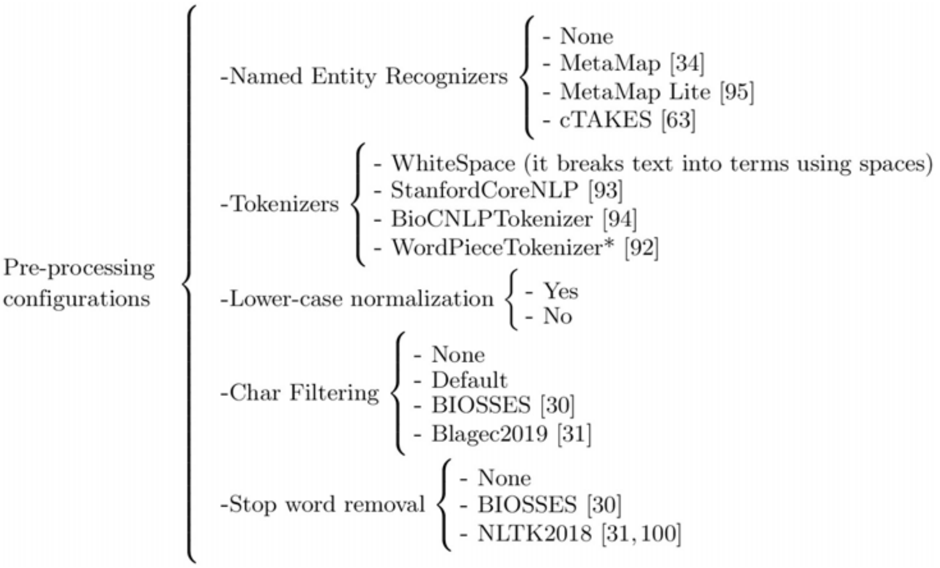


Fig 2. Detail of the pre-processing configurations that are evaluated in this work. (*) WordPieceTokenizer [91] is used only for BERT-based methods [30, 31, 34, 62, 91–94, 99].

https://doi.org/10.1371/journal.pone.0276539.g002

evaluated in this work. String, word and sentence embedding, and ontology-based methods, are evaluated using all the available configurations except the WordPieceTokenizer [91], which is specific to BERT-based methods. Thus, BERT-based methods are evaluated using different char filtering, lower casing normalization, and stop word removal configurations. We use the Pearson and Spearman correlation metrics together with their harmonic score values to determine the impact of the different pre-processing configurations on the performance of the methods evaluated herein. However, we set the best overall performing pre-processing configuration using the harmonic average scores, as well as answering the remaining research questions.

Most methods receive as input the sequences of words making up the sentences to be compared. The process of splitting sentences into words can be carried out by tokenizers, such as the well-known general domain Stanford CoreNLP tokenizer [92], which is used by Blagec et al. [31], or the biomedical domain BioCNLPTokenizer [93]. On the other hand, the use of lexicons instead of tokenizers for sentence splitting would be inefficient because of the vast general and biomedical vocabulary. Besides, it would not be possible to provide a fair comparison of the methods because the pre-trained language models have no identical vocabularies.

The tokenized words that constitute the sentence, named tokens, are usually pre-processed by removing special characters and lower-casing, and removing the stop words. To analyze all the possible combinations of token pre-processing configurations from the literature, we replicate for each method those pre-processing configurations used by other authors, such as Blagec et al. [31] and Sogancioglu et al. [30], and we also evaluate all the pre-processing configurations that have not been evaluated yet. We also study the impact of the pre-processing configurations by not removing special characters and stop words from the tokens, nor normalizing them using lower-casing.

Ontology-based sentence similarity methods estimate the similarity of a sentence by exploiting the 'is-a' relationships between the concepts in an ontology. Therefore, the evaluation of any ontology-based method receives a set of concept-annotated pairs of sentences. The aim of the biomedical NER tools is to recognize automatically biomedical entities in pieces of raw text, such as diseases or drugs. We evaluate the impact of the three more broadly-used biomedical NER tools on the performance of the sentence similarity methods, as follows: (a) MetaMap [34], (b) cTAKES [62], and (c) MetaMap Lite [94]. MetaMap tool [34] is used by UBSM and COM methods [30] for recognizing Unified Medical Language System (UMLS) [95] concepts in the sentences, which is the standard compendium of biomedical vocabularies. Likewise, we use the default configuration of MetaMap restricted to the UMLS sources of SNOMED-CT and MeSH implemented by HESML V1R5 [42, 96], which is defined by the following features: (i) the use of all available semantic types; (ii) the MedPost Part-of-speech tagger [97]; and (iii) the MetaMap Word-Sense Disambiguation (WSD) module. We also evaluate cTAKES [63] because it has shown to be a robust and reliable tool to recognize biomedical entities [98]. Given the high computational cost of MetaMap in evaluating large text corpora, Demner-Fushman et al. [94] introduced a lighter MetaMap version, called Metamap Lite, which provides a real-time implementation of the basic MetaMap annotation capabilities without a large degradation of its performance.

Due to the large number of possible combinations of each pre-processing dimension, such as Named Entity Recognizers, tokenizers or char filtering methods, we have evaluated the pre-processing combinations of each dimension by defining a fixed pre-processing configuration for the rest of the dimensions, except for the string-based methods, whose performance is high enough to not cause a significant variation in the running time of the experiments.

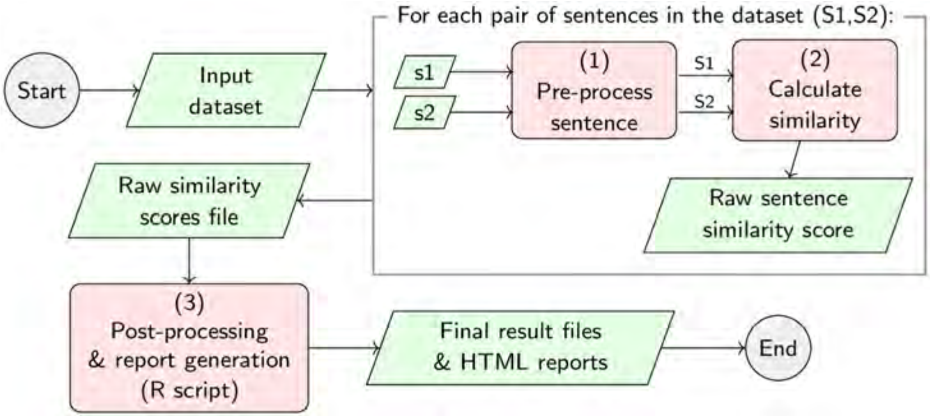

For each pair of sentences in the dataset (S1,S2):
Start
Input dataset
s1
s2
(1) Pre-process sentence
S1
S2
(2) Calculate similarity
Raw sentence similarity score
Raw similarity scores file
(3) Post-processing & report generation (R script)
Final result files & HTML reports
End

Fig 3. Detailed workflow implemented by our experiments for pre-processing the input sentences, calculating the raw similarity scores, and post-processing the results obtained in the evaluation of the biomedical datasets. This workflow generates a collection of raw and processed data files.

https://doi.org/10.1371/journal.pone.0276539.g003

Detailed workflow of our experiments

Fig 3 shows the workflow for running the experiments implemented in this work. Given an input dataset, such as BIOSSES [30], MedSTS [52], or CTR [53], the first step is to pre-process all the sentences, as shown in Fig 4. For each sentence pair (s_1, s_2) in the dataset, the pre-processing stage is divided into four stages as follows: (1.a) named entity recognition of UMLS [95] concepts, using different state-of-the-art NER tools, such as MetaMap [34] or cTAKES [62]; (1.b) tokenization of the sentences, using well-known tokenizers, such as the Stanford CoreNLP tokenizer [92], BioCNLPTokenizer [93], or WordPieceTokenizer [91] for BERT-based methods; (1.c) lower-case normalization; (1.d) character filtering, which allows the removal of punctuation marks or special characters; and finally, (1.e) the removal of stop-words, following different approximations evaluated by other authors like Blagec et al. [31] or Soganciogl u et al. [30]. Once each dataset is pre-processed in step 1 detailed in Fig 3), the aim of step 2 is to calculate the similarity score between each pair of sentences in the dataset to produce a raw output file containing all raw similarity scores, one score per sentence pair. Finally, a R-language script is used in step 3 to process the raw similarity files and produce the final human-readable tables reporting the Pearson and Spearman correlation values shown in Table 8, as well as the statistical significance of the results and any other supplementary data

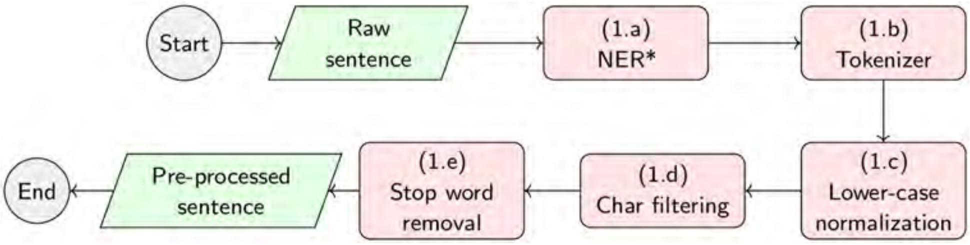


Fig 4. Detailed sentence pre-processing workflow that are implemented in our experiments. The pre-processing stage takes an input sentence and produces a pre-processed sentence as output. (*) The named entity recognizer are only evaluated in ontology-based methods.

https://doi.org/10.1371/journal.pone.0276539.g004

table required by our study on the impact of the pre-processing and NER tools reported in appendices A and B respectively.

Finally, we also evaluate all the pre-processing combinations for each family of methods to study the impact of the pre-processing methods on the performance of the sentence similarity methods, with the only exception of the BERT-based methods. The pre-processing configurations of the BERT-based methods are only evaluated in combination with the WordPiece Tokenizer [91] because it is required by the current BERT implementations.

Evaluation metrics

The evaluation metrics used to compare the performance of the methods analyzed are the following: (1) the Pearson correlation, denoted by r in Eq (6); (2) the Spearman rank correlation, denoted by ρ in equation (Eq (7)); (3) and the harmonic score, denoted by h in equation (Eq (8)). The Pearson correlation evaluates the linear correlation between two random samples, whilst the Spearman rank correlation is rank-invariant and evaluates the monotonic relationship between two random samples, and the harmonic score allows comparing sentence similarity methods by using a single weighted score based on their performance in Pearson and Spearman correlation.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{6}$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \qquad di = (x_i - y_i) \tag{7}$$

$$h = \frac{2r\rho}{r + \rho} \tag{8}$$

Statistical significance of the results

We use the well-known t-Student test to carry-out a statistical significance analysis of the results of the evaluation of the methods in the tree biomedical datasets shown in Table 1. In order to compare the overall performance of the semantic measures that is evaluated in our experiments, we use the harmonic score average in all datasets. The statistical significance of the results is evaluated using the p-values resulting from the t-student test for the mean difference between the harmonic score values reported by each pair of semantic measures in all datasets. The p-values are computed using a one-sided t-student distribution on two paired random sample vectors made up of the harmonic (h) score values obtained in the evaluation of the three aforementioned datasets. Our null hypothesis, denoted by H_0, is that the difference in the average performance between each pair of compared sentence similarity methods is 0, whilst the alternative hypothesis, denoted by H_1, is that their average performance is different. For a 5% level of significance, it means that if the p-value is greater than or equal to 0.05, we must accept the null hypothesis. Otherwise, we can reject H_0 with an error probability of less than the p-value. In this latter case, we say that a first sentence similarity method obtains a statistically significantly higher value than the second one or that the former one significantly outperforms the second one.

Uniform size datasets for our statistical significance analysis. The scarcity of datasets for this problem and the notable size difference among datasets varying from 100 to 1,068 sentence pairs makes it impossible to study the statistical significance of the results with an

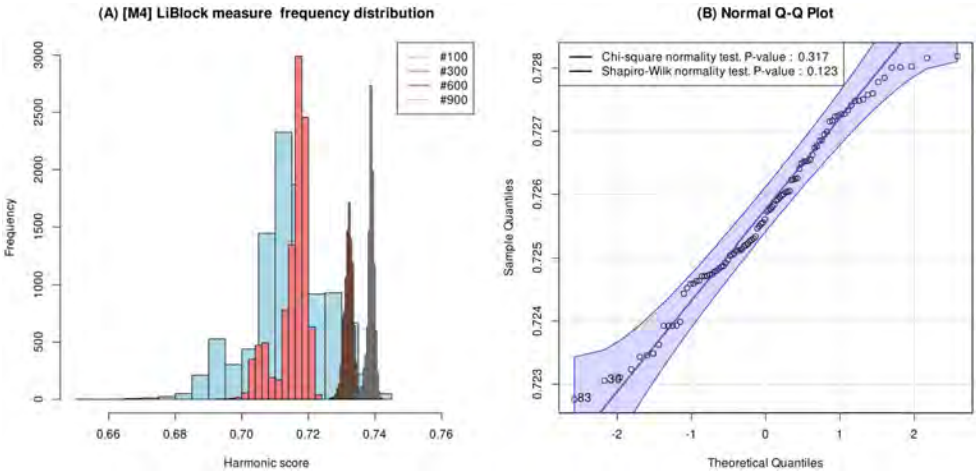


Fig 5. Figure (a) below shows the histogram plots for the harmonic score obtained by the Li-Block measure [M4] in evaluating the sentence similarity of 10,000 different equal-size subsets of sentence pairs extracted from the MedSTS dataset. Each histogram plot represents the frequency distribution of 10,000 samples of the harmonic score with subsets of sentence pairs with sizes: 100, 300, 600, and 900. Figure (b) shows the Q-Q plot normality test for the harmonic score obtained for a random subset with size 100, along with the p-values reported by the Shapiro-Wilk and Chi-square normality tests.

https://doi.org/10.1371/journal.pone.0276539.g005

adequate sample size and to carry out a fair and unbiased comparison of the results. It is a known fact [48] that the statistical distribution of the Pearson and Spearman correlation values reported by any semantic similarity measure can significantly vary regarding the dataset size, which means that the statistical distribution of the harmonic score obtained for small subsets of a large dataset as MedSTS is not the same as that obtained for the whole dataset, as shown in Fig 5a. Fig 5a shows the histogram plots for the harmonic score obtained by the Li-Block measure [M4] in evaluating the sentence similarity of 10,000 different equal-size subsets of sentence pairs extracted from the MedSTS dataset for four different subset sizes: 100, 300, 600, and 900 sentence pairs. Fig 5a shows that the harmonic score follows a different normal distribution for each subset size, whose normality is subsequently confirmed by the Q-Q plot shown in Fig 5b and the Shilford-Wilk (p-value = 0.123) and Chi-square (p-value = 0.317) tests for the sample of harmonic score values for subsets with size 100. Thus, the correlation values derived from MedSTS (1,068 pairs) could bias our results and violate the underlying hypothesis of the t-Student test that requires that the data has the same normal distribution. This potential risk of degradation of our significance analysis increases by the fact that we only have 3 datasets of different sizes (100; 1,068; 170). For this reason, we have divided the MedSTS dataset into 10 parts, considered as independent datasets, to perform the study of the statistical significance of the results. Thus, we have artificially obtained 12 datasets of 100 to 200 pairs of sentences to build the vectors of harmonic score values used in the computation of the p-values. This set of datasets allows us to obtain the p-values to compare the statistical significance between the different measures, but does not affect the processed results from Table 8. All the necessary resources for obtaining both Table 8 and the table containing all the p-values reported in S1 Appendix are publicly available in the reproducibility dataset and the companion Lab Protocol article currently in preparation, as detailed in Table 6.

Bonferroni correction for multiple hypothesis testing. Our discussion introduces some conclusions derived from the evaluation of multiple pairwise hypothesis tests to elucidate the statistical significance of the outperformance of one baseline similarity measure among a family of methods. In these latter cases, we define a set of null hypotheses $\{H_1, \ldots, H_m\}$ setting that

Table 6. Supplementary material and reproducibility resources of this work.

Material	Description
Reproducibility dataset [43]	All raw input and output data files, pre-trained model files, and a long-term reproducibility image based on Docker, which is publicly available on the Spanish Dataverse Network (https://doi.org/10.21950/EPNXTR)
Reproducibility protocol [44]	Raw step-by-step instructions to download the required resources and reproduce the experiments evaluated in this work
Lab Protocol article [44] (under preparation)	Data and methods article introducing a very detailed description of our experiments, datasets, and reproducibility protocol to allow the independent replication of our experiments and results
HESML-STS software library (integrated into HESML V2R1)	Release of the new HESML-STS library. This library is based on the previous HESML V1R5 version [41, 42] published in Github (https://github.com/jjlastra/HESML) and the Spanish Dataverse Network [43] under a CC By-NC-SA-4.0 license.
HESML V2R1 software release [105]	Release of the new HESML V2R1 version. This new release is based on the previous HESML V1R5 version [42], including the new HESML-STS software package that has been developed for this study, after managing all the licensing restrictions of the NER tools.
HESML-STS software paper [104] (under preparation)	Software article introducing our sentence similarity library, called HESML-STS, together with some benchmarks under preparation.

https://doi.org/10.1371/journal.pone.0276539.t006

the pairwise mean difference between the harmonic score obtained by one baseline measure and the remaining methods in the same family is 0. To reduce the family-wise type I error (false positives) derived from our multiple comparisons [100], we define a Bonferroni correction to evaluate the statistical significance of multiple hypothesis tests involved in those conclusions in which one baseline sentence similarity measure is compared with a family of methods. For each single conclusion comparing one baseline measure with other methods, we define a corrected null-hypothesis rejection threshold α_c defined as $\alpha_c = \alpha/m$, where α is equal to 0.05 for a 5% level of significance and m is the number of pairwise comparisons (uncorrected p-values). Thus, the null-hypothesis is only rejected if the p-values are lower than α_c when multiple pairwise hypotheses are tested.

Statistical performance analysis of the best methods

In order to answer the RQ5 research question, we study how well the sentence similarity methods estimate the degree of semantic similarity between two sentences by analyzing the deviation of their estimated values with respect to the human similarity scores. We want to analyze why the methods are doing well or badly on specific sentence pairs to provide an explanation for this behaviour, as well as identifying the main drawbacks and limitations of the current state-of-the-art methods. To carry out this performance analysis, we analyze the statistics of the similarity error function E_{sim} of the methods defined in Eq 9. We only use some sentences extracted from the BIOSSES dataset for this analysis because this dataset has no licensing restrictions on its use, which allows us to reproduce their sentences here, unlike MedSTS. We could have also used CTR because it has no licensing restrictions; however, CTR has not been previously used in this sentence similarity task.

$$\begin{aligned} E_{sim} &: L_{\Sigma} \times L_{\Sigma} \rightarrow [0,1] \subset \mathbb{R} \\ E_{sim}(s_1, s_2) &= sim(s_1, s_2) - humanSim(s_1, s_2) \end{aligned} \tag{9}$$

Our methodology to conduct the performance analysis is detailed below:

1. Selection of the best-performing method from each family of methods.

2. Estimation of the Probability Density Function (PDF) of the E_{sim} function for the evaluation of the selected best-performing methods in each dataset by calling the "*density*" function provided by the R statistical package.

3. Selection of the sentences based on their similarity error in the BIOSSES dataset:

   3.1 the sentences with the lowest and highest absolute similarity error $|E_{sim}|$ for each method are extracted.

   3.2 each sentence selected in the step above is pre-processed using the best pre-processing configuration for each method.

   3.3 the resulting pre-processed sentences and the statistical information of the similarity scores are analyzed in the *Discussion* section.

Software implementation

We have developed a new sentence measures library for the biomedical domain called HESML-STS, which is based on HESML V1R5 [41, 42], as detailed in Table 6. All our experiments are generated by running the *HESMLSTSclient* and *HESMLSTSImpactpre-processingclient* programs, which generates a raw output file in comma-separated file format (*.csv) for each dataset detailed in Table 1. The raw output files contain the raw similarity values returned by each sentence similarity method in the evaluation of the degree of similarity between sentences. The final results for the Pearson and Spearman correlation, and the harmonic values detailed in Table 8 are automatically generated by running an R-language script file on the collection of raw similarity files, which also generates all the tables reported in appendices A and B provided as supplementary material. All tables are written both in LaTeX and comma-separated file format (*.csv) formats. For a more detailed description of the protocol for running our experiments, we refer the reader to the protocol [44] detailed in S3 Appendix.

We implemented a parser for loading pre-trained embedding models based on FastText [58] and other word embedding models [78–82], which are efficiently evaluated as sentence similarity measures in HESML by implementing the averaging Simple Word EMbedding (SWEM) approach introduced by Shen et al. [101]. However, the software replication required to evaluate sentence embedding and BERT-based language models is extremely complex and out of the scope of this work. For this reason, these models are evaluated using the original software artifacts used to generate the aforementioned pre-trained models. Thus, we implemented a collection of Python wrappers for evaluating the available models by using the provided software artifacts as follows: (1) Sent2vec-based models [33] are evaluated using the Sent2vec library [84]; (2) Flair models [77] are evaluated using the flairNLP framework [77]; and USE models [83] are evaluated using the open source platform TensorFlow [102]. All BERT-based pre-trained models are evaluated using the open source bert-as-a-service library [103].

Reproducing our benchmarks

For the sake of reproducibility, we introduce a detailed reproducibility protocol on protocols.io [44] that is based on a reproducibility dataset [43] containing all the software and data necessary to allow the exact replication of all our experiments and results. Our reproducibility protocol is mainly based on a Docker-based image that includes a pre-installation of all the necessary software and the Java source code and binary files of our benchmark program, which is provided as supplementary material in our reproducibility dataset [43] and

DockerHub (https://hub.docker.com/repository/docker/alicialara/hesml_v2r1). Our source code files are tagged on Github with a permanent tag named "Release_HESML_V2R1" (https://github.com/jjlastra/HESML/releases/tag/Release_HESML_V2R1).

In addition, we plan to submit a Lab Protocol article under preparation [44] (https://collections.plos.org/collection/lab-protocols), which will provide a detailed description of the publicly available reproducibility dataset [43] and a very detailed reproducibility protocol [44] to allow the exact replication of all our methods, experiments, and results. We also plan to submit another article [104], currently in preparation, to introduce the new HESML-STS software library integrated into the latest HESML V2R1 version [105], together with a set of reproducible benchmarks on semantic measures libraries for biomedical sentence similarity. However, our reproducibility dataset allows the full and exact replication of all our experiments by completing the licensing requirements of the UMLS databases and the aforementioned NER tools for the National Library of Medicine (NLM) of the United States (https://www.nlm.nih.gov/databases/umls.html#license_request).

Table 6 details all the reproducibility resources provided as supplementary material with this work. Our benchmarks are implemented using Java 8, Python 3 and R programming languages, and thus, they can be reproduced in any Java-compliant or Docker-compliant platforms, such as Windows, MacOS, or any Linux-based system.

Results obtained

Table 7 shows the selected pre-processing configuration of each method for obtaining their best-performing results, whilst Table 8 shows the results obtained in the evaluation of all methods in the three biomedical datasets evaluated herein by using their best pre-processing configurations. Table 9 shows the comparison of results for the highest (best) and lowest (worst) average harmonic score values for the best-performing method of each family shown in blue in Table 8, which are defined by the method obtaining the highest average harmonic score. Furthermore, Table 10 shows the results obtained in our study on the impact of NER tools on the performance of the sentence similarity methods in the evaluation of the MedSTS dataset [52]. Table 11 shows the harmonic and average harmonic scores obtained in the evaluation of the three biomedical datasets, as well as the resulting p-values comparing the NER tools for each ontology-based method. Table 12 shows the results obtained in the evaluation of the LiBlock method in the three biomedical datasets by using its best pre-processing configuration, and annotating the sentences with all the NER tools combinations. In addition, the aforementioned table details the resulting p-values comparing the best-performing LiBlock-NER combination with the other NER tools. Tables 13–16 show the raw input sentence pairs and their corresponding pre-processed versions in which the best-performing methods obtain the lowest and highest similarity error (E_{sim}) in the BIOSSES dataset [30]. Table 17 details the statistical information for the best-performing methods of each family in the evaluation of the three biomedical datasets evaluated in this study. Finally, Fig 6 shows the Probability Density Function (PDF) of the similarity error obtained by the best-performing methods of each family in the evaluation of the BIOSSES, MedSTS, and CTR datasets respectively.

S1 Appendix shows the p-values resulting from comparing all the methods using their best pre-processing configuration as detailed in Table 8, which allows us to study the statistical significance of the results, as detailed in the Discussion section. In addition, appendix B shows the experimental results regarding the impact of pre-processing configurations in all the methods evaluated here; the best configuration has been used to determine the final scores for each method. Finally, S3 Appendix details the protocol for reproducing all the experiments evaluated in this paper, and is also published on protocols.io [44].

Table 7. Best-performing pre-processing configurations used to evaluate the methods compared in this work as reported in Table 8, derived from our cross-evaluation of each method with the pre-processing configurations shown in Fig 2 (see S2 Appendix). (*) COM (M17) uses the best configuration of the WBSM-Rada (M7) and UBSM-Rada (M12) methods for computing the similarity scores.

ID	Sentence similarity method	NER	Tokenizer	Lower-case	Char filtering	Stop words removal
M1	Qgram	None	WhiteSpace	yes	BIOSSES	NLTK2018
M2	Jaccard	None	WhiteSpace	yes	BIOSSES	NLTK2018
M3	Block distance	None	WhiteSpace	yes	BIOSSES	NLTK2018
M4	LiBlock (this work)	cTAKES	CoreNLP	yes	Default	NLTK2018
M5	Levenshtein distance	None	WhiteSpace	no	None	BIOSSES
M6	Overlap coefficient	None	CoreNLP	yes	Default	NLTK2018
M7	WBSM-Rada	Exact matching	CoreNLP	yes	BIOSSES	NLTK2018
M8	WBSM-J&C	Exact matching	CoreNLP	yes	BIOSSES	None
M9	WBSM-cosJ&C (this work)	Exact matching	CoreNLP	yes	BIOSSES	None
M10	WBSM-coswJ&C (this work)	Exact matching	CoreNLP	yes	BIOSSES	NLTK2018
M11	WBSM-Cai	Exact matching	CoreNLP	yes	BIOSSES	None
M12	UBSM-Rada	cTAKES	CoreNLP	yes	BIOSSES	NLTK2018
M13	UBSM-J&C	MetamapLite	CoreNLP	yes	BIOSSES	NLTK2018
M14	UBSM-cosJ&C (this work)	MetamapLite	CoreNLP	yes	BIOSSES	NLTK2018
M15	UBSM-coswJ&C (this work)	cTAKES	CoreNLP	yes	BIOSSES	NLTK2018
M16	UBSM-Cai	MetamapLite	CoreNLP	yes	BIOSSES	NLTK2018
M17	COM (*)	-	-	-	-	
M18	Flair	None	WhiteSpace	no	BIOSSES	None
M19	Pyysalo et al.	None	CoreNLP	yes	Default	BIOSSES
M20	BioConceptVec$_{word2vec\_sg}$	None	CoreNLP	yes	Default	BIOSSES
M21	BioConceptVec$_{word2vec\_cbow}$	None	CoreNLP	yes	Default	BIOSSES
M22	Newman-Griffis$_{word2vec\_sgns}$	None	CoreNLP	yes	Default	NLTK2018
M23	Newman-Griffis$_{word2vec\_cbow}$	None	CoreNLP	yes	Default	NLTK2018
M24	Newman-Griffis$_{glove}$	None	CoreNLP	yes	Default	NLTK2018
M25	BioConceptVec$_{glove}$	None	CoreNLP	yes	Default	BIOSSES
M26	BioWordVec$_{int}$	None	CoreNLP	yes	BIOSSES	None
M27	BioWordVec$_{ext}$	None	CoreNLP	yes	BIOSSES	None
M28	BioNLP2016$_{win2}$	None	CoreNLP	no	Default	NLTK2018
M29	BioNLP2016$_{win30}$	None	CoreNLP	no	Default	NLTK2018
M30	BioConceptVec$_{fastText}$	None	CoreNLP	yes	Default	BIOSSES
M31	USE	None	CoreNLP	no	Default	None
M32	BioSentVec (PubMed+MIMIC-III)	None	CoreNLP	yes	BIOSSES	BIOSSES
M33	FastText-SkGr-BioC (this work)	None	CoreNLP	yes	Default	None
M34	BioBERT Base 1.0 (+ PubMed)	None	WordPiece	yes	BIOSSES	None
M35	BioBERT Base 1.0 (+ PMC)	None	WordPiece	yes	BIOSSES	None
M36	BioBERT Base 1.0 (PubMed+PMC)	None	WordPiece	yes	BIOSSES	None
M37	BioBERT Base 1.1 (+ PubMed)	None	WordPiece	no	Blagec2019	NLTK2018
M38	BioBERT Large 1.1 (+ PubMed)	None	WordPiece	no	Blagec2019	NLTK2018
M39	NCBI-BlueBERT Base PubMed	None	WordPiece	yes	Blagec2019	None
M40	NCBI-BlueBERT Large PubMed	None	WordPiece	yes	BIOSSES	None
M41	NCBI-BlueBERT Base PubMed + MIMIC-III	None	WordPiece	yes	BIOSSES	BIOSSES
M42	NCBI-BlueBERT Large PubMed + MIMIC-III	None	WordPiece	yes	BIOSSES	None
M43	SciBERT	None	WordPiece	yes	BIOSSES	NLTK2018
M44	ClinicalBERT	None	WordPiece	no	Blagec2019	BIOSSES
M45	PubMedBERT (abstracts)	None	WordPiece	yes	Default	NLTK2018
M46	PubMedBERT (abstracts+full text)	None	WordPiece	yes	Default	NLTK2018

(*Continued*)

Table 7. (Continued)

ID	Sentence similarity method	NER	Tokenizer	Lower-case	Char filtering	Stop words removal
M47	ouBioBERT-Base, Uncased	None	WordPiece	yes	Default	None
M48	BioClinicalBERT	None	WordPiece	yes	Blagec2019	BIOSSES
M49	BioDischargesummaryBERT	None	WordPiece	no	Blagec2019	NLTK2018
M50	DischargesummaryBERT	None	WordPiece	no	Blagec2019	NLTK2018

https://doi.org/10.1371/journal.pone.0276539.t007

Discussion

Comparison of string-based methods

LiBlock (M4) obtains the highest average harmonic score among the family of string-based methods and significantly outperforms all of them. This conclusion can be drawn by looking at the average column in Table 8 for this group of methods and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that LiBlock obtains p-values lower than $\alpha_c = 0.05/5$ (0,01) when it is compared with all the string-based methods, such as Block Distance (p-value = 0.000), Jaccard (p-value = 0.000), QGram (p-value = 0.000), Overlap Coefficient (p-value = 0.000), and Levenshtein (p-value = 0.000).

LiBlock (M4) obtains the highest Pearson correlation value in the BIOSSES and MedSTS datasets among the family of string-based methods, whilst Block Distance (M3) obtains the highest Pearson correlation in the CTR dataset. This conclusion can be drawn by looking at the results for the first group of methods detailed in Table 8.

LiBlock (M4) obtains the highest Spearman correlation value in all datasets among the family of string-based methods. This conclusion can be drawn by looking at the results for the first group of methods detailed in Table 8.

LiBlock (M4) obtains the highest harmonic score in all datasets among the family of string-based methods. This conclusion can be drawn by looking at the results for the first group of methods detailed in Table 8.

Comparison of Ontology-based methods

COM (M17) obtains the highest average harmonic score among the family of ontology-based methods and significantly outperforms all of them, with the sole exception of WBSM-Rada (M7). This conclusion can be drawn by looking at the average column in Table 8 for the second group of methods and checking the p-values shown in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that COM obtains a p-value lower than $\alpha_c = 0.05/10$ (0,005) when it is compared with all ontology-based methods, with the only exception of WBSM-Rada (M7) (p-value = 0.088).

COM (M17) obtains the highest Pearson correlation value in the BIOSSES and CTR datasets among the family of ontology-based methods, whilst the WBSM-Rada (M7) methods obtain the highest Pearson correlation value in the MedSTS dataset. This conclusion can be drawn by looking at the second group of methods in 8.

COM (M17) obtains the highest Spearman correlation values in the BIOSSES dataset among the family of ontology-based methods, whilst WBSM-Rada (M7) and UBSM-Rada (M12) do so in the MedSTS and CTR datasets, respectively. This conclusion can be drawn by looking at the second group of methods in 8.

COM (M17) obtains the highest harmonic score in the BIOSSES and CTR datasets among the family of ontology-based methods, whilst WBSM-Rada (M7) does so in the MedSTS dataset. This conclusion can be drawn by looking at the second group of methods detailed in Table 8.

Table 8. Pearson (r), Spearman (ρ), harmonic (*h*), and harmonic average (AVG) scores obtained by each sentence similarity method evaluated herein in the three biomedical sentence similarity benchmarks arranged by families. All reported values were obtained using the best pre-processing configurations detailed in Table 7. The results in bold show the best scores whilst results in blue show the best average harmonic score for each family.

ID	Sentence similarity methods	BIOSSES [30]			MedSTS$_{full}$ [52]			CTR [53]			AVG
		r	ρ	h	r	ρ	h	r	ρ	h	h
M1	Qgram	0.752	0.773	0.763	0.701	0.674	0.687	0.763	0.766	0.764	0.738
M2	Jaccard	0.782	0.815	0.798	0.706	0.680	0.693	0.759	0.797	0.777	0.756
M3	Block distance	0.798	0.818	0.808	0.731	0.683	0.706	0.797	0.801	0.799	0.771
M4	LiBlock (this work)	0.820	**0.828**	**0.824**	0.769	**0.710**	0.739	0.793	0.808	0.800	**0.788**
M5	Levenshtein distance	0.529	0.536	0.533	0.610	0.634	0.622	0.498	0.536	0.516	0.557
M6	Overlap coefficient	0.782	0.795	0.788	0.696	0.564	0.623	0.781	0.793	0.787	0.733
M7	WBSM-Rada	0.772	0.791	0.782	**0.774**	0.709	**0.740**	0.785	0.765	0.775	0.766
M8	WBSM-J&C	0.483	0.549	0.514	0.647	0.614	0.630	0.536	0.516	0.526	0.557
M9	WBSM-cosJ&C (this work)	0.483	0.549	0.514	0.647	0.614	0.630	0.536	0.516	0.526	0.557
M10	WBSM-coswJ&C (this work)	0.571	0.566	0.568	0.705	0.651	0.677	0.637	0.590	0.613	0.619
M11	WBSM-Cai	0.458	0.542	0.497	0.629	0.601	0.615	0.492	0.459	0.475	0.529
M12	UBSM-Rada	0.792	0.809	0.800	0.763	0.700	0.730	0.776	0.794	0.785	0.772
M13	UBSM-J&C	0.529	0.573	0.550	0.683	0.621	0.650	0.620	0.585	0.602	0.601
M14	UBSM-cosJ&C (this work)	0.615	0.648	0.631	0.699	0.638	0.667	0.709	0.646	0.676	0.658
M15	UBSM-coswJ&C (this work)	0.730	0.769	0.749	0.697	0.625	0.659	0.713	0.673	0.693	0.700
M16	UBSM-Cai	0.545	0.579	0.562	0.686	0.628	0.656	0.642	0.576	0.607	0.608
M17	COM	0.793	0.809	0.801	0.773	0.708	0.739	0.789	0.783	0.786	0.776
M18	Flair	0.628	0.625	0.626	-0.014	-0.035	-0.020	0.652	0.719	0.684	0.430
M19	Pyysalo et al. [78]	0.713	0.706	0.709	0.754	0.641	0.693	0.744	0.803	0.773	0.725
M20	BioConceptVec$_{word2vec\_sg}$	0.742	0.743	0.742	0.751	0.652	0.698	0.738	0.800	0.768	0.736
M21	BioConceptVec$_{word2vec\_cbow}$	0.670	0.655	0.662	0.746	0.650	0.695	0.659	0.714	0.685	0.681
M22	Newman-Griffis$_{word2vec\_sgns}$	0.771	0.763	0.767	0.764	0.641	0.697	**0.799**	**0.835**	**0.817**	0.760
M23	Newman-Griffis$_{word2vec\_cbow}$	0.675	0.686	0.681	0.746	0.647	0.693	0.697	0.768	0.731	0.701
M24	Newman-Griffis$_{glove}$	0.671	0.678	0.674	0.740	0.643	0.688	0.732	0.729	0.731	0.698
M25	BioConceptVec$_{glove}$	0.547	0.585	0.565	0.720	0.648	0.682	0.624	0.694	0.657	0.635
M26	BioWordVec$_{int}$	**0.831**	0.806	0.818	0.766	0.686	0.724	0.757	0.735	0.746	0.763
M27	BioWordVec$_{ext}$	0.752	0.725	0.738	0.756	0.673	0.712	0.736	0.729	0.732	0.727
M28	BioNLP2016$_{win2}$	0.697	0.693	0.695	0.699	0.594	0.642	0.691	0.759	0.724	0.687
M29	BioNLP2016$_{win30}$	0.745	0.751	0.748	0.714	0.609	0.657	0.742	0.810	0.774	0.727
M30	BioConceptVec$_{fastText}$	0.091	0.262	0.135	0.416	0.456	0.435	0.178	0.264	0.212	0.261
M31	USE	0.666	0.669	0.668	0.679	0.606	0.640	0.663	0.684	0.674	0.660
M32	BioSentVec	0.797	0.767	0.782	0.763	0.638	0.695	0.791	0.821	0.806	0.761
M33	FastText-SkGr-BioC (this work)	0.814	0.777	0.795	0.758	0.660	0.706	0.761	0.760	0.760	0.754
M34	BioBERT Base 1.0 (+ PubMed)	0.569	0.567	0.568	0.662	0.576	0.616	0.616	0.642	0.629	0.604
M35	BioBERT Base 1.0 (+ PMC)	0.664	0.663	0.664	0.674	0.581	0.624	0.601	0.647	0.623	0.637
M36	BioBERT Base 1.0$_{(PubMed+\ PMC)}$	0.616	0.609	0.612	0.647	0.561	0.601	0.638	0.663	0.650	0.621
M37	BioBERT Base 1.1 (+ PubMed)	0.668	0.647	0.657	0.712	0.616	0.661	0.643	0.663	0.653	0.657
M38	BioBERT Large 1.1 (+ PubMed)	0.557	0.546	0.551	0.695	0.622	0.657	0.579	0.650	0.612	0.607
M39	NCBI-BlueBERT Base PubMed	0.682	0.668	0.675	0.679	0.565	0.617	0.668	0.719	0.693	0.662
M40	NCBI-BlueBERT Large PubMed	0.688	0.712	0.700	0.636	0.588	0.611	0.609	0.674	0.640	0.650
M41	NCBI-BlueBERT Base PubMed + MIMIC-III	0.537	0.536	0.536	0.733	0.624	0.674	0.548	0.553	0.550	0.587
M42	NCBI-BlueBERT Large PubMed + MIMIC-III	0.560	0.578	0.569	0.675	0.628	0.651	0.487	0.504	0.496	0.572
M43	SciBERT	0.653	0.616	0.634	0.727	0.643	0.683	0.604	0.682	0.641	0.652
M44	ClinicalBERT	0.415	0.483	0.447	0.652	0.566	0.606	0.470	0.500	0.485	0.512
M45	PubMedBERT (abstracts)	0.502	0.524	0.513	0.626	0.531	0.575	0.479	0.645	0.550	0.546

(*Continued*)

Table 8. (Continued)

ID	Sentence similarity methods	BIOSSES [30]			MedSTS$_{full}$ [52]			CTR [53]			AVG
		r	ρ	h	r	ρ	h	r	ρ	h	h
M46	PubMedBERT (abstracts+full text)	0.659	0.651	0.655	0.712	0.590	0.645	0.596	0.675	0.633	0.644
M47	ouBioBERT-Base, Uncased	0.687	0.729	0.707	0.707	0.583	0.639	0.670	0.694	0.682	0.676
M48	BioClinicalBERT	0.416	0.447	0.431	0.646	0.562	0.601	0.472	0.478	0.475	0.502
M49	BioDischargesummaryBERT	0.376	0.397	0.387	0.637	0.565	0.599	0.385	0.465	0.421	0.469
M50	DischargesummaryBERT	0.395	0.465	0.427	0.655	0.567	0.608	0.376	0.407	0.391	0.475

https://doi.org/10.1371/journal.pone.0276539.t008

Table 9. Comparison of results for the "best" and the "worst" pre-processing configurations for the best-performing methods of each family in Table 8. The last column shows the t-Student p-values comparing the best and worst configurations.

ID	Methods	Pre-processing configuration	BIOSSES			MedSTS$_{full}$			CTR			AVG	p-val
			r	ρ	h	r	ρ	h	r	ρ	h	h	
M4	LiBlock (worst)	TOK-Whitespace LC-No SW-NLTK2018 CF-None	0.779	0.793	0.786	0.736	0.676	0.704	0.765	0.717	0.741	0.744	
													0.000
M4	LiBlock (best)	TOK-CoreNLP LC-Yes SW-NLTK2018 CF-Default	0.820	0.828	0.824	0.769	0.710	0.739	0.793	0.808	0.800	0.788	
M17	COM (worst)	—WBSM-Rada - UBSM-Rada (worst): TOK-Whitespace LC-Yes SW-None CF-None	0.610	0.635	0.622	0.681	0.648	0.664	0.656	0.662	0.659	0.648	
													0.000
M17	COM (best)	—WBSM-Rada - UBSM-Rada (best): TOK-CoreNLP LC-Yes SW-NLTK2018 CF-BIOSSES	0.793	0.809	0.801	0.773	0.708	0.739	0.789	0.783	0.786	0.776	
M26	BioWordVec$_{int}$ (worst)	TOK-Whitespace LC-No SW-None CF-None Pooling-Sum	0.436	0.497	0.465	0.532	0.619	0.572	0.529	0.674	0.593	0.543	
													0.000
M26	BioWordVec$_{int}$ (best)	TOK-CoreNLP LC-Yes SW-None CF-BIOSSES Pooling-Min	0.831	0.809	0.820	0.764	0.682	0.721	0.761	0.736	0.748	0.763	
M47	OuBioBert (worst)	TOK- WordPiece LC-Yes SW-BIOSSES CF-Default	0.608	0.627	0.617	0.730	0.622	0.672	0.669	0.696	0.682	0.657	
													0.000
M47	OuBioBert (best)	TOK-WordPiece LC-Yes SW-None CF-Default	0.687	0.729	0.707	0.707	0.583	0.639	0.670	0.694	0.682	0.676	

https://doi.org/10.1371/journal.pone.0276539.t009

Table 10. Pearson (r), Spearman (ρ) and harmonic (*h*) values obtained in our experiments from the evaluation of ontology similarity methods detailed below in the MedSTS$_{full}$ [52] dataset for each NER tool.

ID	Methods	MetaMap			MetaMap Lite			cTAKES		
		r	ρ	h	r	ρ	h	r	ρ	h
M12	UBSM-Rada	0.711	0.653	0.681	0.753	0.689	0.720	**0.764**	**0.7**	**0.73**
M13	UBSM-J&C	0.576	0.547	0.561	**0.683**	**0.621**	**0.65**	0.634	0.549	0.588
M14	UBSM-cosJ&C	0.637	0.575	0.605	**0.699**	**0.638**	**0.667**	0.659	0.581	0.617
M15	UBSM-coswJ&C	0.675	0.608	0.64	**0.722**	**0.659**	**0.689**	0.697	0.625	0.659
M16	UBSM-Cai	0.606	0.555	0.58	**0.686**	**0.628**	**0.656**	0.635	0.552	0.591
M17	COM	0.758	0.692	0.724	0.770	0.706	0.737	**0.773**	**0.708**	**0.739**

https://doi.org/10.1371/journal.pone.0276539.t010

Table 11. Harmonic score obtained by each combination of a sentence similarity method with a NER tool in the evaluation of the three sentence similarity datasets. The p-values shown in this table are obtained by using the method for building uniform size datasets detailed above. The last column shows the p-values corresponding to the t-Student test comparing the performance of each combination with the best pair in each group.

ID	Methods	NER tool	BIOSSES	MedSTS	CTR	Avg	p-value
			h	*h*	*h*	*h*	
M12	UBSM-Rada	cTAKES	0.800	0.730	0.785	0.772	—
		MetamapLite	0.744	0.72	0.785	0.751	0.011
		Metamap	0.742	0.680	0.723	0.715	0.000
M13	UBSM-J&C	MetamapLite	0.55	0.65	0.602	0.601	—
		cTAKES	0.595	0.588	0.552	0.578	0.000
		Metamap	0.316	0.561	0.234	0.37	0.000
M14	UBSM-cosJ&C	MetamapLite	0.631	0.667	0.674	0.657	—
		cTAKES	0.681	0.617	0.626	0.641	0.002
		Metamap	0.537	0.605	0.434	0.525	0.000
M15	UBSM-coswJ&C	cTAKES	0.749	0.659	0.693	0.700	—
		MetamapLite	0.678	0.689	0.732	0.700	0.018
		Metamap	0.656	0.64	0.551	0.616	0.005
M16	UBSM-Cai	MetamapLite	0.562	0.656	0.607	0.608	—
		cTAKES	0.616	0.591	0.571	0.593	0.001
		Metamap	0.419	0.58	0.318	0.439	0.000
M17	COM	cTAKES	**0.801**	**0.739**	0.786	**0.776**	—
		MetamapLite	0.788	0.737	**0.789**	0.772	0.052
		Metamap	0.792	0.724	0.768	0.761	0.004

https://doi.org/10.1371/journal.pone.0276539.t011

Table 12. Pearson (r) and Spearman (ρ) correlation values, harmonic score (*h*), and harmonic average (AVG) score obtained by the LiBlock method in combination with each NER tool using the best pre-processing configuration detailed in Table 7. In addition, the last column (p-val) shows the p-values for the comparison of the LiBlock method with cTAKES and the remaining NER combinations.

ID	Sentence similarity methods	BIOSSES [30]			MedSTS$_{full}$ [52]			CTR [53]			AVG	p-val
		r	ρ	h	r	ρ	h	r	ρ	h	h	
M4	LiBlock-cTAKES	**0.820**	**0.828**	**0.824**	0.769	**0.710**	**0.739**	0.793	**0.808**	0.800	**0.788**	-
M4	LiBlock-noNER	0.814	0.823	0.819	**0.770**	0.709	0.738	**0.795**	0.805	0.800	0.786	0.14
M4	LiBlock-MetamapLite	0.799	0.819	0.809	0.763	0.705	0.733	0.794	**0.808**	**0.801**	0.781	0.015
M4	LiBlock-Metamap	0.807	0.826	0.816	0.753	0.690	0.720	0.792	0.807	0.799	0.779	0.003

https://doi.org/10.1371/journal.pone.0276539.t012

Table 13. Raw and pre-processed sentence pairs obtaining the lowest and highest similarity error E_{sim} together with their corresponding Normalized human similarity score (Human) and normalized similarity value (Method) estimated by the LiBlock (M4) method for the raw and pre-processed sentence pairs with the lowest (L) and highest (H) similarity error E_{sim}.

E_{sim}	Input sentence	Pre-processed sentence analyzed by the method	Human	Method
L	s1: "Centrosomes increase both in size and in microtubule-nucleating capacity just before mitotic entry."	s1: "C0242608 increase size C0026046 nucleating capacity mitotic entry"	0.0	0.0
	s2: "Functional studies showed that, when introduced into cell lines, miR-146a was found to promote cell proliferation in cervical cancer cells, which suggests that miR-146a works as an oncogenic miRNA in these cancers."	s2: "functional studies showed introduced C0007634 lines mir 146a found promote C0007634 C0334094 C4048328 C0007634 suggests mir 146a works oncogenic mirna C0006826"		
H	s1: "Consequently miRNAs have been demonstrated to act either as oncogenes (e.g., miR-155, miR-17–5p and miR-21) or tumor suppressors (e.g., miR-34, miR-15a, miR-16–1 and let-7)"	s1: "consequently mirnas demonstrated C0427611 either oncogenes e g mir 155 mir 17 5p mir 21 C0027651 suppressors e g mir 34 mir 15a mir 16 1 let 7"	0.7	0.0
	s2: "Given the extensive involvement of miRNA in physiology, dysregulation of miRNA expression can be associated with cancer pathobiology including oncogenesis], proliferation, epithelial-mesenchymal transition, metastasis, aberrations in metabolism, and angiogenesis, among others"	s2: "given extensive involvement mirna physiology dysregulation mirna C0185117 associated C0006826 pathobiology including oncogenesis C0334094 epithelial mesenchymal transition metastasis aberrations C0025519 angiogenesis among others"		

https://doi.org/10.1371/journal.pone.0276539.t013

Table 14. Raw and pre-processed sentence pairs obtaining the lowest and highest similarity error E_{sim} together with their corresponding Normalized human similarity score (Human) and normalized similarity value (Method) estimated by the COM (M17) method for the raw and pre-processed sentence pairs with the lowest (L) and highest (H) similarity error E_{sim}. We show the raw and pre-processed sentence pairs evaluated by the WBSM and UBSM similarity methods that make up the COM method. The UBSM method use the cTAKES NER tool.

E_{sim}	Input sentence	Pre-processed sentence analyzed by the method	Human	Method
Low	s1: "The in vivo data is still preliminary and other potential roadblocks such as drug resistance have not been examined."	s1, WBSM-Rada: "vivo data still preliminary potential roadblocks drug resistance examined" s1, UBSM-Rada: "vivo data still preliminary potential roadblocks C0013227 resistance examined"	0.0	0.0
	s2: "The GEM model used in this study retains wild-type Tp53, suggesting that the tumors successfully treated with bortezomib and fasudil might not be as aggressive as those in most NSCLC patients"	s2, WBSM-Rada: "gem model used study retains wild type tp53 suggesting tumors successfully treated bortezomib fasudil might aggressive nsclc patients" s2, UBSM-Rada: "gem model used study retains wild type tp53 suggesting C0027651 successfully treated C1176309 fasudil might aggressive C0007131 patients"		
High	s1: "The oncogenic activity of mutant Kras appears dependent on functional Craf, but not on Braf"	s1, WBSM-Rada: "oncogenic activity mutant kras appears dependent functional craf braf" s1, UBSM-Rada: "oncogenic C0026606 mutant kras appears dependent functional craf braf"	0.75	0.0
	s2: "Notably, c-Raf has recently been found essential for development of K-Ras-driven NSCLCs"	s2, WBSM-Rada: "notably c raf recently found essential development k ras driven nsclcs" s2, UBSM-Rada: "notably c raf recently found essential development k C0525678 driven nsclcs"		

https://doi.org/10.1371/journal.pone.0276539.t014

Table 15. Raw and pre-processed sentence pairs obtaining the lowest and highest similarity error E_{sim} together with their corresponding Normalized human similarity score (Human) and normalized similarity value (Method) estimated by the BioWordVec$_{int}$ (M26) method for the raw and pre-processed sentence pairs with the lowest (L) and highest (H) similarity error E_{sim}.

E_{sim}	Input sentence	Pre-processed sentence analyzed by the method	Human	Method
Low	s1: "The up-regulation of miR-146a was also detected in cervical cancer tissues."	s1: "the up regulation of mir 146a was also detected in cervical cancer tissues"	1.0	0.986
	s2: "The expression of miR-146a has been found to be up-regulated in cervical cancer."	s2: "the expression of mir 146a has been found to be up regulated in cervical cancer"		
High	s1: "This oxidative branch activity is elevated in comparison to many cancer cell lines, where the oxidative branch is typically reduced and accounts for <20% of the carbon flow through PPP."	s1: "this oxidative branch activity is elevated in comparison to many cancer cell lines where the oxidative branch is typically reduced and accounts for < 20% of the carbon flow through ppp"	0.0	0.912
	s2: "The Downward laboratory went all the way from identifying GATA2 as a novel synthetic lethal gene to validating it using Kras-driven GEM models."	s2: "the downward laboratory went all the way from identifying gata2 as a novel synthetic lethal gene to validating it using kras driven gem models"		

https://doi.org/10.1371/journal.pone.0276539.t015

Table 16. Raw and pre-processed sentence pairs obtaining the lowest and highest similarity error E_{sim} together with their corresponding Normalized human similarity score (Human) and normalized similarity value (Method) estimated by the OuBioBert (M47) method for the raw and pre-processed sentence pairs with the lowest (L) and highest (H) similarity error E_{sim}.

E_{sim}	Input sentence	Pre-processed sentence analyzed by the method	Human	Method
Low	s1: "Expression of an activated form of Ras proteins can induce senescence in some primary fibroblasts."	s1: "expression activated form ras proteins induce senescence primary fibroblasts"	0.9	0.908
	s2: "The senescent state has been observed to be inducible in certain cultured cells in response to high level expression of genes activated such as the ras oncogene."	s2: "senescent state observed inducible certain cultured cells response high level expression genes activated ras oncogene"		
High	s1: "The in vivo data is still preliminary and other potential roadblocks such as drug resistance have not been examined."	s1: "vivo data still preliminary potential road bl ocks drug resistance examined"	0.0	0.773
	s2: "The GEM model used in this study retains wild-type Tp53, suggesting that the tumors successfully treated with bortezomib and fasudil might not be as aggressive as those in most NSCLC patients"	s2: "gem model used study retains wild type tp53 suggesting tumors successfully treated bortezomib fas udi l might aggressive nsclc patients"		

https://doi.org/10.1371/journal.pone.0276539.t016

Comparison of embedding methods

BioWordVec$_{\text{int}}$ *(M26) obtains the highest average harmonic score in all datasets among the family of embedding methods detailed in Table 4, but does not significantly outperforms all of them.* This conclusion can be drawn by looking at the third group of methods in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the BioWordVec$_{int}$ (M26) obtains p-values higher than $\alpha_c = 0.05/15$ (0,003) when it is compared with the FastText-SkGr-BioC (M33) and Flair (M18) embedding methods.

BioWordVec$_{\text{int}}$ *(M26) obtains the highest Pearson correlation value in the BIOSSES and MedSTS datasets among the family of embedding methods, whilst the*

Table 17. Comparison of the mean, minimum and maximum similarity scores of the Normalized Human similarity scores (Human) and the estimated values returned by the best-performing methods of each family in the evaluation of the three biomedical datasets.

BIOSSES dataset				
ID	Method	Mean similarity	Minimum similarity	Maximum similarity
-	Human	0.549	0	1
M4	LiBlock (this work)	0.194	0	0.506
M17	COM [30]	0.22	0	0.596
M26	BioWordVec$_{int}$ [81]	0.933	0.858	0.987
M47	OuBioBert [90]	0.808	0.582	0.936
MedSTS dataset				
ID	Method	Mean similarity	Minimum similarity	Maximum similarity
-	Human	0.632	0	1
M4	LiBlock (this work)	0.611	0	1
M17	COM [30]	0.631	0	1
M26	BioWordVec$_{int}$ [81]	0.957	0.832	1
M47	OuBioBert [90]	0.885	0.437	0.997
CTR dataset				
ID	Method	Mean similarity	Minimum similarity	Maximum similarity
-	Human	0.254	0	1
M4	LiBlock (this work)	0.103	0	0.743
M17	COM [30]	0.118	0	0.793
M26	BioWordVec$_{int}$ [81]	0.898	0.752	0.992
M47	OuBioBert [90]	0.724	0.472	0.98

https://doi.org/10.1371/journal.pone.0276539.t017

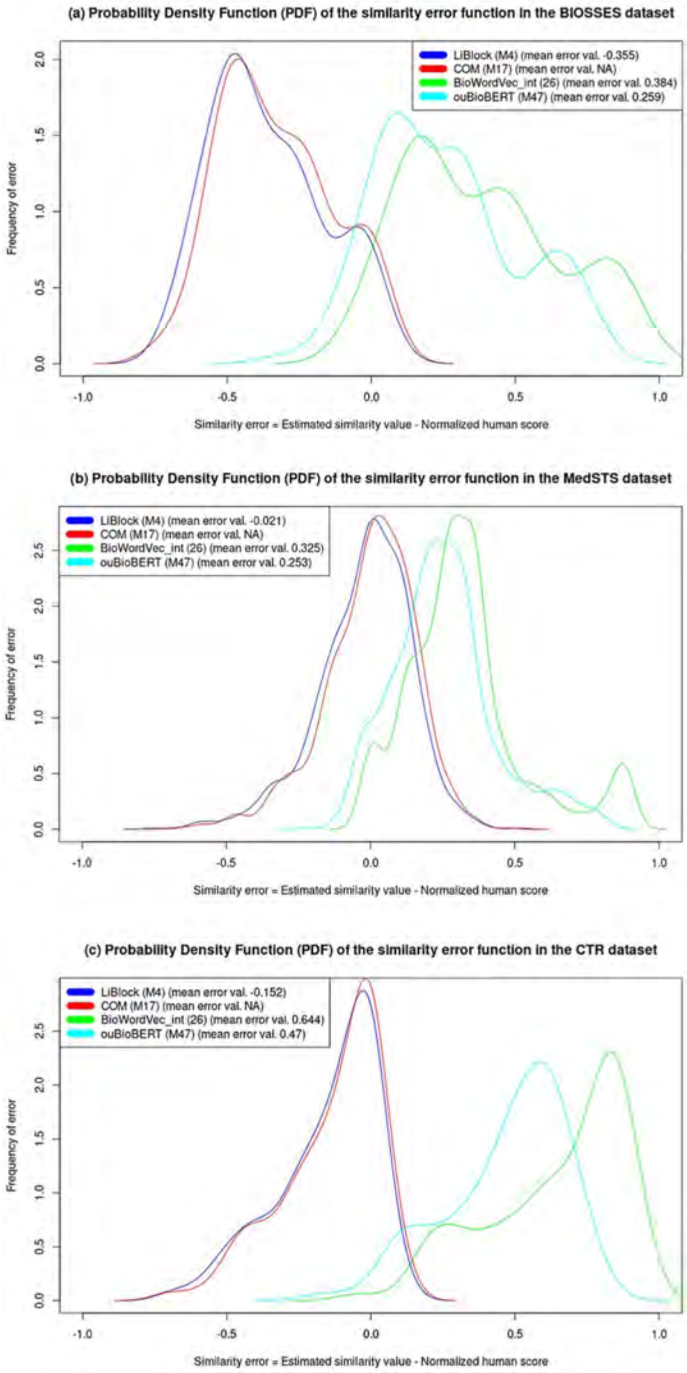


Fig 6. Probability Density Function (PDF) and mean value of the similarity error (E_{sim}) obtained by the best-performing methods in the evaluation of each dataset as follows: (a) BIOSSES, (b) MedSTS, and (c) CTR.

https://doi.org/10.1371/journal.pone.0276539.g006

Newman-Griffis$_{\text{word2vec\_sgns}}$ *(M22) model does so in the CTR dataset*. This conclusion can be drawn by looking at the results for third group of methods detailed in Table 8.

BioWordVec$_{\text{int}}$ *(M26) obtains the highest Spearman correlation in the BIOSSES and MedSTS datasets among the family of embedding methods, whilst the Newman-Griffis*$_{\text{word2vec\_sgns}}$ *(M22)* *model does so in the CTR dataset.* This conclusion can be drawn by looking at the results for the third group of measures detailed in Table 8.

BioWordVec$_{\text{int}}$ *(M26) obtains the highest harmonic score in the BIOSSES and MedSTS datasets among the family of embedding methods, whilst the Newman-Griffis*$_{\text{word2vec\_sgns}}$ *(M22) model does so in the CTR dataset.* This conclusion can be drawn by looking at the results for the third group of measures detailed in Table 8.

Comparison of BERT-based methods

OuBioBERT (M47) obtains the highest average harmonic score among the family of BERT-based methods. However, it does not significantly outperform all of them. This conclusion can be drawn by looking at the last group of methods in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that ouBioBERT obtains p-values higher than $\alpha_c = 0.05/16$ (0,003) when it is compared with many BERT-based methods, such as BioBERT Large 1.1 (p-value = 0.224) and PubMedBERT (abstracts+full text) (p-value = 0.101) among others.

NCBI-BlueBERT Large PubMed (M40) obtains the highest Pearson correlation value in the BIOSSES dataset among the family of BERT-based methods, whilst the NCBI-BlueBERT Base PubMed + MIMIC-III (M41) and the ouBioBERT (M47) models do so in the MedSTS and the CTR datasets, respectively. This conclusion can be drawn by looking at the last group of measures detailed in Table 8.

ouBioBERT (M47) obtains the highest Spearman correlation value in the BIOSSES dataset among the family of BERT-based methods, whilst SciBERT (M43) and NCBI-BlueBERT Base PubMed (M39) do so in the MedSTS and CTR datasets, respectively. These conclusions can be drawn by looking at the last group of measures detailed in Table 8.

ouBioBERT (M47) obtains the highest harmonic score in the BIOSSES dataset among the family of BERT-based methods, whilst SciBERT (M43) and NCBI-BlueBERT Base PubMed (M39) do so in the MedSTS and CTR datasets, respectively. This conclusion can be drawn by looking at the last group of measures detailed in Table 8.

Comparison of all methods

LiBlock (M4) obtains the highest average harmonic score for all the methods evaluated herein, and significantly outperforms all the methods based on language models. However, there is no a statistically significant difference in performance with the embedding methods Flair (M18) and BioWordVec$_{\text{int}}$ *(M26), and the ontology-based methods COM (M17) and WBSM-Rada (M7).* This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the LiBlock obtains p-values higher than $\alpha_c = 0.05/16$ (0,003) when it is compared with the embedding-based methods Flair (M18) and BioWordVec$_{int}$ (M26). In addition, the LiBlock method obtain p-values higher than $\alpha_c = 0.05/11$ (0,004) when it is compared with the ontology-based methods COM (M17) and WBSM-Rada (M7).

BioWordVec$_{\text{int}}$ *(M26) obtains the highest Pearson correlation values in the BIOSSES dataset among all methods evaluated here, whilst WBSM-Rada (M7) and Newman-Griffis*$_{\text{word2vec\_sgns}}$ *(M22) do so in the MedSTS and CTR datasets, respectively.* This conclusion can be drawn by looking at the bold values detailed in Table 8.

LiBlock (M4) obtains the highest Spearman correlation value in the BIOSSES and MedSTS datasets among all methods evaluated here, whilst Newman-Griffis$_{\text{word2vec\_sgns}}$ *(M22) do so in* *the CTR dataset.* These conclusions can be drawn by looking at the bold values detailed in Table 8.

LiBlock (M4) obtains the highest harmonic score in the BIOSSES dataset among all methods evaluated here, whilst WBSM-Rada (M7) and Newman-Griffis$_{\text{word2vec\_sgns}}$ *(M22) do so in the MedSTS and CTR datasets, respectively.* This conclusion can be drawn by looking at the bold values detailed in Table 8.

COM (M17) obtains the second highest average harmonic score among all methods evaluated here, and it is able to outperform significantly all the methods based on language models. However, it does not significantly outperforms all the embedding, ontology or string-based methods. This conclusion can be drawn by looking at the bold values detailed in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that COM obtains p-values lower than $\alpha_c = 0.05/17$ (0,002) when it is compared with all the methods based on language models. On the other hand, the COM method obtains p-values higher than $\alpha_c = 0.05/6$ (0,008), $\alpha_c = 0.05/11$ (0,004) and $\alpha_c = 0.05/16$ (0,003) respectively, when it is compared with string, ontology and embedding-based methods.

Non ML-based methods versus ML-based ones

The string-based method LiBlock (M4) obtain a higher average harmonic score than all the embedding-based methods in all datasets. Moreover, it significantly outperforms all methods based on embedding models, with the only exceptions of Flair (M18) and BioWordVec$_{\text{int}}$ *(M26)* This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that LiBlock obtains p-values lower than $\alpha_c = 0.05/16$ (0,003) when it is compared with all the embedding-based methods except for the BioWordVec$_{int}$ (p-value 0.003) and Flair (p-value 0.027) methods.

All string-based methods obtain a higher average harmonic score than all the BERT-based methods considering all datasets, with the only exception of the Levenshtein distance (M5). However, string-based methods do not significantly outperform all BERT-based methods. This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the string-based methods Qgram (M1), Jaccard (M2), Block distance (M3), Levenshtein distance (M5) and Overlap coefficient (M6) obtain p-values higher than $\alpha_c = 0.05/17$ (0,002) when they are compared with all the BERT-based methods.

The ontology-based methods COM (M17), WBSM-Rada (M7) and UBSM-Rada (M12) obtain a higher average harmonic score than all the embedding-based methods considering all datasets. However, they do not significantly outperform all embedding-based methods. This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the ontology-based methods COM (M17), WBSM-Rada (M7) and UBSM-Rada (M12) obtain p-values higher than $\alpha_c = 0.05/16$ (0,003) when they are compared with all the embedding-based methods.

The ontology-based methods UBSM-Rada (M12), WBSM-Rada (M7), COM (M17) and UBSM-coswJ&C (M15) obtain a higher average harmonic score than all the BERT-based methods. Moreover, the ontology-based methods UBSM-Rada (M12), WBSM-Rada (M7), and COM (M17) significantly outperform all the BERT-based methods. This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the UBSM-Rada (M12), WBSM-Rada

(M7) and COM (M17) obtain p-values lower than α_c = 0.05/17 (0,002) when they are compared with all the BERT-based methods.

All embedding methods obtain a higher average harmonic score than all BERT-based methods, with the only exceptions of Flair (M18), BioConceptVec$_{\text{glove}}$ (M25), BioConceptVec$_{\text{fastText}}$ (M30) and USE (M31). This conclusion can be drawn by looking at the last column in Table 8.

BioWordVec$_{\text{int}}$ (M26) obtains a higher average harmonic score than all the BERT-based methods considering all datasets and significantly outperforms all of them, with the only exception of NCBI-BlueBERT Base PubMed + MIMIC-III (M41). This conclusion can be drawn by looking at the average column in Table 8 and checking the p-values reported in Table A.1 in S1 Appendix. Table A.1 in S1 Appendix shows that the BioWordVec$_{int}$ (M26) method obtains p-values lower than α_c = 0.05/17 (0,002) when it is compared with all the BERT-based methods, except for the NCBI-BlueBERT Base PubMed + MIMIC-III (p-value = 0.002).

Impact of the NER tools on the ontology-based methods

This section analyzes the impact of the NER tools on the performance of the sentence similarity methods, and studies the overall impact of the NER configurations. Table 10 shows the results obtained on the performance of NER tools for the sentence similarity methods evaluated in the MedSTS dataset [52], whilst Table 11 shows the harmonic and average harmonic scores, as well as the p-values which result from comparing the harmonic score of the best-performing NER tool for each ontology-based method in the three datasets with the harmonic scores obtained by the other two NER tools.

MetamapLite obtains the highest Pearson, Spearman, and harmonic scores for the MedSTS dataset in combination with UBSM-J&C (M13), UBSM-cosJ&C (M14), UBSM-coswJ&C (M15) and UBSM-Cai (M16), whilst cTAKES obtains the highest Pearson, Spearman and harmonic scores for the MedSTS dataset in combination with UBSM-Rada (M12) and COM (M17). This latter conclusion can be drawn by looking at the results shown in Table 10.

cTAKES obtains the highest average harmonic score for the three datasets in combination with UBSM-Rada (M12), UBSM-coswJ&C (M15) and COM (M17) methods, whilst MetamapLite obtains the highest average harmonic score for the three datasets in combination with UBSM-J&C (M13), UBSM-cosJ&C (M14) and UBSM-Cai (M16). This conclusion can be drawn by looking at the harmonic scores of the NER tools in Table 11.

cTAKES combined with COM (M17) obtains the best-performing results of ontology-based methods for the three datasets. This conclusion can be drawn by looking at the average harmonic scores column shown in Table 11.

cTAKES is the best-performing tool in combination with the UBSM-Rada (M12), UBSM-coswJ&C (M15), and COM (M17) methods in the three datasets, and significantly outperforms MetamapLite and Metamap or the two former methods. However, there is no a statistically significant difference regarding the Metamap tools when it is combined with the COM (M17) method. This conclusion can be drawn by looking at the average harmonic scores and p-values shown in Table 11, which are lower than α_c = 0.05/2 (0,025).

MetamapLite is the best-performing tool in combination with the UBSM-J&C (M13), UBSM-cosJ&C (M14), and UBSM-Cai (M16) methods in the three datasets, and significantly outperforms cTAKES and Metamap. This conclusion can be drawn by looking at the average harmonic scores and p-values shown in Table 11, which are lower than α_c = 0.05/2 (0,025).

The choice of the best NER tool for each method significantly impacts their performance in most cases. This conclusion follows from the conclusions above.

Answering RQ3. Our results show that the ontology-based methods obtain their best performance in the task of biomedical sentence similarity when they use either MetamapLite or cTAKES. Thus, Metamap should not be used in combination with any of the ontology-based methods evaluated here in this task. Likewise, the results and p-values reported Table 11 show that there is a significant difference in the performance of each ontology-based method according to the NER tool used in most cases. The conclusions above confirm that the selection of the NER tool significantly impacts the performance of the sentence similarity methods using it.

Impact of the NER tools on the new LiBlock measure

This section analyzes the impact of the NER tools on the new sim_{LiBk} similarity measure. Table 12 shows the results obtained by the sim_{LiBk} measure in the three biomedical datasets using its best pre-processing configuration, and annotating the sentences with all the combinations of NER tools. In addition, the aforementioned table details the p-values resulting from comparing the best-performing LiBlock-NER combination with the combinations based on the other two NER tools.

LiBlock-cTAKES obtains the highest average harmonic score for the three datasets among the LiBlock-NER combinations. However, it does not significantly outperform LiBlock with no use of a NER tool. This conclusion can be drawn by looking at the average column in Table 12 and checking the p-values in the last column. This conclusion is especially relevant because it shows that there is no statistically significant difference between using a NER tool like cTAKES or not using it, in the case of the LiBlock measure. We conjecture that this result could have two explanations: firstly, the inability of LiBlock to capture semantic relationships beyond the synonymy, and secondly, the current limitations of cTAKES in recognizing all mentions of biomedical entities.

LiBlock-cTAKES obtains the highest Pearson correlation value in the BIOSSES dataset among all LiBlock-NER combinations, whilst LiBlock with no use of a NER tool obtains the highest Pearson correlation value in the MedSTS and CTR datasets, respectively. This conclusion can be drawn by looking at the results detailed in Table 12.

LiBlock-cTAKES obtains the highest Spearman correlation value in the BIOSSES and MedSTS datasets among the LiBlock-NER combinations, whilst LiBlock-cTAKES and LiBlock-MetamapLite obtain the highest Spearman correlation value in the CTR dataset. This conclusion can be drawn by looking at the results detailed in Table 12.

LiBlock-cTAKES obtains the highest harmonic correlation value in the BIOSSES and MedSTS datasets among the LiBlock-NER combinations, whilst LiBlock-MetamapLite obtains the highest harmonic correlation value in the CTR dataset. This conclusion can be drawn by looking at the results detailed in Table 12.

Impact of the remaining pre-processing stages

This section analyzes the impact of each pre-processing step on the performance of the sentence similarity methods, except for the NER tools already analyzed in the previous section. Finally, we study the overall impact of the pre-processing configurations.

Impact of tokenization. *The family of string-based methods obtains its best-performing results either by splitting the sentence on the spaces between words or using the Stanford CoreNLP tokenizer.* This conclusion can be drawn by looking at Table 7, which summarizes the pre-processing tables detailed in S2 Appendix.

The family of ontology-based methods obtains its best-performing results in combination with the Stanford CoreNLP tokenizer. This conclusion can be drawn by looking at Table 7.

The family of methods based on embedding obtains its best-performing results in combination with the Stanford CoreNLP tokenizer, with the only exception of Flair (M18). This conclusion can be drawn by looking at Table 7.

No method based on strings, ontologies, or embedding obtains its best-performing results in combination with the BioCNLPTokenizer. This conclusion can be drawn by looking at Table 7. Thus, the BioCNLPTokenizer should not be used in combination with any method in the abovementioned families in the task of biomedical sentence similarity. On the other hand, we recall that all BERT-based methods evaluated herein can only be used in combination with the WordPiece Tokenizer [91] based on a subword segmentation algorithm, because it is required by the current BERT implementations.

All families of methods show a strong preference for a specific tokenizer, with the only exception of the string-based one. This conclusion can be drawn from previous conclusions that confirm the preference of the methods based on ontologies and embedding for the CoreNLP tokenizer, and the mandatory use of the WordPiece tokenizer by the family of BERT-based methods.

Impact of character filtering. *The family of string-based methods obtains its best-performing results by using either the BIOSSES char-filtering method or the default method which removes the punctuation marks and special symbols from the sentences, with the only exception of the Levenshtein distance method (M5), which does not remove special characters.* This conclusion can be drawn by looking at Table 7, which summarizes the pre-processing tables detailed in S2 Appendix.

All ontology-based methods obtain their best-performing results in combination with the BIOSSES char-filtering method. This conclusion can be drawn by looking at Table 7.

Most embedding methods obtain their best-performing results in combination with the default char filtering method. However, Flair (M18), BioWordVec (M26,M27), and BioSentVec (M32) do better with BIOSSES char-filtering. This conclusion can be drawn by looking at Table 7.

The BERT-based methods do not show a noticeable preference pattern for a specific char filtering method, obtaining their best-performing results with the BIOSSES, Blagec2019, or the default one. This conclusion can be drawn by looking at Table 7.

Impact of stop-words removal. *All string-based methods obtain their best-performing results in combination with the NLTK2018 stop-word list, with the only exception of the Levenshtein distance (M5).* This conclusion can be drawn by looking at Table 7, which summarizes the pre-processing tables detailed in S2 Appendix.

All ontology-based methods obtain their best-performing results in combination with the NLTK2018 stop-word list, with the only exception of WBSM-J&C (M8), WBSM-cosJ&C (M9), which do not remove stop words. This conclusion can be drawn by looking at Table 7.

The methods based on embedding do not show a noticeable preference pattern for a specific stop-word list, obtaining their best-performing results by using the stop-word list of BIOSSES, NLTK2018, or none at all. This conclusion can be drawn by looking at Table 7.

The methods based on language models do not show a noticeable preference pattern for a specific stop-word list, obtaining their best-performing results by using the stop-word list of BIOSSES, NLTK2018, or none at all. This conclusion can be drawn by looking at Table 7.

The best-performing results for the methods based on strings or ontologies show a noticeable preference for the use of the stop-words list NLTK2018. This conclusion can be drawn by looking at the Table 7.

Impact of lower-casing. *Only 10 of the 50 methods evaluated in this work obtain their best performance without converting words to lowercase at the sentence pre-processing stage.* This conclusion can be drawn by looking at Tables 7 and 8, and the pre-processing tables detailed in S2 Appendix. Moreover, these ten aforementioned methods obtain a low performance in our experiments, with the sole exception of the BioNLP2016$_{win30}$ (M29) pre-trained model, which obtains the third best Spearman correlation value in the CTR dataset. Thus, our experiments confirm that the lower-casing normalization of the sentences positively impacts the performance of the methods, and it should be considered as the default option in any biomedical sentence similarity task.

We conjecture that lower-casing improves the performance of the families of string-based and ontology-based methods because it improves the exact comparison of words. On the other hand, we also conjecture that the impact of lower-casing the sentences on the families of methods based on embedding and language models strongly depends on the pre-processing methods used in their training.

Overall impact of pre-processing. To study the overall impact of the pre-processing stage on the performance of the sentence similarity methods, we selected the configuration reporting the highest (best) and lowest (worst) average harmonic score values for each method, as shown in Table 9. These configurations were selected from a total of 1081 pre-processing configurations reported in S2 Appendix.

The best-performing methods of each family show a statistically significant difference in performance between their best and worst pre-processing configurations. This conclusion can be drawn by looking at the average (AVG) and the p-values in Table 9.

Answering RQ4. Our results and the conclusions above show that the pre-processing configurations significantly impact the performance of the sentence similarity methods, and thus, they should be specifically defined for each method. All families of methods show a strong preference for a specific tokenizer, with the sole exception of the string-based one. In addition, the BioCNLPTokenizer does not contribute to the best-performing configuration of any method evaluated here. The family of string-based methods shows a preference pattern for using either the BIOSSES or default char filtering method, whilst all ontology-based methods use the BIOSSES char filtering method, and most embedding methods use the default char filtering method. However, BERT-based methods do not show a noticeable preference pattern for a specific char filtering method. On the other hand, the families of string and ontology-based methods show a noticeable preference pattern for the use of the NLTK2018 stop-words list, whilst the families of embedding- and BERT-based methods do not show a noticeable pattern. Finally, the experiments confirm that the lower-casing normalization of the sentences positively impacts the performance of the methods, and it should be considered as the default option in any biomedical sentence similarity task.

The new state of the art

We establish the new state of the art to answer our RQ1 and RQ2 questions as follows.

The LiBlock (M4) method sets the new state of the art for the sentence similarity task in the biomedical domain (see Table 8), being the best overall performing method to tackle this task. Moreover, LiBlock significantly outperforms all the methods based on language models. However, LiBlock cannot significantly outperform the ontology-based methods COM (M17) and WBSM-Rada (M7), and the embedding-based methods Flair (M18) and BioWordVec$_{int}$ (M26) (see S1 Appendix). Thus, LiBlock is a convincing but non-definitive winner among the biomedical sentence similarity methods evaluated here.

The COM (M17) method sets the new state of the art among the family of ontology-based methods for biomedical sentence similarity, being the best-performing method in this task (see Table 8). Moreover, COM significantly outperforms all methods based on language models (see S1 Appendix).

BioWordVec$_{int}$ (M26) sets the new state of the art among the family of methods based on pre-trained embedding models, being the best-performing method in this task (see Table 8). However, BioWordVec$_{int}$ does not significantly outperforms the remaining methods in the same family (see S1 Appendix).

OuBioBERT (M47) sets the new state of the art among the family of methods based on pre-trained BERT models, being the best-performing method in this task (see Table 8). However, OuBioBERT is unable to outperform significantly all remaining methods from the same family (see S1 Appendix).

Finally, our results show that our new string-based method, called LiBlock (M4), obtains the best overall results, despite not capturing the semantic information of the sentences. This is a very notable finding because it contradicts a common belief that ontology-based methods, which integrate word and concept semantics, will outperform the non-semantic methods in this similarity task. A second and very interesting finding is that our non-semantic and non-ML LiBlock method is able to outperform significantly state-of-the-art methods based on BERT language models [86] in an unsupervised context. This latter finding is very remarkable because LiBlock is easy to implement and evaluate, very efficient (2635 sentence pairs per second with no use of a NER tool), and it requires neither large text resources nor complex algorithms for its training and evaluation, which is a very clear advantage in the biomedical sentence similarity task.

Answering RQ1 and RQ2. The string-based method LiBlock (M4) obtains the highest average harmonic score in all datasets, and significantly outperforms the remaining string-based methods, as well as all methods based on language models, and all the ontology-based methods with the only exceptions of COM (M17) and WBSM-Rada (M7). In addition, LiBlock obtains the highest Spearman correlation values in the BIOSSES and MedSTS datasets, which contain 100 and 1068 sentence pairs respectively.

Main drawbacks and limitations of current methods

This section analyzes the behaviour of the best-performing methods in each family of sentence similarity methods to answer our RQ5. The best-performing methods of each family, according to the harmonic average value reported in Table 8, are LiBlock (M4), COM (M17), BioWordVec$_{int}$ (M26), and OuBioBERT (M47).

String and ontology-based methods underestimate, on average, the human similarity value in the BIOSSES and CTR datasets, whilst their average similarity error is close to 0 in the MedSTS dataset. This conclusion can be drawn by looking at the average similarity error values and the mean error values shown in Fig 6 together with the mean values shown in Table 17. LiBlock and COM obtain mean error values of -0.021 and -0.001 in MedSTS, as shown in Fig 6b. On the other hand, both methods report a mean similarity score much lower than the mean of the Human normalized score in the BIOSSES and CTR datasets and a mean similarity score close to the Human normalized score in the MedSTS dataset, as shown in Table 17.

The methods based on embedding and language models overestimate, on average, the human similarity value in the three datasets. This conclusion can be drawn by looking at the average similarity error values and the mean error values shown in Fig 6, together with the mean similarity values shown in Table 17. The two aforementioned families of methods report a mean similarity score much higher than the mean of the Human normalized score in the three datasets, as show in Table 17.

String and ontology-based methods share a similar underestimation behavior, in contrast to the overestimation behaviour shown by the methods based on embedding and language models, which is very noticeable in the three datasets. This conclusion can be drawn by looking at the minimum and maximum similarity values columns in Table 17, and the plots of the probability error distribution function for the three datasets in Fig 6. For instance, in spite of the human similarity scores being in the range of 0 to 1 in the BIOSSES dataset, as shown in Table 17, the string and ontology-based methods report similarity scores in the range of 0 to 0.596, whilst the methods based on embedding and language models report similarity scores in the range of 0.582 to 0.987.

String and ontology-based methods tend to obtain their best results in sentences with a Human normalized score close to 0, whilst the methods based on embedding and language models obtain their best results in sentences with a Human normalized score close to 1. This conclusion can be drawn by looking at Tables 13–16. On the other hand, string and ontology-based methods tend to obtain their worst results in sentences with a Human normalized score close to 1, whilst the methods based on embedding and language models obtain their worst results in sentences with a Human normalized score close to 0.

None of the methods for semantic similarity of sentences in the biomedical domain evaluated here use an explicit syntactic analysis or syntax information to obtain the similarity value. We conjecture that syntactic analysis would improve the performance in some cases. For instance, the sentences *s*1 and *s*2 with highest E_{sim} in Table 13 show an implicit relation between the concepts "miRNA" and "oncogenesis", which should increase the final semantic similarity score of the sentences. However, none of the methods evaluated here consider and reward these semantic relationships because its recognition demands some form of syntactic analysis. On the one hand, string and ontology-based methods consider the concepts in a sentence as bags of words, whilst on the other hand the methods based on embedding and language models implicitly consider the structure of the sentences but not the relationships between the parts of the sentences that are related.

Our results show that the family of string-based methods benefits from a high frequency of overlapping words in the sentences of the current biomedical datasets, whilst such methods are not able to deal properly with sentences that are semantically different but not exhibit a word overlapping pattern. The main advantages of the string-based methods are as follows: (1) they are able to obtain high correlation values without the need of using external resources for their training or evaluation; (2) they are fast and efficient; and finally; (3) they require low computational resources. However, string-based methods are unable to capture the semantics of the words in the sentence, which prevent them from recognizing semantic relationships, such as synonymy, meronymy and morphological variants. On the other hand, the use of NER tools in combination with string-based methods is a good option to integrate at least the capability of recognizing synonyms, as shown by LiBlocK-cTAKES (M4).

Ontology-based methods strongly depend on the lexical coverage of the ontologies and the ability to recognize automatically the underlying concepts in sentences. Our results show that the ontology-based methods are able to properly estimate a similarity score when used either with a dataset with high word overlapping, or with NER and WSD tools that find all possible entities to properly calculate the similarity between sentences. The main advantages of ontology-based methods are that they are fast and require low computational resources. However, the effectiveness of the ontology-based methods depends on the lexical coverage of the ontologies and the ability of the NER and WSD tools to recognize the underlying concepts in sentences, whose coverage and performance could be limited in several application domains.

The LiBlock (M4) string-based method and the COM (M17) ontology-based method use a NER tool in the pre-processing stage to recognize the biomedical entities (UMLS CUI codes) present in the input sentences. The objective of annotating entities in the semantic similarity task is the identification and disambiguation of biomedical concepts to provide semantic information to sentences. LiBlock uses the NER tool to normalize and disambiguate the underlying concepts in a sentence, unifying different concepts with acronyms and synonyms in the same CUI code and creating an overlapping between concepts, while ontologies also make use of the similarity of concepts within ontologies.

The biomedical NER tools evaluated in this work are unable to identify and disambiguate correctly many biomedical concepts due to the use of acronyms and different morphological variations, among others. For example, the CUI concepts "KRAS gene" (C1537502), "BRAF gene" (C0812241), and "RAF1 gene" (C0812215) in the sentences $s1$ and $s2$ with highest E_{sim} obtained by the COM (M17) method in Table 14, appear as "K-ras", "Braf", "c-Raf" and "Craf". However, cTAKES is unable to recognize these later morphological variants of the same biomedical concepts. A second example is the word "act" in the sentence "Consequently miRNAs have been demonstrated to act either as oncogenes [. . .]", which is wrongly recognized as the entity "Activated clotting time measurement" (C0427611), rather than as a verb in the sentence $s1$ with highest E_{sim} in Table 13. And finally, a third example is the acronym "NSCLC", which denotes the concept "Non-Small Cell Lung Carcinoma (C0007131), which is not recognized in the plural variant "NSCLCs" in the sentence $s2$ with highest E_{sim} from Table 14.

The methods based on pre-trained embedding and language models provide a broader lexical coverage than the ontology-based methods, and do not need the use of NER or WSD tools to find intrinsic semantic relationships between the words in the sentences. However, these methods need large corpora for their training, as well as a complex training phase and more computational resources than the methods from the string-based and ontology-based families. Moreover, our experiments show that those methods tend to estimate higher similarity values than those estimated by a human being in the three datasets. In most cases, the aforementioned methods report similarity scores that tend towards 1, which indicates that the semantics obtained from the sentences is not sufficient to compute correctly a similarity score. For instance, the sentences $s1$ and $s2$ with highest E_{sim} from Tables 15 and 16 shows similarity values close to 1, where the sentences have neither word overlapping nor similar concepts, and the human similarity score is 0 in both cases. Lastly, BERT-based methods, are trained for downstream tasks, using a supervised approach, and do not perform well in an unsupervised context.

Answering RQ5. String-based methods capture neither the word semantics within the sentences nor the semantic relationships between words, such as synonymy and meronymy, and their effectiveness mainly relies on the word overlapping frequency in the sentences. However, the LiBlock method uses the NER tool to normalize and disambiguate the underlying concepts in a sentence, but unfortunately, it does not significantly outperform LiBlock with no use of a NER tool, which could have two explanations: firstly, the inability of LiBlock to capture semantic relationships beyond the synonymy; secondly, the current limitations of cTAKES in recognizing all mentions of biomedical entities. On the other hand, ontology-based methods use NER and WSD tools to recognize the underlying concepts in the sentences, which are not able correctly to identify and disambiguate these concepts in many cases. In addition, they require external resources to capture the semantic information from the sentences, which limits their lexical coverage. Thus, ontology-based methods require both high word overlapping and high recognition coverage of named entities to properly estimate the similarity between sentences. In comparison, the methods based on pre-trained embedding and language models need large corpora for training, a complex training phase, and considerable computational resources to calculate the similarity between sentences. Moreover, those methods tend to obtain high similarity scores in most cases, which may penalize them in a balanced dataset and in a real environment. Finally, BERT-based methods are trained for downstream tasks, using a supervised approach, and do not perform well in an unsupervised context.

Table 18. This table shows the running times in milliseconds (ms) and the average sentences pairs per second (sent/sec) reported by the best-performing method of each family of methods in the evaluation of the 1339 sentence pairs that comprise the three datasets. (*) The LiBlock method reports the running times in both NER and noNER versions showing that the efficiency of the method with no NER tool is much higher, despite the fact that there is no statistically significant difference in the results between both pre-processing configurations.

ID	Method	Running time (ms)	Sentence pairs / sec
M4	LiBlock-cTAKES	56605	23,66
M4	LiBlock-noNER (*)	508	2635,83
M3	Block distance	308	**4347,4**
M12	UBSM-Rada	32341	41,40
M17	COM	41558	32,22
M27	BioWordVec$_{int}$	1211	1105,69
M32	BioSentVec	54706	24,48
M47	ouBioBERT	575770	2,33
M38	BioBERT Large 1.1 (+ PubMed)	3312566	0,40

https://doi.org/10.1371/journal.pone.0276539.t018

Comparison of running times

Table 18 details the running time reported by the best-performing methods for each family, as well as the sentences per second that each method computes on average for the three datasets evaluated herein. The experiments were executed on a desktop computer with an AMD Ryzen 7 5800x CPU (16 cores) with 64 Gb RAM and a 2TB Gb SSD disk. In all cases, the running time includes the pre-processing time for each method. The string-based method Block Distance (M3) obtains the lowest running times because it does not need complex mechanisms or pre-trained models to calculate the similarity between sentences. On the other hand, the BERT-based methods obtain the worst results mainly due to their pre-processing stage, which uses the WordPiece tokenization method.

Inconsistent results in the calculation of the statistical significance matrix

Despite the artificial increase of datasets to calculate the statistical significance of the results, we have identified an inconsistent result with respect to the comparison of the p-values of the LiBlock (M4) and the WBSM-Rada (M7) and UBSM-Rada (M12) methods. Table 8 shows that the UBSM-Rada method (M12) has a higher average harmonic score compared to WBSM-Rada (M7). However, by building the artificial datasets, the value of UBSM-Rada (M12) with respect to LiBlock (M4) shows a significant difference, while WBSM-Rada (M7) with respect to LiBlock (M4) shows a non-significant difference. We conjecture that this problem could be solved by increasing the number of datasets created for this task, which would allow the sample size to be increased and obtain more consistent results.

Conclusions and future work

We have introduced the largest, detailed, and for the first time, reproducible experimental survey on biomedical sentence similarity reported in the literature. Our work also introduces a collection of self-contained and reproducible benchmarks on biomedical sentence similarity based on the same software platform, called HESML-STS, which has been especially developed for this work, being provided as part of the new HESML V2R1 version that is publicly available [105]. We provide a detailed reproducibility protocol [44] and dataset [43] to allow the exact replication of all our experiments, methods, and results. In addition, we introduce a new aggregated string-based sentence similarity method called LiBlock, together with eight variants of the ontology-based methods introduced by Sogancioglu et al. [30], and a new pre-trained word embedding model based on FastText [58] and trained on the full-text of the articles in the PMC-BioC corpus [19]. We also evaluate for the first time the CTR [53] dataset in a benchmark on biomedical sentence similarity.

The string-based LiBlock (M4) measure sets the new state of the art for the sentence similarity task in the biomedical domain and significantly outperforms all the methods of each family evaluated here, with the only exceptions of the Flair (M18), BioWordVec$_{int}$ (M26), COM (M17) and WBSM-Rada (M7) methods. However, our data analysis shows that at least with the three datasets evaluated herein, there is no statistically significant difference between the performance of the LiBlock (M4) method using the cTAKES or using no NER tool at all. Thus, using the LiBlock method without any NER tool could be a competitive and much more efficient solution for high-throughput applications.

Concerning the impact of the Named Entity Recognition (NER) tools, our results confirm that the choice of the best NER tool for each method significantly impacts their performance. MetamapLite [94] and cTAKES [62] set the best-performing configurations for the family of ontology-based methods, whilst Metamap [34] was not the best performer in any method.

Our experiments confirm that the pre-processing stage has a very significant impact on the performance of the sentence similarity methods evaluated here, and yet this aspect has neither been studied nor reported in the literature. Thus, the selection of the proper configuration for each sentence similarity method should be confirmed experimentally. However, our experiments suggest some default configurations to make these decisions, such as the use of lowercasing normalization, some specific char filtering methods, and some specific tokenizers with the sole exception of BioCNLPTokenizer. Finally, the families of string and ontology-based methods show a noticeable preference pattern for the use of the NLTK2018 stop-words list. For a detailed description of the best pre-processing configurations, we refer the readers to our discussion.

String-based methods do not capture either the semantics of the words in the sentence or the semantic relationships between words, and their effectiveness relies on the word overlapping frequency in the sentences. Ontology-based methods Named Entity Recognition (NER) and Word Sense Disambiguation (WSD) tools to recognize the underlying concepts in the sentences and require external resources to capture the semantic information from the sentences, which limits their lexical coverage. In addition, they require either high word overlapping or high recognition coverage of named entities in order to properly calculate the similarity between sentences. On the other hand, the methods based on pre-trained embedding and language models need a large corpus for training, a complex training phase, and considerable computational resources to calculate the similarity between sentences. Moreover, these methods tend to obtain high similarity scores in most cases, which may penalize them in a balanced dataset and in a real environment. Finally, BERT-based methods are trained for downstream tasks, using a supervised approach, and do not perform well in an unsupervised context.

Our experiments suggest that the current benchmarks do not cover all the language features that characterize the biomedical domain, such as the frequent use of acronyms and rhetorical expressions like synonymy, meronymy, etc. In addition, current benchmarks have a very limited sample size that make the analysis of results difficult. We conjecture that LiBlock, COM, and UBSM-Rada perform well because there is a noticeable overlap of terms that may benefit these methods over the others reported in the literature. Furthermore, Chen et al. [106] highlight the need to improve and create new benchmarks from different perspectives, to reflect the multifaceted notion of the similarity of sentences. Therefore, we found a strong need for improving existing benchmarks for the task of semantic similarity of sentences in the biomedical domain.

As part of our forthcoming activities, we plan to evaluate the new sentence similarity methods introduced herein in a benchmark for the general language domain. In addition, we will study the evaluation of sentence similarity methods in an extrinsic task, such as semantic medical indexing [107] or summarization [108]. We also consider the evaluation of further pre-processing configurations, such as biomedical NER systems based on recent Deep Learning techniques [10], or extending our experiments and research to the multilingual scenario by integrating multilingual biomedical NER systems like Cimind [109]. Finally, we plan to evaluate some recent biomedical concept embeddings based on MeSH [35], which has not been evaluated in the sentence similarity task yet.

Supporting information

S1 Appendix. The statistical significance results. We provide a series of tables reporting the p-values for each pair of methods evaluated in this work as supplementary material.
(PDF)

S2 Appendix. The pre-processing raw output files. We provide all the pre-processing raw output tables for the experiments evaluated herein as supplementary material.
(PDF)

S3 Appendix. A reproducibility protocol and dataset on the biomedical sentence similarity. We provide the reproducibility protocol published at protocols.io [44] as supplementary material to allow the exact replication of all our experiments, methods, and results.
(PDF)

Acknowledgments

We are grateful to Gizem Sogancioglu and Kathrin Blagec for answering kindly our questions to replicate their methods and experiments, Fernando González and Juan Corrales for setting up our reproducibility dataset, and Hongfang Liu and Yanshan Wang for providing us the MedSTS dataset. UMLS CUI codes, SNOMED-CT US ontology and MeSH thesaurus were used in our experiments by courtesy of the National Library of Medicine of the United States. Finally, we thank David Pritchard for checking the use of English in our manuscript.

Author Contributions

Conceptualization: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Data curation: Alicia Lara-Clares.

Formal analysis: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Funding acquisition: Ana Garcia-Serrano.

Investigation: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Methodology: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Resources: Alicia Lara-Clares.

Software: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Supervision: Juan J. Lastra-Díaz, Ana Garcia-Serrano.

Validation: Alicia Lara-Clares, Juan J. Lastra-Díaz.

Visualization: Alicia Lara-Clares.

Writing – original draft: Alicia Lara-Clares.

Writing – review & editing: Alicia Lara-Clares, Juan J. Lastra-Díaz.

References

1. Tafti AP, Behravesh E, Assefi M, LaRose E, Badger J, Mayer J, et al. bigNN: An open-source big data toolkit focused on biomedical sentence classification. In: 2017 IEEE International Conference on Big Data (Big Data); 2017. p. 3888–3896.
2. Kim S, Kim W, Comeau D, Wilbur WJ. Classifying gene sentences in biomedical literature by combining high-precision gene identifiers. In: Proc. of the 2012 Workshop on Biomedical Natural Language Processing; 2012. p. 185–192.
3. Chen Q, Panyam NC, Elangovan A, Davis M, Verspoor K. Document triage and relation extraction for protein-protein interactions affected by mutations. In: Proc. of the BioCreative VI Workshop. vol. 6; 2017. p. 52–51.
4. Sarrouti M, Ouatik El Alaoui S. A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering. J Biomedical Informatics. 2017; 68:96–103. https://doi.org/10.1016/j.jbi.2017.03.001 PMID: 28286031
5. Kosorus H, Bögl A, Küng J. Semantic Similarity between Queries in QA System using a Domain-specific Taxonomy. In: ICEIS (1); 2012. p. 241–246.
6. Ravikumar KE, Rastegar-Mojarad M, Liu H. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database. 2017; 2017(1). https://doi.org/10.1093/database/baw156 PMID: 28365720
7. Rastegar-Mojarad M, Komandur Elayavilli R, Liu H. BELTracker: evidence sentence retrieval for BEL statements. Database. 2016; 2016. https://doi.org/10.1093/database/baw079 PMID: 27173525
8. Du J, Chen Q, Peng Y, Xiang Y, Tao C, Lu Z. ML-Net: multi-label classification of biomedical texts with deep neural networks. J Am Med Inform Assoc. 2019; 26(11):1279–1285. https://doi.org/10.1093/jamia/ocz085 PMID: 31233120
9. Liu H, Hunter L, Kešelj V, Verspoor K. Approximate subgraph matching-based literature mining for biomedical events and relations. PLoS One. 2013; 8(4):e60954. https://doi.org/10.1371/journal.pone.0060954 PMID: 23613763
10. Hahn U, Oleynik M. Medical Information Extraction in the Age of Deep Learning. Yearb Med Inform. 2020; 29(1):208–220. https://doi.org/10.1055/s-0040-1702001 PMID: 32823318
11. Kim SN, Martinez D, Cavedon L, Yencken L. Automatic classification of sentences to support Evidence Based Medicine. BMC Bioinformatics. 2011; 12 Suppl 2:5. https://doi.org/10.1186/1471-2105-12-S2-S5 PMID: 21489224
12. Hassanzadeh H, Groza T, Nguyen A, Hunter J. A supervised approach to quantifying sentence similarity: with application to evidence based medicine. PLoS One. 2015; 10(6):e0129392. https://doi.org/10.1371/journal.pone.0129392 PMID: 26039310
13. Boyack KW, Newman D, Duhon RJ, Klavans R, Patek M, Biberstine JR, et al. Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. PLoS One. 2011; 6(3):e18029. https://doi.org/10.1371/journal.pone.0018029 PMID: 21437291
14. Dey S, Luo H, Fokoue A, Hu J, Zhang P. Predicting adverse drug reactions through interpretable deep learning framework. BMC Bioinformatics. 2018; 19(Suppl 21):476. https://doi.org/10.1186/s12859-018-2544-0 PMID: 30591036
15. Lamurias A, Ruas P, Couto FM. PPR-SSM: personalized PageRank and semantic similarity measures for entity linking. BMC Bioinformatics. 2019; 20(1):534. https://doi.org/10.1186/s12859-019-3157-y PMID: 31664891
16. Aliguliyev RM. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Syst Appl. 2009; 36(4):7764–7772. https://doi.org/10.1016/j.eswa.2008.11.022
17. Shang Y, Li Y, Lin H, Yang Z. Enhancing biomedical text summarization using semantic relation extraction. PLoS One. 2011; 6(8):e23862. https://doi.org/10.1371/journal.pone.0023862 PMID: 21887336
18. Allot A, Chen Q, Kim S, Vera Alvarez R, Comeau DC, Wilbur WJ, et al. LitSense: making sense of biomedical literature at sentence level. Nucleic Acids Res. 2019;. https://doi.org/10.1093/nar/gkz289 PMID: 31020319

19. Comeau DC, Wei CH, Islamaj Doğan R, Lu Z. PMC text mining subset in BioC: about three million full-text articles and growing. Bioinformatics. 2019;. https://doi.org/10.1093/bioinformatics/btz070 PMID: 30715220

20. Agirre E, Cer D, Diab M, Gonzalez-Agirre A. Semeval-2012 task 6: A pilot on semantic textual similarity. In: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proc. of the main conference and the shared task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). ACL; 2012. p. 385–393.

21. Agirre E, Cer D, Diab M, Gonzalez-Agirre A, Guo W. * SEM 2013 shared task: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proc. of the Main Conference and the Shared Task: Semantic Textual Similarity. vol. 1. ACL; 2013. p. 32–43.

22. Agirre E, Banea C, Cardie C, Cer D, Diab M, Gonzalez-Agirre A, et al. Semeval-2014 task 10: Multilingual semantic textual similarity. In: Proc. of the 8th international workshop on semantic evaluation (SemEval 2014). ACL; 2014. p. 81–91.

23. Agirre E, Banea C, Cardie C, Cer D, Diab M, Gonzalez-Agirre A, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In: Proc. of the 9th international workshop on semantic evaluation (SemEval 2015). ACL; 2015. p. 252–263.

24. Agirre E, Banea C, Cer D, Diab M, others. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. 10th International Workshop on Semantic Evaluation (SemEval-2016). 2016;.

25. Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 1–14.

26. Wang Y, Afzal N, Liu S, Rastegar-Mojarad M, Wang L, Shen F, et al. Overview of the BioCreative/OHNLP Challenge 2018 Task 2: Clinical Semantic Textual Similarity. Proc of the BioCreative/OHNLP Challenge. 2018; 2018.

27. Kalyan KS, Sangeetha S. SECNLP: A survey of embeddings in clinical natural language processing. J Biomed Inform. 2020; 101:103323. https://doi.org/10.1016/j.jbi.2019.103323 PMID: 31711972

28. Khattak FK, Jeblee S, Pou-Prom C, Abdalla M, Meaney C, Rudzicz F. A survey of word embeddings for clinical text. Journal of Biomedical Informatics: X. 2019; 4:100057. https://doi.org/10.1016/j.yjbinx.2019.100057 PMID: 34384583

29. Alsentzer E, Murphy J, Boag W, Weng WH, Jindi D, Naumann T, et al. Publicly Available Clinical BERT Embeddings. In: Proc. of the 2nd Clinical Natural Language Processing Workshop. Minneapolis, Minnesota, USA: Association for Computational Linguistics; 2019. p. 72–78.

30. Sogancioglu G, Öztürk H, Özgür A. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017; 33(14):49–58. https://doi.org/10.1093/bioinformatics/btx238 PMID: 28881973

31. Blagec K, Xu H, Agibetov A, Samwald M. Neural sentence embedding models for semantic similarity estimation in the biomedical domain. BMC Bioinformatics. 2019; 20(1):178. https://doi.org/10.1186/s12859-019-2789-2 PMID: 30975071

32. Peng Y, Yan S, Lu Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In: Proc. of the 18th BioNLP Workshop and Shared Task. Florence, Italy: Association for Computational Linguistics; 2019. p. 58–65.

33. Chen Q, Peng Y, Lu Z. BioSentVec: creating sentence embeddings for biomedical texts. In: 2019 IEEE International Conference on Healthcare Informatics (ICHI). IEEE; 2019. p. 1–5.

34. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010; 17(3):229–236. https://doi.org/10.1136/jamia.2009.002733 PMID: 20442139

35. Abdeddaïm S, Vimard S, Soualmia LF. The MeSH-Gram Neural Network Model: Extending Word Embedding Vectors with MeSH Concepts for Semantic Similarity. In: Ohno-Machado L, Séroussi B, editors. MEDINFO 2019: Health and Wellbeing e-Networks for All—Proceedings of the 17th World Congress on Medical and Health Informatics. vol. 264 of Studies in Health Technology and Informatics. IOS Press; 2019. p. 5–9.

36. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Research. 2005; 33(suppl1):D514–D517. https://doi.org/10.1093/nar/gki033 PMID: 15608251

37. Tawfik NS, Spruit MR. Evaluating Sentence Representations for Biomedical Text: Methods and Experimental Results. J Biomed Inform. 2020; p. 103396. https://doi.org/10.1016/j.jbi.2020.103396 PMID: 32147441

38. Chen Q, Du J, Kim S, Wilbur WJ, Lu Z. Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records. BMC Medical Informatics and Decision Making. 2020; 20(1):73. https://doi.org/10.1186/s12911-020-1044-0 PMID: 32349758

39. Breiman L. Random Forests. Machine Learning. 2001; 45(1):5–32. https://doi.org/10.1023/A:1010933404324

40. Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. Protocol for a reproducible experimental survey on biomedical sentence similarity. PLoS One. 2021; 16(3):e0248663. https://doi.org/10.1371/journal.pone.0248663 PMID: 33760855

41. Lastra-Díaz JJ, García-Serrano A, Batet M, Fernández M, Chirigati F. HESML: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems. 2017; 66:97–118. https://doi.org/10.1016/j.is.2017.02.002

42. Lastra-Díaz JJ, Lara-Clares A, Garcia-Serrano A. HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey. BMC Bioinformatics. 2022; 23(23). https://doi.org/10.1186/s12859-021-04539-0 PMID: 34991460

43. Lara-Clares A, Lastra Diaz JJ, Garcia Serrano A. Reproducible experiments on word and sentence similarity measures for the biomedical domain; 2022. e-cienciaDatos, v1. https://doi.org/10.21950/EPNXTR.

44. Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. A reproducibility protocol and dataset on the biomedical sentence similarity; 2022. Protocols.io, v1. https://www.protocols.io/view/a-reproducibility-protocol-and-dataset-on-the-biom-b5ckq2uw.

45. Lastra-Díaz JJ, García-Serrano A. A new family of information content models with an experimental survey on WordNet. Knowledge-Based Systems. 2015; 89:509–526. https://doi.org/10.1016/j.knosys.2015.08.019

46. Lastra-Díaz JJ, García-Serrano A. A novel family of IC-based similarity measures with a detailed experimental survey on WordNet. Engineering Applications of Artificial Intelligence Journal. 2015; 46:140–153. https://doi.org/10.1016/j.engappai.2015.09.006

47. Lastra-Díaz JJ, García-Serrano A. A refinement of the well-founded Information Content models with a very detailed experimental survey on WordNet. ETSI Informática. Universidad Nacional de Educación a Distancia (UNED). http://e-spacio.uned.es/fez/view/bibliuned:DptoLSI-ETSI-Informes-Jlastra-refinement; 2016. TR-2016-01.

48. Lastra-Diaz JJ, Goikoetxea J, Hadj Taieb MA, GarcÃa-Serrano A, Ben Aouicha M, Agirre E. A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence. 2019; 85:645–665. https://doi.org/10.1016/j.engappai.2019.07.010

49. Lastra-Díaz JJ, García-Serrano A. WordNet-based word similarity reproducible experiments based on HESML V1R1 and ReproZip; 2016. Mendeley Data, v1. http://doi.org/10.17632/65pxgskhz9.1.

50. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb MA, García-Serrano A, Aouicha MB, Agirre E. Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity. Data in Brief. 2019; 26:104432. https://doi.org/10.1016/j.dib.2019.104432 PMID: 31516953

51. Lastra-Díaz JJ, Goikoetxea J, Hadj Taieb M, García-Serrano A, Ben Aouicha M, Agirre E, et al. A large reproducible benchmark of ontology-based methods and word embeddings for word similarity. Information Systems. 2021; 96:101636. https://doi.org/10.1016/j.is.2020.101636

52. Wang Y, Afzal N, Fu S, Wang L, Shen F, Rastegar-Mojarad M, et al. MedSTS: a resource for clinical semantic textual similarity. Language Resources and Evaluation. 2018; p. 1–16.

53. Lithgow-Serrano O, Gama-Castro S, Ishida-Gutiérrez C, Mejía-Almonte C, Tierrafría VH, Martínez-Luna S, et al. Similarity corpus on microbial transcriptional regulation. Journal of Biomedical Semantics. 2019; 10(1):8. https://doi.org/10.1186/s13326-019-0200-x PMID: 31118102

54. Lithgow-Serrano O, Gama-Castro S, Ishida-Gutiérrez C, Collado-Vides J. L-Regulon: A novel soft-curation approach supported by a semantic enriched reading for RegulonDB literature. bioRxiv. 2020.

55. Gerlach M, Shi H, Amaral LAN. A universal information theoretic approach to the identification of stopwords. Nature Machine Intelligence. 2019; 1(12):606–612. https://doi.org/10.1038/s42256-019-0112-6

56. Li Y, McLean D, Bandar ZA, James DO, Crockett K. Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Trans Knowl Data Eng. 2006; 18(8):1138–1150. https://doi.org/10.1109/TKDE.2006.130

57. Krause EF. Taxicab Geometry: An Adventure in Non-Euclidean Geometry. Online: Courier Corporation; 1986.

58. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 2017; 5:135–146. https://doi.org/10.1162/tacl_a_00051

59. Song B, Li F, Liu Y, Zeng X. Deep learning methods for biomedical named entity recognition: a survey and qualitative comparison. Brief Bioinform. 2021; 22(6). https://doi.org/10.1093/bib/bbab282 PMID: 34308472

60. Miller GA. WordNet: A Lexical Database for English. ACM. 1995; 38(11):39–41. https://doi.org/10.1145/219717.219748

61. Donnelly K. SNOMED-CT: The advanced terminology and coding system for eHealth. Books Google. 2006; 121:279–290. PMID: 17095826

62. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010; 17(5):507–513. https://doi.org/10.1136/jamia.2009.001560 PMID: 20819853

63. Dijkstra EW. A note on two problems in connexion with graphs. Numerische Mathematik. 1959; 1(1):269–271. https://doi.org/10.1007/BF01386390

64. Johnson AEW, Pollard TJ, Shen L, Lehman LWH, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Sci Data. 2016; 3:160035. https://doi.org/10.1038/sdata.2016.35 PMID: 27219127

65. Mikolov T, Sutskever I, Chen K, Corrado GS, others. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. 2013;.

66. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP). ACL Web; 2014. p. 1532–1543.

67. Sánchez D, Batet M, Isern D. Ontology-based information content computation. Knowledge-Based Systems. 2011; 24(2):297–303. https://doi.org/10.1016/j.knosys.2010.10.001

68. Cai Y, Zhang Q, Lu W, Che X. A hybrid approach for measuring semantic similarity based on IC-weighted path distance in WordNet. Journal of intelligent information systems. 2017; p. 1–25.

69. Rada R, Mili H, Bicknell E, Blettner M. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics. 1989; 19(1):17–30. https://doi.org/10.1109/21.24528

70. Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of International Conference Research on Computational Linguistics (ROCLING X); 1997. p. 19–33.

71. Chapman S, Norton B, Ciravegna F. Armadillo: Integrating knowledge for the semantic web. In: Proceedings of the Dagstuhl Seminar in Machine Learning for the Semantic Web. Researchgate; 2005. p. 90.

72. Ukkonen E. Approximate string-matching with q-grams and maximal matches. Theor Comput Sci. 1992; 92(1):191–211. https://doi.org/10.1016/0304-3975(92)90143-4

73. ,Jaccard P. Nouvelles recherches sur la distribution florale. Bull Soc Vaud sci nat. 1908; 44:223–270.

74. Manning CD, Manning CD, Schütze H. Foundations of Statistical Natural Language Processing. Online: MIT Press; 1999.

75. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10. Springer; 1966. p. 707–710.

76. Lawlor LR. Overlap, Similarity, and Competition Coefficients. Ecology. 1980; 61(2):245–251. https://doi.org/10.2307/1935181

77. Akbik A, Blythe D, Vollgraf R. Contextual String Embeddings for Sequence Labeling. In: Proc. of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics; 2018. p. 1638–1649.

78. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. Distributional semantics resources for biomedical text processing. Proc of LBM. 2013; p. 39–44.

79. Chen Q, Lee K, Yan S, Kim S, Wei CH, Lu Z. BioConceptVec: Creating and evaluating literature-based biomedical concept embeddings on a large scale. PLOS Computational Biology. 2020; 16(4):1–18. https://doi.org/10.1371/journal.pcbi.1007617 PMID: 32324731

80. Newman-Griffis D, Lai A, Fosler-Lussier E. Insights into Analogy Completion from the Biomedical Domain. In: BioNLP 2017. Vancouver, Canada,: Association for Computational Linguistics; 2017. p. 19–28.

81. Zhang Y, Chen Q, Yang Z, Lin H, Lu Z. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Sci Data. 2019; 6(1):52. https://doi.org/10.1038/s41597-019-0055-0 PMID: 31076572

82. Chiu B, Crichton G, Korhonen A, Pyysalo S. How to Train good Word Embeddings for Biomedical NLP. In: Proc. of the 15th Workshop on Biomedical Natural Language Processing. Berlin, Germany: Association for Computational Linguistics; 2016. p. 166–174.

83. Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, St John R, et al. Universal Sentence Encoder for English. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 169–174.

84. Pagliardini M, Gupta P, Jaggi M. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 528–540.

85. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2019; 36(4):1234–1240.

86. Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein J, Doran C, Solorio T, editors. Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, (Long and Short Papers). Minneapolis, MN, USA: Association for Computational Linguistics; 2019. p. 4171–4186. Available from: https://doi.org/10.18653/v1/n19-1423.

87. Beltagy I, Lo K, Cohan A. SciBERT: A Pretrained Language Model for Scientific Text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019. p. 3615–3620.

88. Huang K, Altosaar J, Ranganath R. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. arXiv e-prints. 2019; p. arXiv:1904.05342.

89. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, et al. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. arXiv e-prints. 2020; p. arXiv:2007.15779.

90. Wada S, Takeda T, Manabe S, Konishi S, Kamohara J, Matsumura Y. A pre-training technique to localize medical BERT and to enhance biomedical BERT. arXiv e-prints. 2020; p. arXiv:2005.07202.

91. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv. 2016;.

92. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP natural language processing toolkit. In: Proc. of 52nd annual meeting of the association for computational linguistics: system demonstrations. ACL; 2014. p. 55–60.

93. Comeau DC, Islamaj Doğan R, Ciccarese P, Cohen KB, Krallinger M, Leitner F, et al. BioC: a minimalist approach to interoperability for biomedical text processing. Database. 2013; 2013:bat064. https://doi.org/10.1093/database/bat064 PMID: 24048470

94. Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. J Am Med Inform Assoc. 2017; 24(4):841–844. https://doi.org/10.1093/jamia/ocw177 PMID: 28130331

95. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004; 32(Database issue):267–70. https://doi.org/10.1093/nar/gkh061 PMID: 14681409

96. Lastra-Díaz JJ, Lara-Clares A, Garcia-Serrano A. HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models; 2020. e-cienciaDatos, v1. https://doi.org/10.21950/1RRAWJ.

97. Smith L, Rindflesch T, Wilbur WJ. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics. 2004; 20(14):2320–2321. https://doi.org/10.1093/bioinformatics/bth227 PMID: 15073016

98. Reátegui R, Ratté S. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. BMC Med Inform Decis Mak. 2018; 18(Suppl 3):74. https://doi.org/10.1186/s12911-018-0654-2 PMID: 30255810

99. Bird S, Klein E, Loper E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc.; 2009.

100. Ludbrook J. Multiple comparison procedures updated. Clinical and experimental pharmacology & physiology. 1998; 25(12):1032–1037. https://doi.org/10.1111/j.1440-1681.1998.tb02179.x PMID: 9888002

101. Shen D, Wang G, Wang W, Min MR, Su Q, Zhang Y, et al. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In: Proceedings of the 56th

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 440–450.

102. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation OSDI 16). usenix.org; 2016. p. 265–283.

103. Xiao H. bert-as-service; 2018. https://github.com/hanxiao/bert-as-service.

104. Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. HESML Java software library of semantic similarity measures for the biomedical domain. To be submitted. 2020.

105. Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. HESML V2R1 Java software library of semantic similarity measures for the biomedical domain; 2022. e-cienciaDatos, v2. https://doi.org/10.21950/AQLSMV.

106. Chen Q, Rankine A, Peng Y, Aghaarabi E, Lu Z. Benchmarking Effectiveness and Efficiency of Deep Learning Models for Semantic Textual Similarity in the Clinical Domain: Validation Study. JMIR Medical Informatics. 2021; 9(12):e27386. https://doi.org/10.2196/27386 PMID: 34967748

107. Couto FM, Krallinger M. Proposal of the First International Workshop on Semantic Indexing and Information Retrieval for Health from Heterogeneous Content Types and Languages (SIIRH). In: Advances in Information Retrieval. Springer International Publishing; 2020. p. 654–659.

108. Mishra R, Bian J, Fiszman M, Weir CR, Jonnalagadda S, Mostafa J, et al. Text summarization in the biomedical domain: a systematic review of recent research. J Biomed Inform. 2014; 52:457–467. https://doi.org/10.1016/j.jbi.2014.06.009 PMID: 25016293

109. Cabot C, Darmoni S, Soualmia LF. Cimind: A phonetic-based tool for multilingual named entity recognition in biomedical texts. J Biomed Inform. 2019; 94:103176. https://doi.org/10.1016/j.jbi.2019.103176 PMID: 30980962

Chapter 10

Information Systems article

Reproducible experiments on Three-Dimensional Entity Resolution with JedAI

George Mandilaras[1], George Papadakis[1*], Luca Gagliardelli[2], Giovanni Simonini[2],
Emmanouil Thanos[3], George Giannakopoulos[4], Sonia Bergamaschi[2], Themis Palpanas[5],
Manolis Koubarakis[1], Alicia Lara-Clares[6**], Antonio Fariña[7**]

[1] *National and Kapodistrian University of Athens, Greece* {gmandi,gpapadis,koubarak}@di.uoa.gr
[2] *University of Modena and Reggio Emilia, Italy* {name.surname}@unimore.it
[3] *KU Leuven, Belgium* emmanouil.thanos@kuleuven.be
[4] *NCSR "Demokritos", Greece* ggianna@iit.demokritos.gr
[5] *University of Paris & French University Institute (IUF), France* themis@mi.parisdescartes.fr
[6] *NLP&IR Research Group, Universidad Nacional de Educación a Distancia (UNED), Spain* alara@lsi.uned.es
[7] *University of A Coruña, CITIC, Database Lab, Spain* antonio.farina@udc.es

Abstract

In Papadakis et al. [1], we presented the latest release of JedAI, an open-source Entity Resolution (ER) system that allows for building a large variety of end-to-end ER pipelines. Through a thorough experimental evaluation, we compared a schema-agnostic ER pipeline based on blocks with another schema-based ER pipeline based on similarity joins. We applied them to 10 established, real-world datasets and assessed them with respect to effectiveness and time efficiency. Special care was taken to juxtapose their scalability, too, using seven established, synthetic datasets. Moreover, we experimentally compared the effectiveness of the batch schema-agnostic ER pipeline with its progressive counterpart. In this companion paper, we describe how to reproduce the entire experimental study that pertains to JedAI's serial execution through its intuitive user interface. We also explain how to examine the robustness of the parameter configurations we have selected.

Keywords: Entity Resolution, Batch Methods, Progressive Methods, Reproducibility

1. Introduction

Entity Resolution (ER) is the task of identifying *matches* or *duplicates*, i.e., different entity profiles that describe the same real-world object. For example, ER should match the entity profiles `https://www.wikidata.org/wiki/Q30` and `https://en.wikipedia.org/wiki/United_States`, which refer to the United States of America in two different data sources, Wikidata[1] and Wikipedia[2] respectively. ER constitutes a core data integration task and, thus, numerous approaches for tackling it have been proposed in the literature. Overviews of the main methods can be found in recent books [2, 3, 4, 5], surveys [6, 7, 8] and tutorials [9, 10, 11, 12].

To facilitate the use of the main ER methods, we created JedAI [1], an open-source system that allows for building end-to-end pipelines. JedAI enables users to effectively address the ER problem by categorizing the main methods into three orthogonal dimensions:

1. *Schema-awareness* categorizes ER methods into *schema-based* and *schema-agnostic* ones, depending on whether they rely on schema knowledge or not.
2. *Budget-awareness* categorizes ER methods into *budget-agnostic* ones, which operate as batch processes, and *budget-aware* ones, which operate in a pay-as-you-go manner that produces results progressively — they maximize the detected matches within a specific budget of temporal or computational resources.
3. *Execution mode* categorizes ER methods into serial and massively parallelized ones, e.g., over Apache Spark.[3]

Using JedAI, we experimentally evaluated in [1] the relative performance of the main end-to-end ER pipelines that are defined by the three aforementioned dimensions. In this work, we focus on serially executed pipelines of any type.

Regarding schema-awareness, the **schema-agnostic pipeline** consists of the following steps, as shown in Figure 1(a):

- *Data Reading* loads the data to be processed into main memory.
- *Schema Clustering* is an optional step that groups together different attributes that share syntactically similar values so as to improve the performance of the subsequent steps. Note that this task differs from Schema Matching, which tries to identify the semantically matching attributes.

[*] Corresponding author
[**] Reviewer
[1] `https://www.wikidata.org`
[2] `https://www.wikipedia.org`
[3] `https://spark.apache.org`

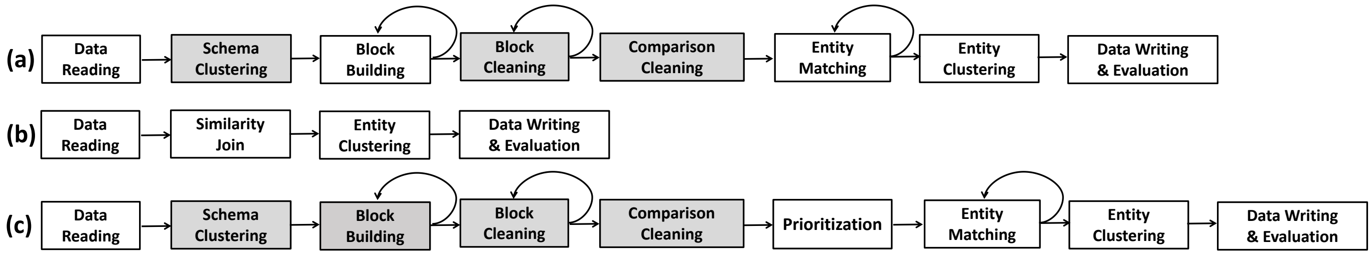


Figure 1: The three main end-to-end ER pipelines implemented by JedAI: (a) the budget- & schema-agnostic one, (b) the budget-agnostic, schema-based one, and (c) the budget-aware, schema-agnostic one. Shaded rectangles indicate optional steps.

- *Block Building* aims to reduce the computational cost of the brute-force approach, by limiting the search space to similar entity profiles. To this end, it clusters together entity profiles that share identical or similar signatures.

- *Block Cleaning* is an optional step that further curtails the computational cost of ER by refining the output of Block Building. Its goal is actually to discard those blocks that are dominated by *redundant* and *superfluous comparisons*; the former involve pairs of entities co-occurring in multiple blocks, while the latter compare pairs of entities that do not match.

- *Comparison Cleaning* is another optional step that serves the same purpose as Block Cleaning. It offers a more time-consuming, but more precise functionality that operates at the level of individual comparisons.

- *Entity Matching* estimates the matching likelihood for all entity pairs in the final set of blocks, using string similarity measures.

- *Entity Clustering* models the estimated similarities as a weighted, undirected graph and then partitions it into *equivalence clusters*, i.e., disjoint sets of entity profiles that are considered as matches.

- *Data Writing & Evaluation* allows for storing the final results and for assessing the performance of the selected ER pipeline with respect to the main effectiveness and time efficiency measures.

The **schema-based end-to-end pipeline** also starts with Data Reading and ends with Entity Clustering and Data Writing & Evaluation, as shown in Figure 1(b). In between, it applies a single step, called *Similarity Join*, which rapidly estimates the pairs of entity profiles that satisfy a given *matching rule*, which consists of:

1. a similarity measure,
2. the attribute on which the measure is applied, and
3. a threshold designating the minimum acceptable similarity for two entity profiles that are considered as matching.

As an example, consider the following matching rule for bibliographic entities: $JaccardSim(title_1, title_2) > 0.8$.

In [1], we also compare the batch, schema-agnostic pipeline with its progressive counterpart, i.e., the **budget-aware, schema-agnostic pipeline**, which is shown in Figure 1(c). The only difference from the batch pipeline is the *Prioritization* step, which intervenes between Comparison Cleaning and Entity Matching. Its goal is to define the optimal processing order of the entity pairs in the final set of blocks so that the matching ones are detected as early as possible.

A video demonstrating JedAI in action is available at: `https://www.youtube.com/watch?v=OJY1DUrUAe8`

2. The reproducible experiments on Entity Resolution

2.1. Preliminaries

Depending on the input data, Entity Resolution is categorized into two main categories:

1. *Clean-Clean ER* receives as input two datasets, which are individually duplicate-free (e.g., Wikipedia and Wikidata), and its goal is to identify the matches they share.
2. *Dirty ER* receives as input one or more datasets, with at least one of them containing duplicates in itself. Its goal is to partition all entity profiles into equivalence clusters.

In both cases, the end-result of any end-to-end pipeline is evaluated with respect to three **effectiveness** measures:

- *Recall* assesses the portion of existing duplicates that are actually identified as such.

- *Precision* estimates the portion of entity pairs that are marked as matches and are indeed duplicates.

- *F-Measure* is the harmonic mean of Recall and Precision.

The progressive pipelines are additionally assessed through *Progressive Recall*, which quantifies the evolution of recall as more entity pairs are compared. We actually consider the area under its curve (AUC), which is derived from a two-dimensional diagram, where horizontal axis corresponds to the number of executed comparisons and the vertical one to the number of detected duplicates. The larger (the area under the curve of) Progressive Recall is, the earlier are the matches identified and the better is the progressive pipeline.

All effectiveness measures are defined in the interval $[0, 1]$, with higher values corresponding to higher effectiveness.

The **time efficiency** of an end-to-end pipeline is measured through its *run-time*, i.e., the time that intervenes between receiving the input entity profiles and producing the end result.

Table 1: Technical characteristics of the Dirty ER datasets. $|E|$ stands for the number of entity profiles, NVP for the total number of name-value pairs in the dataset, $|N|$ for the number of distinct attributes, $|\bar{p}|$ for the average profile size (in terms of name-value pairs), $|D(E)|$ for the number of duplicate pairs, and $||E||$ for the comparisons executed by the brute-force approach.

	$\mathbf{D_{cora}}$	$\mathbf{D_{cddb}}$	$\mathbf{D_{10K}}$	$\mathbf{D_{50K}}$	$\mathbf{D_{100K}}$	$\mathbf{D_{200K}}$	$\mathbf{D_{300K}}$	$\mathbf{D_{1M}}$	$\mathbf{D_{2M}}$
$\|E\|$	1,295	9,763	10,000	50,000	100,000	200,000	300,000	1,000,000	2,000,000
NVP	7,166	183,072	106,108	530,854	1,061,421	2,123,728	3,184,885	10,617,729	21,238,252
$\|N\|$	12	106	12	12	12	12	12	12	12
$\|\bar{p}\|$	5.53	18.75	10.61	10.62	10.61	10.62	10.62	10.62	10.62
$\|D(E)\|$	17,184	299	8,705	43,071	85,497	172,403	257,034	857,538	1,716,102
$\|\|E\|\|$	$8.38{\cdot}10^5$	$4.77{\cdot}10^7$	$5.00{\cdot}10^7$	$1.25{\cdot}10^9$	$5.00{\cdot}10^9$	$2.00{\cdot}10^{10}$	$4.50{\cdot}10^{10}$	$5.00{\cdot}10^{11}$	$2.00{\cdot}10^{12}$

Table 2: Technical characteristics of the Clean-Clean ER datasets.

	$\mathbf{D_{c1}}$	$\mathbf{D_{c2}}$	$\mathbf{D_{c3}}$	$\mathbf{D_{c4}}$	$\mathbf{D_{c5}}$	$\mathbf{D_{c6}}$	$\mathbf{D_{c7}}$	$\mathbf{D_{c8}}$
Dataset$_1$	Rest.1	Abt	Amazon	DBLP	Walmart	DBLP	DBPedia	DBPedia 3.0rc
Dataset$_2$	Rest.2	Buy	Google Pr.	ACM	Amazon	Scholar	IMDB	DBPedia 3.4
$\|E_1\|/\|E_2\|$	339/2,256	1,076/1,076	1,354/3,039	2,616/2,294	2,554/22,074	2,516/61,353	27,615/23,182	$1.19{\cdot}10^6/2.16{\cdot}10^6$
NVP$_1$/NVP$_2$	1,130/7,519	2,568/2,308	5,302/9,110	10,464/9,162	$14{,}143/1.1{\cdot}10^5$	$10{,}064/2{\cdot}10^5$	$1.6{\cdot}10^5/8.2{\cdot}10^5$	$1.69{\cdot}10^7/3.50{\cdot}10^7$
$\|N_1\|/\|N_2\|$	7/7	3/3	4/4	4/4	6/6	4/4	4/7	30,688/52,489
$\|\bar{p}_1\|/\|\bar{p}_2\|$	3.33/3.33	2.39/2.14	3.92/3.00	3.99/4.00	5.54/5.18	3.23/3.26	5.63/35.20	14.19/16.18
$\|D(E_1 \cap E_2)\|$	89	1,076	1,104	2,224	853	2,308	22,863	892,579
$\|\|E_1 \times E_2\|\|$	$7.65{\cdot}10^5$	$1.16{\cdot}10^6$	$4.11{\cdot}10^6$	$6.00{\cdot}10^6$	$5.64{\cdot}10^7$	$1.54{\cdot}10^8$	$6.40{\cdot}10^8$	$2.58{\cdot}10^{12}$

Note that we also provide the minimum amount of main memory that is required to successfully run each test in a way that approximates the lowest possible running time by minimizing the impact of the garbage collector. The reported values correspond to the $-Xmx$ parameter when running each experiment as a Java process, independently of Docker and the browser, which raise additional memory requirements.

2.2. Sets of Experiments

The experimental analysis of [1] used 17 datasets. Each of them consists of one or two sets of entity profiles, in the case of Dirty and Clean-Clean ER, respectively, as well as a golden standard, i.e., the complete ground-truth of the actual duplicate entity profiles. They are all publicly available in the form of Java serialized objects as a Mendeley dataset [13] and through JedAI's repository.[4] Their technical characteristics are reported in Tables 1 and 2, which are the same as Tables 1 and 2 in [11], but are repeated here for convenience. Additional information about all datasets is provided in Table 3.

Our experiments are divided into three sets as follows:

1. The *Performance Tests* examine the relative performance of the two budget-agnostic pipelines - the schema-based and the schema-agnostic one.
2. The *Scalability Tests* examine how the performance of the two budget-agnostic pipelines evolves as the size of the input data increases.
3. The *Budget-awareness Tests* examine the relative performance of the two forms of the schema-agnostic pipeline: the budget-agnostic and the budget-aware.

Below, we describe every set of experiments in more detail.

Performance Tests. These experiments, which are reported in Table 4 of [1], compare the schema- and budget-agnostic pipeline with its schema-based counterpart over 10 real-world datasets. Two of them pertain to Dirty ER (D_{cora} and D_{cddb}) and the rest to Clean-Clean ER (D_{c1}-D_{c8}). The goal of these experiments is to evaluate both the relative effectiveness and the relative time efficiency of these pipelines. For the schema-agnostic pipeline, we consider two configurations:

1. the *best* one, which uses the parameters that maximize the F-Measure per dataset, and
2. the *default* one, which uses the default parameters for each method in the pipeline, thus being the same for all datasets.

For the schema-based pipeline, we exclusively consider the best configuration per dataset, which maximizes F-Measure.

Note that these tests involve two baseline systems that have been developed by other research groups, Magellan [26] and DeepMatcher [27]. Due to their human-in-the-loop approach and the lack of necessary details, we could not test their performance ourselves. Instead, we reported their top F-measure per dataset in [27], among all configurations and dataset versions. For this reason, we disregard both systems in the following.

Scalability Tests. These experiments are described in the diagrams of Figure 7 in [1], comparing again the two budget-agnostic end-to-end pipelines. In this case, though, the goal is to assess how their time efficiency and effectiveness evolve as the size of the data increase from several thousand to few million entity profiles. To this end, we use seven datasets that pertain exclusively to Dirty ER; their names indicate their size, i.e., the number of their entity profiles: D_{10K}, D_{50K}, D_{100K}, D_{200K}, D_{300K}, D_{1M} and D_{2M}. These datasets contain synthetic census data, i.e., information about individuals that has been enriched with various forms of artificial

[4] https://github.com/scify/JedAIToolkit

Table 3: Core information about each dataset: its reference work, its type (i.e., whether it involves real or synthetic data), the corresponding ER task (Clean-Clean or Dirty ER), the paths of its entity profiles and its golden standard files in the data repository of [13] and the original data source. We have categorized the 17 datasets in three groups according to their type and task, following [13], which contains a different folder for each group. Note that in [13], all parts of D_{c8} are provided through a single zipped file, `newDBPedia.tar.xz`, to minimize their large size.

Dataset	Type	Task	Path to the Entity Profiles File in [13]	Path to the Golden Standard File in [13]	Source
D_{c1} [14]	Real	Clean-Clean ER	Real Clean-Clean ER data/restaurant1Profiles Real Clean-Clean ER data/restaurant2Profiles	Real Clean-Clean ER data/restaurant1IdDuplicates	[15]
D_{c2} [16]	Real	Clean-Clean ER	Real Clean-Clean ER data/abtProfiles Real Clean-Clean ER data/buyProfiles	Real Clean-Clean ER data/abtBuyIdDuplicates	[17]
D_{c3} [16]	Real	Clean-Clean ER	Real Clean-Clean ER data/amazonProfiles Real Clean-Clean ER data/gpProfiles	Real Clean-Clean ER data/amazonGpIdDuplicates	[17]
D_{c4} [16]	Real	Clean-Clean ER	Real Clean-Clean ER data/dblpProfiles Real Clean-Clean ER data/acmProfiles	Real Clean-Clean ER data/dblpAcmProfiles	[17]
D_{c5} [18]	Real	Clean-Clean ER	Real Clean-Clean ER data/walmartProfiles Real Clean-Clean ER data/amazonProfiles2	Real Clean-Clean ER data/amazonWalmartIdDuplicates	[19]
D_{c6} [16]	Real	Clean-Clean ER	Real Clean-Clean ER data/dblpProfiles2 Real Clean-Clean ER data/scholarProfiles	Clean-Clean ER data/dblpScholarIdDuplicates	[17]
D_{c7} [20]	Real	Clean-Clean ER	Real Clean-Clean ER data/imdbProfiles Real Clean-Clean ER data/dbpediaProfiles	Clean-Clean ER data/moviesIdDuplicates	[21]
D_{c8} [20]	Real	Clean-Clean ER	Real Clean-Clean ER data/cleanDBPedia1 Real Clean-Clean ER data/cleanDBPedia2	Clean-Clean ER data/newDBPediaMatches	[21]
D_{cora} [22]	Real	Dirty ER	Real Dirty ER data/coraProfiles	Real Dirty ER data/coraIdDuplicates	[23]
D_{cddb} [24]	Real	Dirty ER	Real Dirty ER data/cddbProfiles	Real Dirty ER data/cddbIdDuplicates	[23]
D_{10K} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/10Kprofiles	Synthetic Dirty ER data/10KIdDuplicates	[21]
D_{50K} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/50Kprofiles	Synthetic Dirty ER data/50KIdDuplicates	[21]
D_{100K} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/100Kprofiles	Synthetic Dirty ER data/100KIdDuplicates	[21]
D_{200K} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/200Kprofiles	Synthetic Dirty ER data/200KIdDuplicates	[21]
D_{300K} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/300Kprofiles	Synthetic Dirty ER data/300KIdDuplicates	[21]
D_{1M} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/1Mprofiles	Synthetic Dirty ER data/1MIdDuplicates	[21]
D_{2M} [25]	Synthetic	Dirty ER	Synthetic Dirty ER data/2Mprofiles	Synthetic Dirty ER data/2MIdDuplicates	[21]

noise (see [1] for more details). For both pipelines, we consider a single configuration that is applied to all datasets: the default configuration for the schema-agnostic pipeline and the matching rule that consistently achieves reasonable performance across all datasets for the schema-based one, i.e., $JaccarSim(all\_tokens\_1, all\_tokens\_2) > 0.4$, executed by PPJoin and followed by Connected Components with the same similarity threshold.

Budget-awareness Tests. These experiments are reported in the diagrams of Figure 8 in [1]. They compare the budget- and schema-agnostic pipeline with its budget-aware counterpart across the same datasets as the Performance Tests - except the largest one, D_{c8}. For each dataset, the parameter configuration that corresponds to the optimal performance of the budget- and schema-agnostic pipeline is also used for the common methods of its budget-aware version. In this way, these tests assess the impact of the Prioritization step, which constitutes the sole difference between the two pipelines. We evaluate the time efficiency of the two workflows through their running times and the effectiveness through the area under their Progressive Recall.

2.3. Experimental setup in our primary paper

All single-core experiments in [1] were implemented in Java 8 and can be reproduced through JedAI's Docker image, which is publicly available.[5] The only requirement is to have Docker[6] installed. Table 4 provides detailed instructions for installing the latest version of Docker on Ubuntu. A similar procedure is required for other Linux distributions, like Debian,[7] Fedora[8] and CentOS.[9] JedAI's Docker image is expected to run seamlessly in all these cases. Upon successful completion of these commands, JedAI's Web application appears in a browser at: `http://localhost:8080`.

Note that the option `-e JAVA_OPTIONS=‘-Xmx4g’` determines that 4 Gigabytes (GB) of RAM memory is allocated to Java to run JedAI's Web application. This is an optional parameter, as the vast majority of our experiments can be run with much fewer memory, as indicated by the memory requirements that are reported in Tables 8, 9 and 10 for each experiment. In our tests, though, we noticed that 4GB are more suitable for ensuring Docker's stability. Otherwise, it needs restarting after some tests. When experimenting with larger datasets, it is actually recommended to devote all or most of the available memory to Docker so as to avoid out-of-memory exceptions or excessively large running times, due to the overuse of the garbage collector.

Note also that the option `-v /absolute/path` is necessary because JedAI's Docker starts by downloading all datasets from the Mendeley data repository [13]. Thus, this option determines the directory on the host system (e.g., `/home/user/jedai`),

[5] `https://hub.docker.com/repository/docker/gmandi/jedai-webapp`

[6] `https://www.docker.com`

[7] See `https://docs.docker.com/engine/install/debian` for detailed instructions.

[8] See `https://docs.docker.com/engine/install/fedora` for detailed instructions.

[9] See `https://docs.docker.com/engine/install/centos` for detailed instructions.

Table 4: Detailed instructions for installing and running JedAI's Docker image on Ubuntu. The steps 1-7 install the latest version of Docker Community Edition. For more details, please refer to the official Docker setup page at: `https://docs.docker.com/engine/install/ubuntu`. The remaining steps download JedAI's Docker image from the Docker Hub (step 8) or from JedAI's Mendeley data repository (step 8') and execute it (step 9).

Step	Setup instructions
	Update the apt package index.
(1)	$ sudo apt-get update
	Install packages to allow apt to use a repository over HTTPS.
(2)	$ sudo apt-get -y install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
	Add Docker's official GPG key.
(3)	$ curl -fsSL `https://download.docker.com/linux/ubuntu/gpg` | sudo apt-key add -
	Set up the stable repository.
(4)	$ sudo add-apt-repository "deb [arch=amd64] `https://download.docker.com/linux/ubuntu` $(lsb_release -cs) stable"
	Update the apt package index.
(5)	$ sudo apt-get update
	Install the latest version of Docker Engine.
(6)	$ sudo apt-get -y install docker-ce docker-ce-cli containerd.io
	Verify that Docker Engine is installed correctly.
(7)	$ sudo docker run hello-world
	Download the latest JedAI Docker image from Docker Hub.
(8)	$ sudo docker pull gmandi/jedai-webapp:latest
(8')	*Alternatively, download JedAI's Docker image from the Mendeley dataset.* wget -O jedai.tar `https://data.mendeley.com/public-files/datasets/4whpm32y47/files/79f5ccdd-e60a-4f9c-99cb-8f2d7ef0fc25/file_downloaded` $ sudo docker load < jedai.tar
	Launch the JedAI Web application. *Note that parameter -Xmx4g allows JedAI to use up to 4Gb RAM. Higher values can be used if more main memory is available.* *Note also that parameter -v should point to a directory, e.g., `/home/user/jedai`, with user-write permissions.*
(9)	$ sudo docker run -e 'JAVA_OPTIONS=-Xmx4g' -p 8080:8080 -v /absolute/path gmandi/jedai-webapp

where Docker will store and unpack the dataset files as long as it has user-write permissions.

It is also worth noting that in the option `-p 8080:8080`, the first 8080 refers to the host port, and could be replaced by any other free port in the host. Docker will map the first port 8080 to the http port (second 8080) from the docker container.

Finally, it is worth noting that it is also possible to use Docker on Windows 10. The installation is a straightforward procedure[10] that merely needs some additional steps.[11] After the successful installation, all experiments can be seamlessly run, without any performance issue. Indeed, one of our testing platforms runs on Windows 10 Pro ($Windows - base1$ in Table 5).

2.4. System requirements and performance evaluation

All single-core experiments in [1] can be reproduced on any Java 8 compliant platform, which practically includes all major Linux distributions. Our experiments have been successfully reproduced on all testing platforms reported in Table 5, with the aggregate running times that are reported in Table 6. Note that in all systems, a single CPU core was used for each experiment.

Our original configuration corresponds to $Ubuntu-base1$ for the Performance and Scalability Tests and to $Ubuntu - base1'$ for the Budget-awareness Tests. $Ubuntu - base2$ is a similar server but with a different CPU that accounts for significant diversity in the running times. A more important difference is that in $Ubuntu - base1$ and $Ubuntu - base1'$, all experiments were run through script files,[12] whereas in $Ubuntu - base2$, the experiments were carried out through the user interface of JedAI's Web application. The same applies to all other systems.

Among the other platforms, it is worth stressing that $Ubuntu-base4$ consists of a bootable USB stick that runs a live Ubuntu instance on top of a Windows 10 laptop. The only implication was that it required a different approach for installing Docker.[13] No performance issue arose. In fact, $Ubuntu - base4$ is often one of the fastest testing platforms, due to the newer generation of CPU and RAM technology.

Regarding the minimum system specifications required by our experiments, the size of the hard disk plays a minor role. Given that all experiments are executed in main memory and produce no output files, the hard disk requirements are determined by the space occupied by the Java JDK and the Docker installation as well as the size of JedAI's Docker image, which also includes all datasets. In total, this amounts to around 4 GB, assuming an underlying blank Ubuntu installation. Note, though, that this space is occupied whenever command 9 in Table 4 is executed. To recover the space occupied after multiple

[10] See `https://docs.docker.com/docker-for-windows/install` for detailed instructions.

[11] See `https://docs.docker.com/docker-for-windows/wsl` for more details.

[12] The source code of all tests is available at: `https://github.com/scify/JedAIToolkit/tree/master/src/test/java/org/scify/jedai/version3`.

[13] For more details, please refer to `https://stackoverflow.com/questions/30248794/run-docker-in-ubuntu-live-disk`.

Table 5: The testing platforms that were successfully used to reproduce our experiments. Note that $Ubuntu - base1$ was used in [1] for performing the experiments reported in Tables 8 and 9, while $Ubuntu - base1'$ was only used for the experiments in Table 10.

Testing platform	Type	Software Configuration	Hardware Configuration	Tested by
$Ubuntu - base1$	Server	Ubuntu 14.04.5 LTS OpenJDK 1.8.0	1 Intel Xeon E5-4603 v2 @2.20GHz, 128 Gb DDR3 RAM, 1.6 Tb mechanical disk	Authors
$Ubuntu - base1'$	Server	Ubuntu 14.04 LTS Java 1.8.0	1 Intel Xeon E5-2670 v2 @2.50GHz, 80GB DDR3 RAM, 1Tb mechanical disk	Authors
$Ubuntu - base2$	Server	Ubuntu 14.04.6 LTS Docker 19.03.13, Java 1.8.0	1 AMD Opteron 6320 @2.80GHz, 128 Gb DDR3 RAM, 1.6 Tb mechanical disk	Authors
$Ubuntu - base3$	Laptop	Ubuntu 18.04.5 LTS Docker 20.10.5, Java 1.8.0	1 Intel Core i7-4710MQ @2.50GHz, 16 Gb DDR3 RAM, 120 Gb SSD	Authors
$Ubuntu - base4$	Laptop	Ubuntu 20.04 LTS Docker 19.03.8, OpenJDK 1.8.0	1 Intel Core i5-1035G1 @1.00GHz, 4 Gb DDR4 RAM, 32 Gb flash drive	Authors
$Ubuntu - base5$	Laptop	Linux Mint 19.1 Tessa Docker 19.03.8, Java 1.8.0	1 Intel Core i7-3770 @3.40GHz, 16 Gb DDR3 RAM, 1 Tb mechanical disk	Authors
$Ubuntu - base6$	Laptop	Ubuntu 20.04.2 LTS Docker 19.03.14, OpenJDK 1.8.0	Intel Core i7-9750H @2.60GHz, 32 GB RAM, 2.5Tb mechanical disk	Reviewer
$Ubuntu - base7$	Server	Ubuntu 20.04.2 LTS	1 Intel Xeon Bronze 3204 @1.9GHz, 512 Gb DDR4 RAM,120Gb mechanical disk	Reviewer
$Ubuntu - base8$	Server	Ubuntu 16.04.7 LTS	1 Intel Core i7 8700k @3.7GHz, 64Gb swap, 64 Gb DDR4 RAM, 3Tb mechanical disk	Reviewer
$Ubuntu - base9$	Laptop	Ubuntu 20.04.1 LTS	1 Intel Core i5 8265u @1.6GHz, 16 DDR4 RAM, no swap, 34Gb virtual disk over SSD	Reviewer
$Windows - base1$	Laptop	Windows 10 Pro v. 20H2, Docker 20.10.5, Java 15.0.1	1 Intel Core i5-1035G1 @1.00GHz, 6 Gb DDR4 RAM, 240 Gb SSD	Authors

Table 6: The aggregate time required to run all the experiments included in Tables 8, 9 and 10 (that could be completed in less than 40 hours) for each testing platform, while reproducing most experiments from [1]. The testing platforms $Ubuntu - base3$, $Ubuntu - base4$, $Ubuntu - base5$, $Ubuntu - base6$, $Ubuntu - base9$ and $Windows - base1$ were limited in some experiments by the available main memory, thus exhibiting lower aggregate running times.

Run	Testing platform	Running time	Tested by
1	$Ubuntu - base1$	5,526 min ≈ 92.1 hrs	Authors
2	$Ubuntu - base2$	6,832 min ≈ 113.9 hrs	Authors
3	$Ubuntu - base3$	2,678 min ≈ 44.6 hrs	Authors
4	$Ubuntu - base4$	187 min ≈ 3.1 hrs	Authors
5	$Ubuntu - base5$	2,198 min ≈ 36.6 hrs	Authors
6	$Ubuntu - base6$	1,428 min ≈ 23.8 hrs	Reviewer
7	$Ubuntu - base7$	6,393 min ≈ 106.5 hrs	Reviewer
8	$Ubuntu - base8$	3,212 min ≈ 53.5 hrs	Reviewer
9	$Ubuntu - base9$	1,731 min ≈ 28.8 hrs	Reviewer
10	$Windows - base1$	1,743 min ≈ 29.1 hrs	Authors

runs, we can:

- Remove the existing Docker containers:
  ```
  sudo docker container ls -a | grep gmandi
  ```
  obtains the IDs of JedAI's containers, and
  ```
  sudo docker rm -f containerID
  ```
  removes a given container.

- Remove JedAI's Docker image:
  ```
  sudo docker rmi gmandi/jedai-webapp.
  ```
  [14]

- Finally, recover disk space for unused volumes:
  ```
  sudo docker volume prune.
  ```

Regarding the size of main memory (RAM), the vast majority of experiments require less than 2 Gb, as reported in Tables 8, 9 and 10, but 4 Gb are suggested to ensure Docker's stability, as explained above. However, the experiments with the two largest synthetic datasets, D_{1M} and D_{2M}, require up to 25 Gb, whereas the largest real dataset, D_{c8}, requires up to 105 Gb. The corresponding experiments cannot be run on most testing platforms that are equipped with 16 Gb RAM or less, namely $Ubuntu - base3$, $Ubuntu - base4$, $Ubuntu - base5$, $Ubuntu - base6$, $Ubuntu - base9$ and $Windows - base1$. Below, we report in detail the memory requirements of every experiment, highlighting the experiments that were not feasible, due to insufficient main memory in the testing platforms.

Finally, it is worth noting that the times reported in Table 6 merely correspond to the time taken by each system to run all experiments. Given that each experiment is carried out through the user interface of JedAI's Web application (i.e., they are not executed through a script), significant time is taken to manually navigate through all menus. Among them, the Entity Matching step requires additional time to transform the selected dataset into the textual representation that is suitable for assessing entity similarity (e.g., by tokenizing all attribute values into character n-grams). This time, which is negligible only for the smallest datasets, is not added to the overall running times in Table 6, which disregard completely the navigation time.

[14] Alternatively, run `sudo docker images` to obtain the IDs of the images, and then use `sudo docker rmi imageID` to remove them.

Table 7: Detailed instructions for reproducing all single-core experiments in [1] using the graphical user interface of JedAI's Web application.

Step	Reproduction instructions
	After launching JedAI's Docker image with the last command in Table 4:
(1)	Open a browser at `http://localhost:8080`. *If Docker runs on a server, replace '`localhost` with its URL. The host port 8080 was arbitrarily specified by the last command in Table 4 and can be changed at will. JedAI's homepage, depicted in Figure 2(a), shows up.*
(2)	Press the button '`New Workflow`'. *The window '`Choose New Workflow mode`' in Figure 2(b) pops up.*
(3)	Press the button '`Desktop Mode`'. *Because we are interested in the serial execution of JedAI's experiments.* *The Web page '`Select Workflow`' in Figure 2(c) shows up.*
(4)	Press the button '`Run tests`' at the bottom right corner. *The window '`Select Test to execute`' in Figure 2(d) shows up.* *The web application is already equipped with the parameters of all experiments.* *Thus, any experiment in [1] can be reproduced simply by selecting it from the menus of Figure 2(d).*
(5)	In '`Test Type`', select '`Performance Test`', '`Scalability Test`' or '`Budget-awareness Test`'. *The options for the rest of the selection criteria in the same window are activated.*
(6)	In '`ER Mode`', select '`Clean-Clean ER`' or '`Dirty ER`'. *For Scalability Tests, only '`Dirty ER`' is available.*
(7)	In '`Workflow Type`', select '`Best Schema-agnostic`', '`Default Schema-agnostic`' or '`Schema-aware`' pipelines. *For Scalability Tests, only the last two options are available.*
(8)	In '`Datasets`', select one among the available datasets in Tables 1, 2 and 3.
(9)	Press the button '`Confirm`'. *JedAI loads the selected pipeline with the parameter configuration corresponding to the selected dataset.* *One Web page for each step in the selected pipeline (see Figure 1) shows up.*
(10)	Press the button '`Next`' in the window of each pipeline step to proceed to the next one. *After going through all pipeline steps, the Web page '`Confirm Configurations`' in Figure 2(e) shows up.*
(11)	Press the button '`Confirm`'. *The Web page 'Workflow Execution' shows up.*
(12)	Press the button '`Execute Workflow`'. *The selected experiment is carried out. Upon completion, the respective performance is reported in the same window with respect to Recall, Precision, F-Measure and running time, as in Figure 2(f).*
(13)	In case of Budget-awareness Tests, press the button '*Show Plot*' at the bottom left corner. *A window similar to the one in Figure 2(g) shows up, depicting Progressive Recall along with the area under its curve.*
(14)	Press JedAI logo at the top of the window to return to the first screen and proceed with the next test.

2.5. Obtaining and compiling our source code

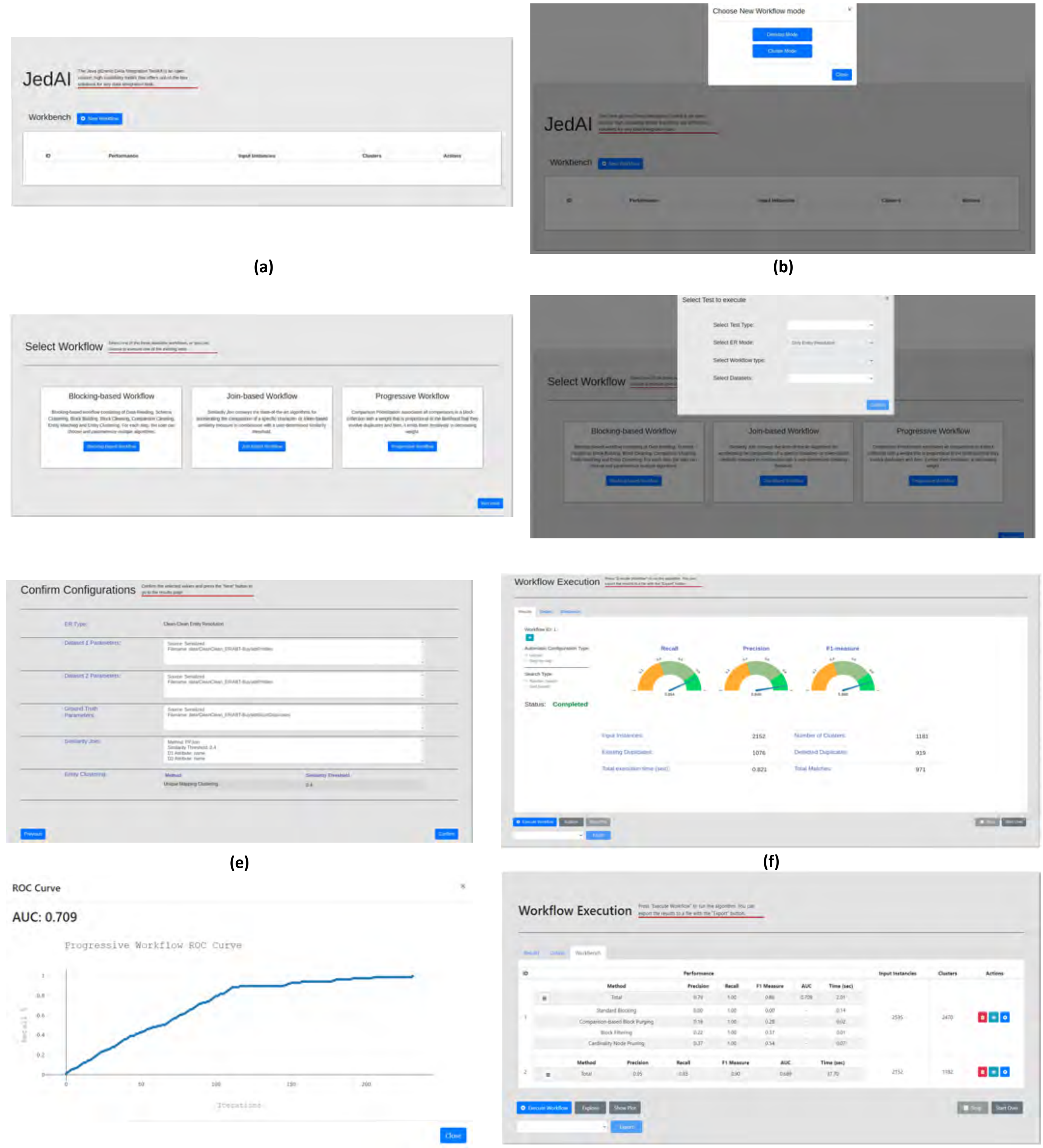
The source code for JedAI version 3.0, which is used in [1] and in the present experimental study, has been publicly released at: `https://github.com/scify/JedAIToolkit`. Any development kit and/or IDE for Java 8 or higher can be used for compiling it, but this is not necessary. JedAI's Docker image contains an executable jar file with the entire source code and its dependencies. When executed, it deploys JedAI's Web application, allowing users to reproduce all experiments by following the instructions below, in Section 2.6.

2.6. Running the experiments

Table 7 provides detailed guidelines for reproducing all experiments. In essence, the user merely needs to navigate through the windows of JedAI's user interface, which are illustrated in Figure 2. This means that minimal human intervention is required. For example, all datasets in Tables 1, 2 and 3 are already included in JedAI's Docker image; the one selected in Step 8 is automatically loaded after the Data Reading step, which follows Step 9 in all pipelines. Similarly, there is a separate window with all available methods for each pipeline step, but no particular action is required from the user: the method used in the chosen experiment is already marked as selected and its parameters are appropriately configured. The user simply needs to press '`Next`' in each step to proceed with the next one.

It is worth stressing at this point the wealth of information that is provided by the final window, called '`Workflow Execution`', after completing an experiment:

1. The button '`Explore`' presents the entity profiles that form each equivalence cluster.
2. The tab '`Details`' contains the output of each step in the latest pipeline so as understand its operation and contribution to the overall performance.
3. The tab '`Workbench`' summarizes the performance of all pipelines executed so far, as shown in Figure 2(h). This allows for juxtaposing the performance of different pipelines over the same dataset, even at the level of individual steps: pressing the button ≡ in the leftmost column displays a performance breakdown among all steps.



JedAI

Workbench

ID	Performance	Input Instances	Clusters	Actions

(a)

Choose New Workflow mode

(b)

Select Workflow

Blocking-based Workflow

Join-based Workflow

Progressive Workflow

Select Test to execute

Select Test Type:

Select ER Mode: Dirty Entity Resolution

Select Workflow type:

Select Datasets:

Confirm Configurations

ER Type:	Clean-Clean Entity Resolution
Dataset 1 Parameters:	Source: Serialized Filename: data/CleanClean_ER/ABT-Buy/abtProfiles
Dataset 2 Parameters:	Source: Serialized Filename: data/CleanClean_ER/ABT-Buy/abtProfiles
Ground Truth Parameters:	Source: Serialized Filename: data/CleanClean_ER/ABT-Buy/abtBuyIdDuplicates
Similarity Join:	Method: PPJoin Similarity Threshold: 0.4 D1 Attribute: name D2 Attribute: name
Entity Clustering:	Method: Unique Mapping Clustering Similarity Threshold: 0.4

(e)

Workflow Execution

Workflow ID: 1

Status: Completed

Input Instances:	2152	Number of Clusters:	1181
Existing Duplicates:	1076	Detected Duplicates:	919
Total execution time (sec):	0.821	Total Matches:	971

(f)

ROC Curve

AUC: 0.709

(g)

Workflow Execution

Press "Execute Workflow" to run the algorithm. You can export the results to a file with the "Export" button.

ID	Method	Precision	Recall	F1 Measure	AUC	Time (sec)	Input Instances	Clusters
1	Total	0.79	1.00	0.88	0.709	2.01	2535	2470
	Standard Blocking	0.00	1.00	0.07	-	0.14		
	Comparison-based Block Purging	0.18	1.00	0.28	-	0.02		
	Block Filtering	0.22	1.00	0.37	-	0.01		
	Cardinality Node Pruning	0.37	1.00	0.54	-	0.07		
2	Total	0.95	0.85	0.90	0.689	17.70	2152	1182

(h)

Figure 2: The screens of JedAI's Web application for reproducing all single core experiments in [1]: (a) The initial screen of JedAI's Web application. The button '`New Workflow`' should be pressed. (b) The second screen, which defines the execution mode. The button '`Desktop Mode`' should be pressed for the single-core experiments. (c) The third screen, which defines the type of the end-to-end pipeline. The button '`Run tests`' should be pressed to start the reproduction of the experiments. (d) The fourth screen, which defines the experimental settings we want to reproduce with respect to the type of experiments, the type of ER, the type of end-to-end pipeline and the dataset. (e) The '`Confirm Configuration`' screen that summarizes the experimental settings we have selected. (f) The final screen, '`Workflow Execution`', which presents the performance of the selected end-to-end pipeline. (g) The screen showing the area under the curve of Progressive Recall (AUC) in case of Budget-awareness Tests. (h) The benchmark screen summarizing the performance of all pipelines executed so far with respect to Precision, Recall, F-Measure, Run-time and Progressive Recall (AUC), in case of Budget-awareness Tests.

Table 8: The results of the Performance Tests over all real datasets across all testing platforms. For each pipeline, the effectiveness measures per dataset are common among all testing platforms. Only the running times differ among them. *IM* indicates a test that was not carried out due to insufficient memory. Note that Precision, Recall and F-Measure are rounded to three decimal places, memory requirements to two decimal places and running times to one decimal place.

	Clean-Clean ER								Dirty ER	
	Restaurants $\mathbf{D_{c1}}$	Abt Buy $\mathbf{D_{c2}}$	Amazon GP $\mathbf{D_{c3}}$	DBLP ACM $\mathbf{D_{c4}}$	Walmart Amazon $\mathbf{D_{c5}}$	DBLP Scholar $\mathbf{D_{c6}}$	IMDB DBPedia $\mathbf{D_{c7}}$	DBP-3.0rc DBP-3.4 $\mathbf{D_{c8}}$	$\mathbf{D_{cora}}$	$\mathbf{D_{cddb}}$
Precision	0.473	0.902	0.544	0.975	0.310	0.887	0.908	0.806	0.876	0.874
Recall	1.000	0.836	0.653	0.988	0.878	0.952	0.834	0.819	0.816	0.856
F-Measure	0.643	0.867	0.594	0.981	0.459	0.919	0.869	0.813	0.845	0.865
Memory (Gb)	0.02	0.04	0.19	0.09	0.32	0.75	0.99	105.00	0.17	1.45
$Ubuntu-base1$	1.1 sec	1.3 sec	12.0 sec	2.0 sec	8.3 sec	23.5 sec	91.0 sec	14.5 hrs	5.5 sec	65.0 sec
$Ubuntu-base2$	0.6 sec	1.3 sec	15.1 sec	1.3 sec	6.2 sec	28.9 sec	113.0 sec	22.1 hrs	2.7 sec	61.8 sec
$Ubuntu-base3$	0.5 sec	1.0 sec	11.2 sec	0.9 sec	4.4 sec	10.2 sec	68.0 sec	*IM*	1.8 sec	30.6 sec
$Ubuntu-base4$	0.2 sec	0.6 sec	8.6 sec	0.7 sec	3.5 sec	9.2 sec	53.4 sec	*IM*	1.8 sec	23.4 sec
$Ubuntu-base5$	0.3 sec	0.6 sec	8.2 sec	0.8 sec	3.7 sec	9.1 sec	48.5 sec	*IM*	1.3 sec	23.9 sec
$Ubuntu-base6$	0.1 sec	0.6 sec	8.3 sec	0.7 sec	3.0 sec	7.7 sec	51.9 sec	*IM*	1.3 sec	21.7 sec
$Ubuntu-base7$	0.2 sec	1.3 sec	15.9 sec	1.2 sec	5.3 sec	15.3 sec	98.4 sec	16.8 hrs	2.4 sec	49.0 sec
$Ubuntu-base8$	0.2 sec	0.6 sec	7.9 sec	0.6 sec	2.5 sec	6.5 sec	39.5 sec	*IM*	1.0 sec	18.8 sec
$Ubuntu-base9$	0.4 sec	1.1 sec	16.1 sec	1.0 sec	5.1 sec	11.8 sec	75.5 sec	*IM*	1.8 sec	30.8 sec
$Windows-base1$	0.3 sec	1.0 sec	8.7 sec	0.9 sec	4.2 sec	18.2 sec	98.6 sec	*IM*	1.7 sec	26.7 sec

(a) Default configuration of the budget- and schema-agnostic pipeline

	Restaurants $\mathbf{D_{c1}}$	Abt Buy $\mathbf{D_{c2}}$	Amazon GP $\mathbf{D_{c3}}$	DBLP ACM $\mathbf{D_{c4}}$	Walmart Amazon $\mathbf{D_{c5}}$	DBLP Scholar $\mathbf{D_{c6}}$	IMDB DBPedia $\mathbf{D_{c7}}$	DBP-3.0rc DBP-3.4 $\mathbf{D_{c8}}$	$\mathbf{D_{cora}}$	$\mathbf{D_{cddb}}$
Precision	0.788	0.946	0.576	0.993	0.590	0.946	0.905	0.841	0.912	0.869
Recall	1.000	0.854	0.646	0.992	0.753	0.949	0.876	0.821	0.819	0.886
F-Measure	0.881	0.898	0.609	0.992	0.662	0.948	0.890	0.831	0.863	0.877
Memory (Gb)	0.03	0.04	0.07	0.04	0.12	0.98	0.80	64.00	0.02	1.47
$Ubuntu-base1$	1.0 sec	1.1 sec	4.5 sec	1.3 sec	5.3 sec	30.0 sec	46.0 sec	12.7 hrs	0.9 sec	65.7 sec
$Ubuntu-base2$	0.7 sec	1.1 sec	6.1 sec	0.8 sec	12.9 sec	45.1 sec	49.5 sec	21.9 hrs	0.8 sec	70.0 sec
$Ubuntu-base3$	0.5 sec	0.8 sec	5.0 sec	0.6 sec	2.4 sec	16.4 sec	29.0 sec	*IM*	0.6 sec	32.4 sec
$Ubuntu-base4$	0.5 sec	0.7 sec	4.1 sec	0.4 sec	1.8 sec	12.6 sec	23.9 sec	*IM*	0.4 sec	25.6 sec
$Ubuntu-base5$	0.4 sec	0.6 sec	3.2 sec	0.7 sec	2.0 sec	12.2 sec	24.5 sec	*IM*	0.5 sec	30.8 sec
$Ubuntu-base6$	0.1 sec	0.5 sec	3.1 sec	0.5 sec	1.7 sec	13.1 sec	21.8 sec	*IM*	0.3 sec	23.3 sec
$Ubuntu-base7$	0.1 sec	0.6 sec	7.3 sec	0.8 sec	2.8 sec	23.9 sec	41.7 sec	16.5 hrs	0.7 sec	51.3 sec
$Ubuntu-base8$	0.1 sec	0.4 sec	3.5 sec	0.4 sec	1.2 sec	11.2 sec	18.6 sec	12.9 hrs	0.3 sec	24.5 sec
$Ubuntu-base9$	0.2 sec	0.4 sec	5.9 sec	0.6 sec	2.2 sec	16.8 sec	34.8 sec	*IM*	0.3 sec	34.9 sec
$Windows-base1$	0.2 sec	0.6 sec	4.8 sec	0.5 sec	2.3 sec	18.7 sec	28.9 sec	*IM*	0.5 sec	32.5 sec

(b) Best configuration of the budget- and schema-agnostic pipeline

	Restaurants $\mathbf{D_{c1}}$	Abt Buy $\mathbf{D_{c2}}$	Amazon GP $\mathbf{D_{c3}}$	DBLP ACM $\mathbf{D_{c4}}$	Walmart Amazon $\mathbf{D_{c5}}$	DBLP Scholar $\mathbf{D_{c6}}$	IMDB DBPedia $\mathbf{D_{c7}}$	DBP-3.0rc DBP-3.4 $\mathbf{D_{c8}}$	$\mathbf{D_{cora}}$	$\mathbf{D_{cddb}}$
Precision	0.755	0.884	0.663	0.978	0.829	0.953	0.931	0.833	0.751	0.278
Recall	0.933	0.438	0.423	0.932	0.552	0.775	0.499	0.370	0.859	0.719
F-Measure	0.834	0.585	0.517	0.954	0.663	0.855	0.649	0.512	0.802	0.401
Memory (Gb)	0.01	0.02	0.02	0.02	0.06	0.11	0.42	30.00	0.02	0.06
$Ubuntu-base1$	0.2 sec	0.4 sec	0.5 sec	0.6 sec	0.5 sec	14.0 sec	7.7 sec	15.2 min	0.3 sec	0.6 sec
$Ubuntu-base2$	0.2 sec	0.2 sec	0.2 sec	0.5 sec	0.2 sec	13.8 sec	6.9 sec	12.4 min	0.3 sec	0.3 sec
$Ubuntu-base3$	0.2 sec	0.3 sec	0.3 sec	0.2 sec	0.2 sec	10.6 sec	5.2 sec	*IM*	0.2 sec	0.3 sec
$Ubuntu-base4$	0.1 sec	0.1 sec	0.1 sec	0.1 sec	0.1 sec	10.2 sec	3.5 sec	*IM*	0.2 sec	0.3 sec
$Ubuntu-base5$	0.1 sec	0.1 sec	0.2 sec	0.2 sec	0.3 sec	7.4 sec	3.4 sec	*IM*	0.2 sec	0.3 sec
$Ubuntu-base6$	0.1 sec	0.1 sec	0.2 sec	0.3 sec	0.1 sec	6.3 sec	3.3 sec	*IM*	0.1 sec	0.3 sec
$Ubuntu-base7$	0.1 sec	0.2 sec	0.2 sec	0.2 sec	0.2 sec	14.2 sec	7.7 sec	11.0 min	0.1 sec	0.2 sec
$Ubuntu-base8$	0.1 sec	0.1 sec	0.1 sec	0.1 sec	0.1 sec	5.9 sec	3.2 sec	5.2 min	0.1 sec	0.1 sec
$Ubuntu-base9$	0.1 sec	0.1 sec	0.2 sec	0.2 sec	0.1 sec	16.5 sec	5.6 sec	*IM*	0.1 sec	0.2 sec
$Windows-base1$	0.1 sec	0.1 sec	0.2 sec	0.2 sec	0.1 sec	16.5 sec	5.6 sec	*IM*	0.1 sec	0.2 sec

(c) Best configuration of the budget-agnostic, schema-based pipeline

The outcomes of the Performance, the Scalability and the Budget-awareness tests over all testing platforms are reported in Tables 8, 9 and 10, respectively. In all cases, the effectiveness measures are common among all platforms, with the only differences corresponding to the running times. Compared to the experiments reported in [1], the effectiveness results of Budget-awareness tests are practically identical in most cases. The only significant exceptions pertain to the best schema-agnostic pipeline over D_{c2}, D_{c3} and D_{cddb}, whose F-Measure has now changed from 0.900, 0.607 and 0.872 to 0.898, 0.609 and 0.877, respectively, after some bug fixes. The F-Measure of the default schema-agnostic pipeline over D_{c3} has also increased from 0.586 to 0.594. The effectiveness results of the Scalability and the Budget-awareness tests are also identical

Table 9: The results of the Scalability Tests over the seven synthetic datasets across all testing platforms. For each pipeline, the effectiveness measures per dataset are common among all testing platforms. Only the running times differ among them. IM indicates a test that was not carried out due to insufficient memory. Note that Precision, Recall and F-Measure are rounded to three decimal places, memory requirements to two decimal places and running times to one decimal place.

	D_{10K}	**D_{50K}**	**D_{100K}**	**D_{200K}**	**D_{300K}**	**D_{1M}**	**D_{2M}**
Precision	0.948	0.899	0.887	0.844	0.866	0.868	0.836
Recall	0.994	0.989	0.983	0.978	0.973	0.960	0.954
F-Measure	0.970	0.942	0.933	0.906	0.916	0.911	0.891
Memory (Gb)	0.12	0.80	3.10	6.20	7.20	15.00	25.00
$Ubuntu-base1$	1.8 sec	12.8 sec	35.1 sec	120.2 sec	193.1 sec	32.3 min	147.1 min
$Ubuntu-base2$	1.6 sec	11.4 sec	37.3 sec	130.8 sec	199.3 sec	33.4 min	145.4 min
$Ubuntu-base3$	1.4 sec	5.2 sec	19.8 sec	51.9 sec	141.9 sec	22.1 min	IM
$Ubuntu-base4$	0.9 sec	4.3 sec	10.7 sec	54.3 sec	IM	IM	IM
$Ubuntu-base5$	1.0 sec	3.7 sec	11.5 sec	37.8 sec	77.0 sec	16.9 min	IM
$Ubuntu-base6$	0.7 sec	4.2 sec	10.8 sec	36.7 sec	114.6 sec	–	–
$Ubuntu-base7$	0.8 sec	6.3 sec	19.9 sec	63.4 sec	148.0 sec	24.2 min	93.5 min
$Ubuntu-base8$	0.5 sec	3.6 sec	9.0 sec	27.1 sec	71.3 sec	12.9 min	51.4 min
$Ubuntu-base9$	1.5 sec	8.2 sec	15.7 sec	46.9 sec	118.4 sec	22.6 min	IM
$Windows-base1$	1.4 sec	5.0 sec	10.8 sec	47.2 sec	232.5 sec	IM	IM

(a) Default configuration of the budget- and schema-agnostic pipeline

	D_{10K}	**D_{50K}**	**D_{100K}**	**D_{200K}**	**D_{300K}**	**D_{1M}**	**D_{2M}**
Precision	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Recall	0.593	0.598	0.602	0.600	0.602	0.603	0.602
F-Measure	0.744	0.749	0.752	0.750	0.751	0.752	0.752
Memory (Gb)	0.03	0.10	0.30	1.15	1.75	11.00	16.00
$Ubuntu-base1$	7.0 sec	137.2 sec	695.3 sec	55.6 min	140.3 min	17.8 hrs	>40 hrs
$Ubuntu-base2$	5.3 sec	120.3 sec	534.8 sec	49.6 min	96.9 min	19.3 hrs	>40 hrs
$Ubuntu-base3$	4.0 sec	89.4 sec	367.8 sec	26.3 min	69.4 min	13.0 hrs	>40 hrs
$Ubuntu-base4$	3.9 sec	74.6 sec	316.5 sec	24.0 min	48.4 min	IM	IM
$Ubuntu-base5$	3.6 sec	67.9 sec	298.2 sec	21.3 min	55.9 min	10.3 hrs	>40 hrs
$Ubuntu-base6$	3.8 sec	78.6 sec	341.5 sec	23.6 min	57.1 min	–	>40 hrs
$Ubuntu-base7$	7.9 sec	172.2 sec	704.1 sec	49.5 min	111.9 min	19.8 hrs	>40 hrs
$Ubuntu-base8$	3.1 sec	140.1 sec	375.2 sec	22.3 min	49.7 min	10.2 hrs	39.8 hrs
$Ubuntu-base9$	5.3 sec	96.8 sec	376.4 sec	28.7 min	64.5 min	10.8 hrs	>40 hrs
$Windows-base1$	4.3 sec	87.3 sec	376.7 sec	26.7 min	56.8 min	IM	IM

(b) Best configuration of the budget-agnostic, schema-based pipeline

with those reported in [1]; only their format has changed from diagrams to tables. In all cases, the running times in [1] are reproduced here, corresponding to $Ubuntu-base1$ in Tables 8 and 9 and to $Ubuntu-base1'$ in Table 10.

Finally, it is worth stressing that there is a delay when pressing the '`Next`' button in the window '`Entity Matching`' of the schema-agnostic pipelines. For small datasets, the delay is hardly observable, but it increases for larger datasets, raising up to few minutes for D_{1M}, D_{2M} and D_{c8}. This delay is caused by a process that converts all entity profiles into the representation model of the selected Entity Matching method. This is included in the running times of $Ubuntu-base1$, where all experiments were run through script files, but is not considered by any other testing platform, where all experiments were executed through JedAI's user interface. This is one of the reasons for the significantly higher running times of $Ubuntu-base1$ even in comparison to similar testing platforms, like $Ubuntu-base2$.

3. Reconfiguring and Extending our Experiments

3.1. Evaluating different experimental setups

To test the robustness of our experimental study, the configuration of a particular experiment can be adjusted in two different ways as follows:

1. by enriching or modifying the methods of at least one pipeline step, and/or
2. by altering the value of at least one parameter in one of the selected methods.

This is possible by repeating the procedure in Table 7 up to the first window of Step 10, namely '`Data Reading`'. Subsequently, in the separate window of each step, the pre-selected options can be modified as described below, in Sections 3.1.1, 3.1.2 and 3.1.3, for each type of experiments.

Note that every method in every pipeline step is associated with three configuration approaches: '`Default`', '`Automatic`', '`Manual`'. The '`Default`' configuration is already widely used in the experimental analysis of [1]. The '`Automatic`' configuration applies grid or random search over numerous iterations

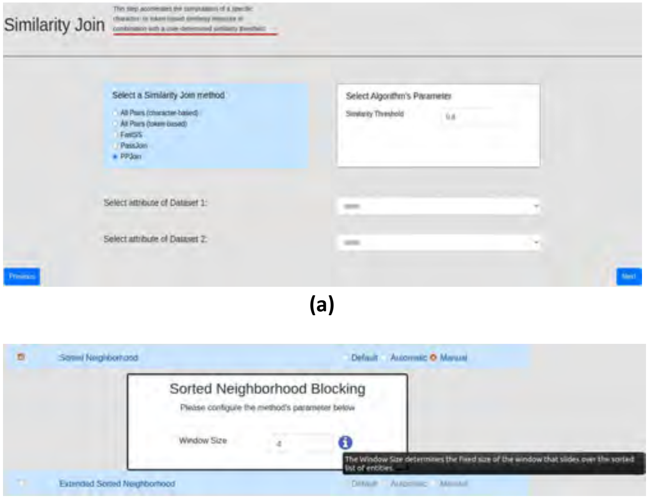
Table 10: The results of the Budget-awareness Tests over all real datasets across all testing platforms. For each pipeline, effectiveness is measured through the area under the curve of Progressive Recall, which is common among all testing platforms in each dataset only for the budget-aware pipeline. Its budget-agnostic counterpart arranges all pairwise comparisons in a random order, thus yielding a Progressive Recall that differs in each run and, thus, among the testing platforms. Note that Precision, Recall and F-Measure are rounded to three decimal places, memory requirements to two decimal places and running times to one decimal place. Note also that D_{c8} is omitted, as in [1], due to the excessively large running time and the very high memory requirements of the corresponding experiment.

	Clean-Clean ER							Dirty ER	
	Restaurants $\mathbf{D_{c1}}$	Abt Buy $\mathbf{D_{c2}}$	Amazon GP $\mathbf{D_{c3}}$	DBLP ACM $\mathbf{D_{c4}}$	Walmart Amazon $\mathbf{D_{c5}}$	DBLP Scholar $\mathbf{D_{c6}}$	IMDB DBPedia $\mathbf{D_{c7}}$	$\mathbf{D_{cora}}$	$\mathbf{D_{cddb}}$
Progressive Recall	0.709	0.689	0.573	0.866	0.635	0.930	0.616	0.416	0.585
Memory (Gb)	0.06	0.08	0.30	0.16	0.65	4.00	6.00	0.30	3.50
$Ubuntu-base1'$	0.5 sec	13.7 sec	1.8 min	32.9 sec	5.2 min	46.3 min	18.4 hrs	16.9 sec	79.3 sec
$Ubuntu-base2$	0.6 sec	19.0 sec	3.4 min	49.8 sec	7.8 min	68.2 min	20.1 hrs	15.8 sec	96.5 sec
$Ubuntu-base3$	0.4 sec	14.6 sec	2.2 min	48.5 sec	7.4 min	54.1 min	12.7 hrs	12.2 sec	73.5 sec
$Ubuntu-base4$	0.3 sec	12.5 sec	1.5 min	35.7 sec	6.1 min	40.7 min	IM	9.9 sec	55.6 sec
$Ubuntu-base5$	0.6 sec	12.2 sec	2.0 min	31.5 sec	5.1 min	37.2 min	10.8 hrs	9.4 sec	49.9 sec
$Ubuntu-base6$	0.3 sec	11.9 sec	1.4 min	33.5 sec	4.9 min	34.5 min	9.6 hrs	10.1 sec	53.8 sec
$Ubuntu-base7$	0.4 sec	20.9 sec	2.7 min	63.1 sec	9.4 min	64.7 min	22.0 hrs	19.6 sec	107.8 sec
$Ubuntu-base8$	0.3 sec	10.2 sec	1.2 min	30.0 sec	4.2 min	28.7 min	9.5 hrs	8.8 sec	45.8 sec
$Ubuntu-base9$	0.8 sec	16.6 sec	2.2 min	48.8 sec	6.9 min	38.8 min	13.9 hrs	14.3 sec	66.9 sec
$Windows-base1$	0.5 sec	14.9 sec	1.9 min	20.4 sec	8.3 min	52.4 min	11.3 hrs	15.3 sec	80.0 sec

(a) Budget-aware, schema-agnostic pipeline

Memory (Gb)	0.09	0.20	0.20	0.35	0.65	4.00	6.00	0.30	3.50
Progressive Recall	0.489	0.418	0.337	0.491	0.386	0.478	0.435	0.661	0.451
$Ubuntu-base1'$	1.6 sec	19.2 sec	2.4 min	34.4 sec	11.6 min	51.0 min	20.8 hrs	30.4 sec	13.5 min
Progressive Recall	0.491	0.400	0.341	0.489	0.383	0.479	0.436	0.665	0.446
$Ubuntu-base2$	0.5 sec	15.5 sec	2.6 min	40.4 sec	13.6 min	61.5 min	21.8 hrs	18.5 sec	13.6 min
Progressive Recall	0.481	0.403	0.328	0.488	0.397	0.474	0.437	0.659	0.466
$Ubuntu-base3$	0.4 sec	12.4 sec	1.9 min	31.4 sec	12.6 min	49.5 min	14.5 hrs	15.4 sec	11.9 min
Progressive Recall	0.521	0.405	0.335	0.495	0.371	0.488	IM	0.668	0.464
$Ubuntu-base4$	0.5 sec	10.2 sec	1.6 min	25.9 sec	9.7 min	36.7 min	IM	10.4 sec	7.8 min
Progressive Recall	0.487	0.402	0.322	0.488	0.383	0.475	0.435	0.678	0.482
$Ubuntu-base5$	0.5 sec	11.1 sec	1.4 min	25.8 sec	8.8 min	37.3 min	12.2 hrs	10.8 sec	8.0 min
Progressive Recall	0.483	0.399	0.326	0.498	0.374	0.476	0.436	0.666	0.460
$Ubuntu-base6$	0.2 sec	10.7 sec	1.6 min	28.9 sec	9.0 min	35.6 min	11.1 hrs	10.6 sec	8.0 min
Progressive Recall	0.457	0.406	0.344	0.501	0.381	0.463	0.436	0.661	0.510
$Ubuntu-base7$	0.4 sec	21.9 sec	2.9 min	57.4 sec	19.0 min	72.7 min	23.3 hrs	21.8 sec	15.9 min
Progressive Recall	0.528	0.416	0.319	0.502	0.376	0.464	0.435	0.668	0.488
$Ubuntu-base8$	0.2 sec	9.0 sec	1.4 min	25.1 sec	7.5 min	29.8 min	10.8 hrs	9.9 sec	6.9 min
Progressive Recall	0.453	0.414	0.330	0.490	0.379	0.476	0.436	0.669	0.463
$Ubuntu-base9$	0.3 sec	14.4 sec	2.4 min	39.7 sec	12.4 min	46.4 min	14.5 hrs	18.0 sec	11.9 min
Progressive Recall	0.520	0.396	0.334	0.498	0.381	0.469	0.434	0.659	0.453
$Windows-base1$	0.7 sec	14.5 sec	2.2 min	37.9 sec	13.8 min	50.8 min	13.5 hrs	15.1 sec	24.3 min

(b) Budget- and schema-agnostic pipeline


Similarity Join

This step accelerates the computation of a specific character- or token-based similarity measure in combination with a user-determined similarity threshold

Select a Similarity Join method
- ○ All Pairs (character-based)
- ○ All Pairs (token-based)
- ○ FastSS
- ○ PassJoin
- ● PPJoin

Select Algorithm's Parameter

Similarity Threshold: 0.8

Select attribute of Dataset 1:

Select attribute of Dataset 2:

(a)

Sorted Neighborhood ○ Default ○ Automatic ● Manual

Sorted Neighborhood Blocking

Please configure the method's parameter below

Window Size: 4

The Window Size determines the fixed size of the window that slides over the sorted list of entities.

Extended Sorted Neighborhood ○ Default ○ Automatic ○ Manual

(b)

Figure 3: (a) The screen showing the configuration for a particular pipeline step. (b) The tooltip that explains the role of a particular parameter during the manual configuration of a method.

so as to identify the settings that maximize F-Measure. The random search involves 100 iterations, while the grid search might yield an exponential number of iterations in case multiple parameters are simultaneously fine-tuned. As both options might lead to long running times, the preferred approach is the '`Manual`' configuration. After selecting it, JedAI presents all parameters of the current pipeline along with their default values, as in Figure 3(a). The user can alter these values at will and store them by pressing '`Next`' to proceed to the next window.

Note also that every method in JedAI implements the `IDocumentation` interface, which conveys all necessary information for its manual configuration. When configuring a specific parameter, the information image ⓘ is shown. When leaving the mouse cursor over it, a tooltip appears that describes the role of this parameter. An example is shown in Figure 3(b).

Below, we explain the restrictions that apply to each pipeline step with respect to the methods that can be selected.

3.1.1. Schema-Agnostic End-to-End Pipeline

As explained above, this pipeline involves six steps:

1. Schema Clustering. At most one method can be selected, but this step is not used in the considered experiments.
2. Block Building. One or more of the nine available methods can be selected. All experiments exclusively employ Token Blocking, which is a parameter-free approach.
3. Block Cleaning. Any combination of the three available methods is possible. All experiments apply Comparison-based Block Purging and Block Filtering with their default parameter values.
4. Comparison Cleaning. At most one of the nine available methods can be selected. In our experiments, we exclusively use Cardinality Node Pruning (CNP) with its default configuration. All methods are configured simply by selecting one of the six weighting schemes.
5. Entity Matching. One of the two available methods can be applied. All experiments employ the Profile Matcher. Both methods are configured by selecting a similarity measure and a compatible representation model, which transforms the set of textual attribute values in each entity profile into a suitable format. These two parameters give rise to numerous configurations.
6. Entity Clustering. At most one method can be selected in this step. There are three methods available for Clean-Clean ER, but all experiments employ the Unique Mapping Clustering approach. For Dirty ER, there are seven methods for Dirty ER, but all experiments use the Connected Components Clustering. All methods are configured by setting their similarity threshold, below which all pairwise comparisons are discarded.

3.1.2. Schema-Based End-to-End Pipeline

This pipeline consists of two steps:

1. The Similarity Join step offers five similarity join algorithms. Among them, PPJoin is used in all experiments. All methods are configured by setting their similarity threshold along with the attribute(s), to which they are applied.
2. The Entity Clustering step is the same as the schema-agnostic pipeline. In most cases, it uses the same similarity threshold as the previous step.

3.1.3. Budget-Aware Schema-Agnostic Pipeline

This pipeline differs from its budget-agnostic counterpart (see Section 3.1.1) only in the Prioritization step that intervenes between Comparison Cleaning and Entity Matching. There are different options for this step, depending on the preceding pipeline steps: if no Block Building method is employed, two methods are available, otherwise one of five different methods can be used. The latter approach was used in all Budget-awareness tests. In both cases, at most one approach can be selected and it is configured by setting its budget (i.e., number of executed comparisons) and the weighting scheme that lies at its core.

Note that for all tests, the next configuration experiment is performed by pressing the '`Start Over`' button at the bottom right corner of Figure 2(f) to return to the Data Reading step of the current experiment.

3.2. Extending our experiments

Our experimental study can be extended in two ways. First, by adding new datasets through the 'Data Reading' step. The window of this step allows users to select any dataset in any of the supported formats (CSV, relational DB, XML or RDF) that is stored either locally or is available through a server with a public URL. Note that each dataset should be accompanied by the golden standard comprising all duplicates.

Second, it is possible to extent our experimental analysis with new methods in any of the considered pipeline steps by leveraging JedAI's extensible architecture. The only requirement is that every new method is available through a Java class that implements the interface of the corresponding pipeline step - as explained in [1], every step is associated with a simple Java interface that determines its input and output. In this way, new methods can be seamlessly integrated into JedAI's code and be treated like the already available methods. Ideally, the new methods should also implement the `IDocumentation` interface, which exposes the following functions that return textual descriptions about the core characteristics of an algorithm:

- `getMethodName()` returns the name of the method.
- `getParameterName(int parameterId)` returns the name of a particular configuration parameter.
- `getParameterDescription(int parameterId)` returns a short description for a particular configuration parameter.
- `getMethodParameters()` returns a description for all configuration parameters of the method, using the above functions.
- `getMethodInfo()` returns a short description of the method's internal functionality.
- `getMethodConfiguration()` returns the parameter configuration of the current instance of a method. It is called by logger.
- `getParameterConfiguration()` returns a `JsonArray` object with a `JsonObject` for every configuration parameter that comprises the following information: the class of the parameter (e.g., `java.lang.Integer`), its name, determined by the function `getParameterName`, its default, minimum and maximum values along with the step one, and its description, determined by the function `getParameterDescription`. This information is used for the manual configuration through JedAI's interface.

This documentation, which is also leveraged by JedAI's user interface, ensures that new methods can be easily employed by users other than their creators. For more details on extending JedAI please refer to [1].

4. Conclusions

We have presented an analytical user guide for JedAI's Web application, which is available through a Docker image. Our instructions allow a user with limited or no familiarity with Entity Resolution to repeat all single-core experiments in [1] so as to evaluate the relative performance of the main end-to-end pipelines. Our instructions also facilitate the reconfiguration of these experiments, by constructing and evaluating pipelines of arbitrary complexity.

All these experiments involve learning-free methods. In the future, we plan to extend JedAI with learning-based methods, paying particular attention to the integration of Deep Learning technologies.

5. Revision Comments

This reproducibility manuscript is a valuable complement to the parent paper [1], where the last release of JedAI software was presented. JedAI includes a web-based user interface and a complete library of techniques needed to create end-to-end Entity Resolution (ER) pipelines. The authors compared different ER techniques by considering three different dimensions that included: (a) Schema-awareness, (b) Budget-awareness, and (c) Execution mode. The wide set of experiments provided included the evaluation of 17 datasets and considered the performance, scalability, and budget awareness of the ER pipelines. This paper provides the actual configuration used for those ER pipelines, and gives some ideas regarding how they can be personalized. Furthermore, some guidelines showing how JedAI can be extended are also devised.

Apart from creating a permanent repository in Mendeley with the necessary software and datasets, the authors provide a Docker-based system to reproduce those experiments. Using the web-based interface of JedAI, any researcher can easily use the default configuration parameters provided for each experiment, execute it, and finally see the results of that execution. Besides, JedAI also allows to configure and personalize those default parameters, as well as the addition of new methods for the comparison with existing methods, adding extra value to the current work.

While reviewing this manuscript, a few issues around reproducibility were brought into the discussion, which show how difficult it can be to provide a complete reproducible framework. We dealt with some experiments where the provided default parameters were wrong, which led to unexpected results. Another minor issue was related to yielding slightly different values than those reported in the parent paper or figures showing the results in a rather different shape. We also found some mismatches concerning the memory requirements needed to run some experiments, which would not end or report higher execution times than expected. All those issues were successfully fixed during the revision process. The authors satisfactorily took all our comments into account and improved their software library and web application. Finally, the JedAI reproduction framework does not provide a mechanism to automatically run all the experiments, gather all the results, and create the same tables and figures of the parent paper, which would be extremely interesting to reproduce the original work easily. However, the workflow included in JedAI still allows any researcher to effortlessly reproduce each experiment. The process consists of choosing the experiment to perform, going through the screens that display the default parameters, starting the execution, waiting for it to complete, and finally gathering the results.

We would like to thank the authors for their considerable effort to provide a valuable software library to the research community. This library allows new researchers to understand and reproduce state-of-the-art experiments with minimal effort and guarantees long-term software support, following a sequence of precise and straightforward instructions.

Acknowledgements. This work was partially funded by the

EU H2020 project ExtremeEarth (Grant No. 825258).

References

[1] G. Papadakis, G. Mandilaras, L. Gagliardelli, G. Simonini, E. Thanos, G. Giannakopoulos, S. Bergamaschi, T. Palpanas, M. Koubarakis, Three-dimensional entity resolution with jedai, Inf. Syst. 93 (2020) 101565.

[2] G. Papadakis, E. Ioannou, E. Thanos, T. Palpanas, The four generations of entity resolution, Synthesis Lectures on Data Management.

[3] P. Christen, Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Data-Centric Systems and Applications, Springer, 2012.

[4] X. L. Dong, D. Srivastava, Big Data Integration, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2015.

[5] V. Christophides, V. Efthymiou, K. Stefanidis, Entity Resolution in the Web of Data, Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers, 2015.

[6] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, Duplicate record detection: A survey, IEEE Trans. Knowl. Data Eng. 19 (1) (2007) 1–16.

[7] G. Papadakis, D. Skoutas, E. Thanos, T. Palpanas, Blocking and filtering techniques for entity resolution: A survey, ACM Computing Surveys 53 (2) (2020) 31:1–31:42.

[8] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, K. Stefanidis, An overview of end-to-end entity resolution for big data, ACM Computing Surveys 53 (6).

[9] L. Getoor, A. Machanavajjhala, Entity resolution: Theory, practice & open challenges, PVLDB 5 (12) (2012) 2018–2019.

[10] K. Stefanidis, V. Efthymiou, M. Herschel, V. Christophides, Entity resolution in the web of data, in: WWW, 2014, pp. 203–204.

[11] G. Papadakis, T. Palpanas, Web-scale, schema-agnostic, end-to-end entity resolution, in: The Web Conference (WWW), Lyon, France, 2018.

[12] G. Papadakis, E. Ioannou, T. Palpanas, Entity resolution: Past, present and yet-to-come, in: EDBT, 2020, pp. 647–650.

[13] G. Papadakis, Entity resolution benchmark dataset, `https://data.mendeley.com/datasets/4whpm32y47` (2020).

[14] J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. T. dos Santos, Results of the ontology alignment evaluation initiative 2010, in: Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), 2010.

[15] Ontology alignment evaluation initiative, `http://oaei.ontologymatching.org/2010` (2010).

[16] H. Köpcke, A. Thor, E. Rahm, Evaluation of entity resolution approaches on real-world match problems, Proc. VLDB Endow. 3 (1) (2010) 484–493.

[17] Benchmark datasets for entity resolution, `https://dbs.uni-leipzig.de/research/projects/object_matching/benchmark_datasets_for_entity_resolution` (2010).

[18] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. W. Shavlik, X. Zhu, Corleone: hands-off crowdsourcing for entity matching, in: SIGMOD, 2014, pp. 601–612.

[19] S. Das, A. Doan, P. S. G. C., C. Gokhale, P. Konda, Y. Govind, D. Paulsen, The magellan data repository, `https://sites.google.com/site/anhaidgroup/projects/data`.

[20] G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser, Efficient entity resolution for large heterogeneous information spaces, in: WSDM, 2011, pp. 535–544.

[21] G. Papadakis, Blocking framework, `https://sourceforge.net/projects/erframework/`.

[22] A. McCallum, K. Nigam, L. H. Ungar, Efficient clustering of high-dimensional data sets with application to reference matching, in: ACM SIGKDD, 2000, pp. 169–178.

[23] Repeatability datasets, `https://hpi.de/naumann/projects/repeatability/datasets.html`.

[24] U. Draisbach, F. Naumann, A comparison and generalization of blocking and windowing algorithms for duplicate detection, in: Proceedings of the International Workshop on Quality in Databases (QDB), 2009, pp. 51–56.

[25] B. Kenig, A. Gal, Mfiblocks: An effective blocking algorithm for entity resolution, Inf. Syst. 38 (6) (2013) 908–926.

[26] P. Konda, S. Das, P. S. G. C., A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. F. Naughton, S. Prasad, G. Krishnan, R. Deep, V. Raghavendra, Magellan: Toward building entity matching management systems, Proc. VLDB Endow. 9 (12) (2016) 1197–1208.

[27] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, Deep learning for entity matching: A design space exploration, in: SIGMOD, 2018, pp. 19–34.

Chapter 11

IberLEF-SEPLN Workshops articles

Key Phrases Annotation in Medical Documents: MEDDOCAN 2019 Anonymization Task[⋆]

Alicia Lara-Clares[1] and Ana Garcia-Serrano[2]

[1] Universidad Nacional de Educación a Distancia (UNED), Spain
alara@lsi.uned.es

[2] Universidad Nacional de Educación a Distancia (UNED), Spain
agarcia@lsi.uned.es

Abstract. There is a vast amount of digitized information about medical records, treatments and diseases, that used to be in an unstructured or semi-structured format. In order to take advantage of all the potential data that can be extracted from this information, it is necessary to deploy systems capable of converting it into annotated and structured information. In the context of the MEDDOCAN shared task of IberLEF2019, we use a Few-Shot Learning approach for Named Entity Recognition (NER) in medical documents to identify and classify key phrases in a document. The architecture of the system is an hybrid Bi-LSTM and CNN model with four input layers that can recognize multi-word entities using the BIO encoding format for the labels.

Keywords: NER · Bi-LSTM · CNN · wikipedia2vec

1 Introduction

Nowadays, there is a vast amount of digitized information about medical records, treatments and diseases, but is not completely annotated yet so there is unstructured or semi-structured information. In order to take advantage of all the potential data that can be extracted from this information, it is necessary to deploy systems capable of processing and converting it into structured information.

Recently, neural networks are shown to be especially successful in complex NLP tasks [14]. For example, G. Fabregat et al. [2] use a deep learning model for disabilities and diseases recognition using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Also the work with word embedding is one hot topic in this area, for example to simplify drug package leaflets written in Spanish [10] or to define reproducible experiments and replication datasets [6].

The MEDDOCAN task presented at the Iberian Languages Evaluation Forum (Iberlef) 2019 [8] has the objective of anonymize medical documents in Spanish. The task is structured in two sub-tasks: NER offset and entity type classification and sensitive token detection.

[⋆] Supported by UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018)

Our work is based in the Few-shot Learning Model to learn high level features from datasets [3, 12]. We propose a hybrid Bi-LSTM CNN model by extending the model presented in [4] adding a Part-of-Speech (POS) tagging layer, that is, information about multi-word entities. Moreover, in this work, we use wikipedia2vec [13], a pre-trained word embedding model from Wikipedia. This approach to automatically extract and classify keywords is detailed at [5]. The code is available on Github [3].

The rest of the paper is organized as follows. In section 2, we describe the architecture of the system. Section 3 describes the evaluation process and results obtained. Finally, section 4 outlines the conclusions and future works.

2 System description

The system process is organized into (1) a pre-process of the data to be the input of the neural network, (2) its processing with the neural network and (3) the post-process of the output data format.

All documents are pre-processed following the next steps. First, sentences are splitted and tokenized using the Stanford CoreNLP natural language processing toolkit [7], ignoring all non-alphanumeric symbols. Then, each token is annotated using the BIO scheme, to preserve the multi-word entities. After that, we get the POS tag of each token (using the Stanford Core-NLP POS tagger).

The output of the system (annotated as shown in the Table 1: concept, POS tags and BIO-label) is converted into the BRAT format [11]. The BRAT format store all the information of the initial data together with the labels of each category and the positions of the tokens in the text.

Table 1. Structure of processed data in this work

Concept	POS tag	BIO label
Edad	PROPN	O
70	NUM	B-EDAD_SUJETO_ASISTENCIA
anyos	NOUN	I-EDAD_SUJETO_ASISTENCIA
Sexo	NOUN	O

The network architecture of this work is detailed also in [5]. It has four input layers, named as character level, word level, casing input and POS tag level, as can be seen in Figure 1.

- The character level starts with a character embedding that maps a vocabulary of 120 possible characters to an embedding initialized randomly. The maximum number of character per word is 52. It has a dropout layer (with drop rate 0.5) used to avoid the risk of overfitting. Finally, it has a convolutional layer to process the 1-dimension character layer.

[3] https://github.com/alicialara/lsi2_uned_at_MEDDOCAN2019

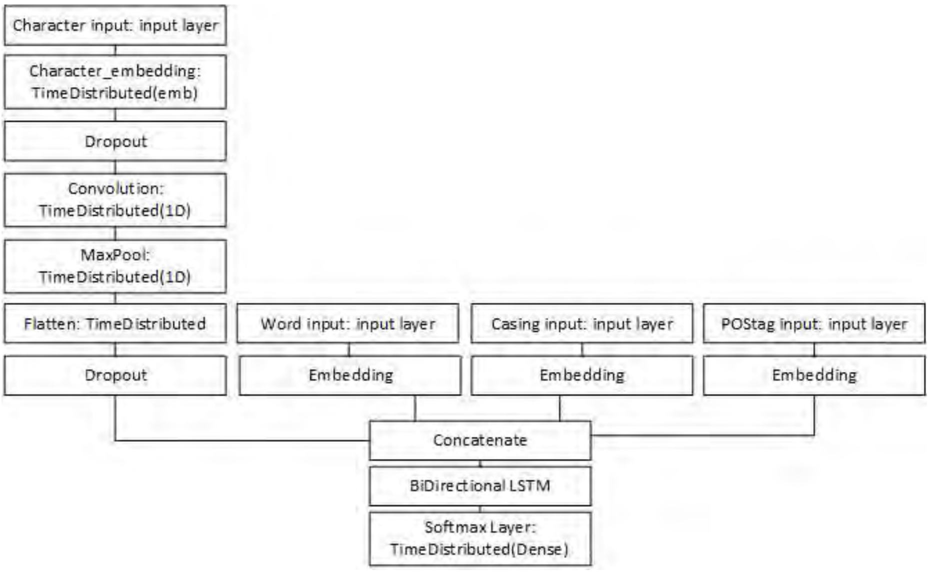

Character input: input layer
Character_embedding: TimeDistributed(emb)
Dropout
Convolution: TimeDistributed(1D)
MaxPool: TimeDistributed(1D)
Flatten: TimeDistributed
Dropout
Word input: input layer
Embedding
Casing input: input layer
Embedding
POStag input: input layer
Embedding
Concatenate
BiDirectional LSTM
Softmax Layer: TimeDistributed(Dense)

Fig. 1. Network architecture used in this work

- The second input layer uses the wikipedia2vec pretrained embeddings in Spanish language of 300 dimensions [4], mapping the existing vocabulary from the dataset.
- The third layer maps a vocabulary of eight casing types: numeric, allLower, allUpper, mainly_numeric, initialUpper, contains_digit, padding and other.
- The fourth layer maps into a one-hot embedding the POS tags existing in the vocabulary.

The system starts processing these four inputs independently, to finally merge them to be processed. The bidirectional LSTM layer Bi-LSTM [9] transforms the input data into two vectors of 200 units. Finally, the softmax function is used to obtain a prediction for locating and classifying sequences of words in the input text.

3 Evaluation and results

The evaluation of the proposed model was carried out using the MEDDOCAN corpus, that includes 1000 clinical cases, with around 495 thousand words, with an average of 494 words per clinical case. The corpus is annotated in both BRAT and i2b2 formats[5], and is divided in three sections: training, development and test. The training set comprises 500 clinical cases, and the development and

[4] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

[5] https://www.i2b2.org/

test set 250 clinical cases each. The test set is an additional collection of 2000 documents previously non-annotated for competition purposes.

The detailed information of the evaluation is in the MEDDOCAN competition related paper [8]. There are 29 categories for key phrases and the evaluation is divided in two subtasks. The first task is an entity-based evaluation and the second one evaluates whether spans belonging to sensitive phrases are detected correctly.

In the first task, we have obtained a F-score of 90%. In the second one we have obtained a 91.5% of F-score. The documents are semi-structured, which facilitates the correct learning of certain entities. For example, patient names begin with "Name: ". The main difficulty was the detection of discontinuous, overlapped or nested entities. For example, the names in different lines are annotated discontinuously: the entity "T1 NOMBRE_SUJETO_ASISTENCIA 29 63 Pedro De Miguel Rivera" is annotated in this system as "T1 NOMBRE_SUJETO_ASISTENCIA 47 63 De Miguel Rivera" and "T2 NOMBRE_SUJETO_ASISTENCIA 29 34 Pedro". Other difficulties are the recognition of entities in the text, such as the recognition of numbers as years (ages) or dates.

4 Conclusions and Future Works

In this work, we propose a hybrid Bi-LSTM and CNN model with four input layers that can recognize multi-word entities using the BIO encoding format for the labels. Our system achieve a satisfactory performance without requiring hand-crafted features.

We plan to experiment with other BIO-based formats to detect discontinuous, overlapped or nested entities, such as BMEWO-V [15]. Moreover, we will extend the annotation using domain-specific formats and using external sources (such as Wikipedia with cui2vec format [1]).

Acknowledgements

This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018). The authors want to thank PhD Juan J. Lastra-Diaz for his support.

References

1. Beam, A.L., Kompa, B., Fried, I., Palmer, N.P., Shi, X., Cai, T., Kohane, I.S.: Clinical concept embeddings learned from massive sources of multimodal medical data. arXiv preprint arXiv:1804.01486 (2018)
2. Fabregat, H., Araujo, L., Martinez-Romo, J.: Deep neural models for extracting entities and relationships in the new rdd corpus relating disabilities and rare diseases. Computer Methods and Programs in Biomedicine **164**, 121 – 129 (2018). https://doi.org/https://doi.org/10.1016/j.cmpb.2018.07.007, http://www.sciencedirect.com/science/article/pii/S0169260718301330
3. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)
4. Hofer, M., Kormilitzin, A., Goldberg, P., Nevado-Holgado, A.: Few-shot learning for named entity recognition in medical text. arXiv preprint arXiv:1811.05468 (2018)
5. Lara-Clares, A., Garcia-Serrano, A.: A few-shot learning model for knowledge discovery from ehealth documents (2019)
6. Lastra-Daz, J.J., Garcia-Serrano, A., Batet, M., Fernandez, M., Chirigati, F.: Hesml: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems **66**, 97–118 (2017)
7. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
8. Marimon, M., Gonzalez-Agirre, A., Intxaurrondo, A., Rodrguez, H., Lopez Martin, J.A., Villegas, M., Krallinger, M.: Automatic de-identification of medical texts in spanish: the meddocan track, corpus, guidelines, methods and evaluation of results. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). vol. TBA, p. TBA. CEUR Workshop Proceedings (CEUR-WS.org), Bilbao, Spain (Sep 2019), TBA

9. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
10. Segura-Bedmar, I., Martinez, P.: Simplifying drug package leaflets written in spanish by using word embedding. Journal of Biomedical Semantics **8**(45) (2017)
11. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a web-based tool for nlp-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 102–107. Association for Computational Linguistics (2012)
12. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence (2018)
13. Yamada, I., Asai, A., Shindo, H., Takeda, H., Takefuji, Y.: Wikipedia2vec: An optimized implementation for learning embeddings from wikipedia. arXiv preprint arXiv:1812.06280 (2018)
14. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine **13**(3), 55–75 (2018)
15. Zavala, R.M.R., Martınez, P., Segura-Bedmar, I.: A hybrid bi-lstm-crf model for knowledge recognition from ehealth documents. Proceedings of TASS **2172** (2018)

LSI2_UNED at eHealth-KD Challenge 2019

A Few-shot Learning Model for Knowledge Discovery from eHealth Documents

Alicia Lara-Clares[1] and Ana Garcia-Serrano[2]

[1] Universidad Nacional de Educación a Distancia (UNED), Spain
alara@lsi.uned.es
[2] Universidad Nacional de Educación a Distancia (UNED), Spain
agarcia@lsi.uned.es

Abstract. In this work, we describe a Few-Shot Learning approach for Named Entity Recognition (NER) in eHealth documents to identify and classify key phrases in a document (subtask A in the IberLEF eHealth-KD 2019 competition [10]). The architecture is an hybrid Bi-LSTM and CNN model with four input layers that can recognize multi-word entities using the BIO encoding format for the labels. The system obtained a F-score of 73.15% (baseline is 54,66%), with a 78,17% of precision, according to the eHealth-KD evaluation procedure. This improvement is reached mainly because (a) the correct selection of the hybrid model for NER that obtains better results using a POS tagger and (2) the addition of Wikidata entities to extend the vocabulary that improves the precision by nearly 10%.

Keywords: NER · Knowledge Discovery · Bi-LSTM · CNN · wikipedia2vec

1 Introduction

Currently, the number of medical data is growing at an exponential rate. Literature in the medical domain, moreover, is often found as unstructured or semi-structured data. In these cases, it is necessary to find methods to automatically extract and categorize the data contained in them, using different techniques as, for example, Named Entity Recognition (NER). NER aim is to recognize, identify and categorize pieces of information that refers to different entities of interest, i.e. a disease, a treatment or a patient name. First NER systems relied heavily on heuristic, hand-crafted features and language-specific knowledge as in the work presented by Rau[11] to extract and recognize company names.

In any research domain approximations based on the integration of different approaches or the integration of external resources are commonly used in order to improve the outcome of the research goal ([2], [3]). This is the case of neural networks that are especially successful in complex NLP tasks [17], as for example, G. Fabregat et al. [5] work that use a deep learning model for disabilities and diseases recognition using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Also research work with word embedding based techniques is frequently used, for example to simplify drug package leaflets written in Spanish [13] or to define reproducible experiments and replication datasets [8].

The aim of Few-shot Learning is to extract complex statistics and learn high level features using a very small set of training data. This problem has been addressed in several domains, such as [6] with one-shot learning, or [15] using zero-shot learning. M. Hofer et al.[7] demonstrate the effect of five sequential improvements on the learning capabilities of a neural network when having very few annotated examples, using as baseline the state-of-the-art NER architecture [4].

In this paper, we propose a hybrid Bi-LSTM CNN model following the work presented at [7]. Specifically, we have extended the model by adding a Part-of-speech (POS) tagging layer and information about multi-word entities. Moreover, in this work, we use wikipedia2vec [16], a pre-trained word embedding model from Wikipedia, and we extend the vocabulary by adding wikidata entities such diseases, health problems, etc. The results obtained in the eHealth-KD evaluation, improves the baseline by 18,5%.

The rest of the paper is organized as follows. In section 2, we describe the architecture of the system. Section 3 describes the evaluation process and results obtained. Finally, section 4 outlines the conclusions and future work.

2 System description

The system process is divided into two steps. First, it is necessary to pre-process the data and prepare it to be the input of the neural network and secondly after to process the data using the implemented neural network it is needed a post-proccess of the output to be evaluated in the tasks of the IberLEF eHealth-KD 2019 competition [10]. In the next sub-sections both descriptions are included.

2.1 Pre and Post processing of the data

All documents are pre-processed following the next steps. First, sentences are splitted and tokenized using the Stanford CoreNLP natural language processing toolkit [9], ignoring all non-alphanumeric symbols. Then, each token is annotated using the BIO scheme, to preserve the multi-word entities. After that, we get the POS tag of each token (using the Stanford Core-NLP POS tagger). After the proccessing of the input data, the output data has to be converted into the BRAT format [14]. The BRAT format allows to include some aspects of the data original file, because it store all the information together with the labels of each category and the positions of the tokens in the text. Given this difference between data formats the final step is to process the documents as shown in the Table 1: concept, POS tags and BIO-label.

Word	POS tag	BIO-label
No	ADV	O
existe	VERB	B-Action
un	NUM	O
tratamiento	NOUN	B-Concept
que	CONJ	O
restablezca	NOUN	B-Action
la	DET	O
funcion	NOUN	B-Concept
ovarica	ADJ	I-Concept
normal	ADJ	B-Concept

Table 1. Structure of processed data in this work

2.2 Network architecture

The network architecture used in this work is shown in Figure 1. It has four input layers, named as character level, word level, casing input and POS tag level, described in the following:

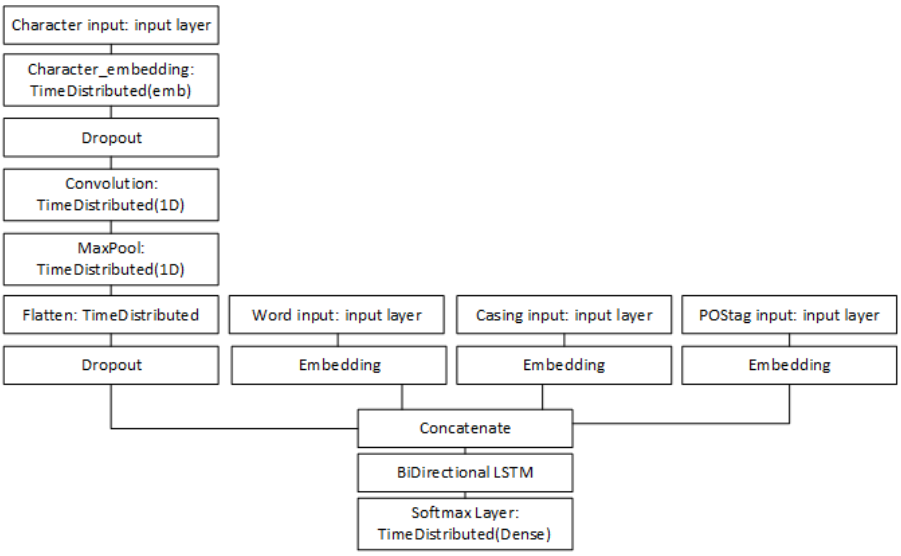

Character input: input layer
Character_embedding: TimeDistributed(emb)
Dropout
Convolution: TimeDistributed(1D)
MaxPool: TimeDistributed(1D)
Flatten: TimeDistributed
Dropout
Word input: input layer
Embedding
Casing input: input layer
Embedding
POStag input: input layer
Embedding
Concatenate
BiDirectional LSTM
Softmax Layer: TimeDistributed(Dense)

Fig. 1. Network architecture used in this work

- The first input layer corresponds to the character level. It starts with a character embedding that maps a vocabulary of 120 possible characters to an embedding initialized randomly. The maximum number of character per

word is 52. It has a dropout layer (with drop rate 0.5) used to avoid the risk of overfitting. Finally, it has a convolutional layer to process the 1-dimension character layer.
- The second input layer uses the wikipedia2vec pretrained embeddings in Spanish language of 300 dimensions [3], mapping the existing vocabulary from the dataset.
- The third layer maps a vocabulary of eight casing types: numeric, allLower, allUpper, mainly_numeric, initialUpper, contains_digit, padding and other.
- The fourth layer maps into a one-hot embedding the POS tags existing in the vocabulary.

The architecture starts processing these four inputs independently, to finally merge them into the last process. The bidirectional LSTM layer Bi-LSTM [12] transforms the input data into two vectors of 200 dimensions. In the last step, the softmax function is used to obtain a prediction for locating and classifying sequences of words in the input text.

3 Evaluation

The evaluation of the proposed model is carried out using the annotated corpus delivered in the 2019 competition that was extracted from the available MedlinePlus resources [4].

The IberLEF eHealth-KD 2019 corpus is divided in three sections: training, development and test. The training set contains a total of 600 sentences manually annotated in Brat and post-processed to match the input format. The development set has 100 annotated sentences, and the test data has 8800 non-annotated sentences for competition purposes.

Entity	**Tags**
Concept	B/I/O-Concept
Action	B/I/O-Action
Predicate	B/I/O-Predicate
Reference	B/I/O-Reference
Others	O

Table 2. Tokens labeled in this work

There are four categories or classes for key phrases:

1. **Concept**, a general category that indicates the key phrase is a relevant term, concept, idea, in the knowledge domain of the sentence.
2. **Action**, a concept that indicates a process or modification of other concepts.

[3] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

[4] https://medlineplus.gov/

3. **Predicate**, used to represent a function or filter of another set of elements, which has a semantic label in the text
4. **Reference**, a textual element that refers to a concept of the same sentence or of different one, which can be indicated by textual clues.

In this work, tokens are annotated with the previous categories using the different labels (see Table 2) following the BIO encoding format.

Then the scores are computed (correct, partial, missing, incorrect and spurious matches). The expected and actual output files do not need to agree on the ID for each phrase, nor on their order. The detailed information of the evaluation is in the eHealth KD competition website [5].

3.1 Results

In this work has been carried out a series of experiments on the development corpus delivered by eHealth-KD 2019. The most interesting results are briefly described below, and they can be seen in Table 3.

Method	Recall	**Precision**	**F1**
wikipedia2vec (300) + wikidata entities + POStags	**0,6796**	**0,8429**	**0,7525**
wikipedia2vec (300) + wikidata entities	0,6887	0,8109	0,7449
wikipedia2vec (300 dim)	0,6788	0,8151	0,7407
wikipedia2vec (100 dim) + POStags	0.6515	0.7918	0.7148
wikipedia2vec (100 dim)	0,6432	0,7864	0,7077
fastext (300 dim)	0,6184	0,7638	0,6834
SBWC_glove	0,5828	0,6998	0,636
SBWC_fastext	0,5728	0,6906	0,6262
fastext (300 dim) + POStags	0.5646	0.6973	0.624
baseline	**0,6358**	**0,5416**	**0,5849**

Table 3. Results of experiments in this work

The experiments have been focused on the embeddings model used, and in the impact of the POS tagging in the neural network results. We used four embedding models Fastext [6], FastText and GloVe embeddings from SBWC [7] and wikipedia2vec [8]. The first experimental conclusions achieved are:

[5] https://knowledge-learning.github.io/ehealthkd-2019/evaluation

[6] https://github.com/facebookresearch/fastText/blob/master/docs/pretrained-vectors.md

[7] https://github.com/dccuchile/spanish-word-embeddings

[8] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

1. The use of wikipedia2vec improves the performance and maintains the results from FasText in Spanish language.
2. Adding Wikidata entities improve the precision by approximate 10%.
3. POS tags do not improve results significantly in this task.
4. Adding fastext embeddings decreases system efficiency and does not improve results over wikipedia2vec.
5. Other embeddings in Spanish language are worse in terms of efficiency and accuracy.

4 Conclusions and Future Work

In this work, we propose a hybrid Bi-LSTM and CNN model with four input layers that can recognize multi-word entities using the BIO encoding format for the labels. The vocabulary is improved using Wikidata entities such as diseases, health problems, treatments, etc. This entities are labeled as BIO-concepts and added in the corpus data as sentences. Our system can achieve satisfactory performance without requiring hand-crafted features. Our results demonstrated that in Spanish language, the wikipedia2vec pretrained embedding vectors has better performance in this task than other embeddings such as Fastext or Glove.

We plan to experiment with other BIO-based formats to detect discontinuous, overlapped or nested entities, such as BMEWO-V [18]. Moreover, we will extend the annotation using domain-specific formats and using external sources (such as Wikipedia with cui2vec format [1]).

Acknowledgements

Funding: This work was supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018).

The authors want to thank PhD Juan J. Lastra-Diaz for his support.

References

1. Beam, A.L., Kompa, B., Fried, I., Palmer, N.P., Shi, X., Cai, T., Kohane, I.S.: Clinical concept embeddings learned from massive sources of multimodal medical data. arXiv preprint arXiv:1804.01486 (2018)
2. Benavent, J., Benavent, X., de Ves, E., Granados, R., Garcia-Serrano, A.: Experiences at imageclef 2010 using cbir and tbir mixing information approaches. In: CEUR Proceedings. 2010 CLEF September, Padua, Italy. vol. 1176 (2010)
3. Castellanos, A., Cigarran, J., Garcia-Serrano, A.: Formal concept analysis for topic detection: A clustering quality experimental analysis. Information Systems (66), 24–42 (2017)
4. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional lstm-cnns. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016)
5. Fabregat, H., Araujo, L., Martinez-Romo, J.: Deep neural models for extracting entities and relationships in the new rdd corpus relating disabilities and rare diseases. Computer Methods and Programs in Biomedicine **164**, 121 – 129 (2018)

6. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)
7. Hofer, M., Kormilitzin, A., Goldberg, P., Nevado-Holgado, A.: Few-shot learning for named entity recognition in medical text. arXiv preprint arXiv:1811.05468 (2018)
8. Lastra-Diaz, J.J., Garcia-Serrano, A., Batet, M., Fernandez, M., Chirigati, F.: Hesml: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems **66**, 97–118 (2017)
9. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
10. Piad-Morffis, A., Gutiérrez, Y., Consuegra-Ayala, J.P., Estevez-Velarde, S., Almeida-Cruz, Y., Muñoz, R., Montoyo, A.: Overview of the ehealth knowledge discovery challenge at iberlef 2019. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS.org (2019)
11. Rau, L.F.: Extracting company names from text. In: [1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application. vol. 1, pp. 29–32. IEEE (1991)
12. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
13. Segura-Bedmar, I., Martinez, P.: Simplifying drug package leaflets written in spanish by using word embedding. Journal of Biomedical Semantics **8**(45) (2017)
14. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: a web-based tool for nlp-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 102–107. Association for Computational Linguistics (2012)
15. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence (2018)
16. Yamada, I., Asai, A., Shindo, H., Takeda, H., Takefuji, Y.: Wikipedia2vec: An optimized implementation for learning embeddings from wikipedia. arXiv preprint arXiv:1812.06280 (2018)
17. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine **13**(3), 55–75 (2018)
18. Zavala, R.M.R., Martınez, P., Segura-Bedmar, I.: A hybrid bi-lstm-crf model for knowledge recognition from ehealth documents. Proceedings of TASS **2172** (2018)

Chapter 12

CLEF Workshops articles

Statistical graph matching for indexing Spanish biomedical documents*

Alicia Lara-Clares and Ana Garcia-Serrano

ETSI Informática
Universidad Nacional de Educación a Distancia (UNED)
{alara,agarcia}@lsi.uned.es

Abstract. In this work, we describe a statistical graph matching method for semantic indexing of documents from large-scale biomedical repositories in Spanish language provided at the MESINESP 2020 task (8th BioASQ Workshop [15]). The results obtained show enough accurate behavior, especially with respect to the rest of the results in the task. The execution time and computational requirements have been a priority in our approximation, which has proved to be efficient and robust for tackle further improvements.

Keywords: Biomedical semantic indexing · Knowledge Discovery · Graph matching

1 Introduction

Although the number of medical data is growing at an exponential rate, literature in the medical domain is often found as unstructured or semi-structured data. In these cases, it is necessary to find methods to automatically extract and categorize the data contained in them, using different techniques as, for example, biomedical semantic indexing.

The BioASQ [15] is an EU-funded support action [1] to set up a challenge on biomedical semantic indexing and question answering (QA). The MESINESP task is based on the use of resources such as a structured medical vocabulary DeCS [2] used in two databases for Spanish health content: IBECS [3] and LILACS [7]. The main objective of this task is the development of a semantic indexing tool for Spanish content. Other objectives are: (a) determining the current-state-of-the art, (b) identifying challenges, and (c) comparing the strategies and results to those published for English data[1].

* Supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018)

[1] https://temu.bsc.es/mesinesp/

In this paper, we propose a statistical graph matching method implemented as a module into the HESML framework [9–11]. This method obtains information on the frequency with which DeCS codes are annotated to rank the list of candidates that are extracted from the text following two different methods described in Section 2.1.

The results are encouraging enough, especially when compared to the rest of the experiments and knowing the main difficulties. We will continue working in this task with mixed approaches ([6]), looking forward to obtaining a robust and efficient method capable of correctly indexing DeCS codes. An important feature of this approach is its independence of the language.

The rest of the paper is organized as follows. In section 2, we describe the architecture of the system. Section 3 describes the evaluation process and the results obtained. Finally, section 4 outlines the conclusions and future work.

2 System description

The MESINESP task, is the first task on semantic indexing of Spanish medical texts, provides a dataset divided in training (318.658 documents), development (750 documents) and test (23.509 documents) sets. The average of DeCS codes per document is 8.12, and the document with the maximum number of codes has a total of 53 different ones. At a glance, the training set give us the idea of a scattered distribution of the codes annotated in the documents, as described in Table 1. Our proposed method try to overcome this problem using statistical information about the frequency of a DeCS code annotated in a document.

Total codes that appear more than 10 of documents%	6
Total codes that appear more than 1 of documents%	48
Total codes that appear less than 1 of documents%	33654
Total codes that never appear	22523
Total codes	**33702**

Table 1. Frequencies of the codes in the training set[2]

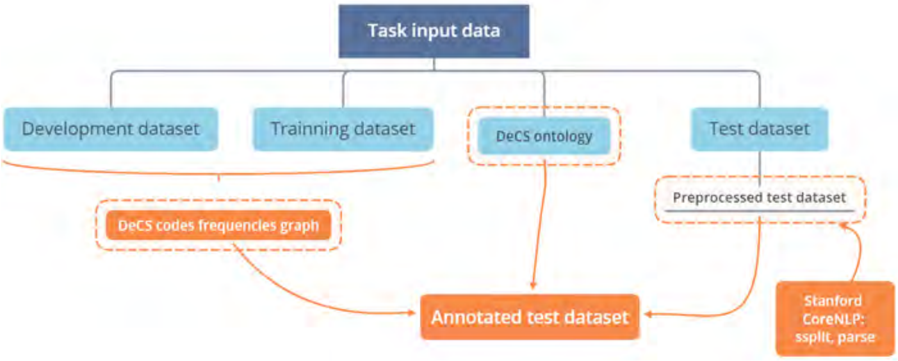
2.1 Proposed method

The method proposed herein is a first approximation focused on the efficiency and robustness of the system. Figure 1 represents the information flow to annotate the test set.

The process stages are the following:

1. Creation of the frequencies graph from the training and development data. In this step, a directed graph is developed, where each DeCS code represent a node, and the edges are the number of times the codes co-occur in each document.

Fig. 1. Information resources.



2. Parse the DeCS ontology and the list of codes and descriptors provided for the competition [3].
3. Split the sentences and identification of the chunks for each sentence using the Stanford CoreNLP library [12].

Once the test dataset is processed, the next step is the alignment of the documents with a list of DeCS code candidates. In this work, there has been carried out using two different methods, (a) exact matching and (b) graph-based matching. In the first one, every possible descriptor is matched with each document. If the descriptor exists in the text, it is selected as a candidate. In the second method, every chunk is compared with the list of possible descriptors, aligning as much DeCS codes as possible.

Other experiments have been planned but could not be carried out due to time constraints. The first one is the alignment of chunks with DeCS codes using a semantic sentence similarity measure, as for example, the Jaccard similarity [13, 4]. The HESML framework provides a set of semantic similarity measures that allows the comparison between every descriptor with all the available chunks. The problem of this approximation is that the annotation of each document takes about 30 seconds, so the method would take more than a week to annotate all the documents.

3 Evaluation and results

MESINESP task has been evaluated using the following measures: (a) Accuracy (Acc.), (b) Example Based Precision (EBP), (c) Example Based Recall (EBR), (d) Example Based F-Measure (EBF), (e) Macro Precision (MaP), (f) Macro Recall (MaR), (g) Macro F-Measure (MaF), (h) Micro Precision (MiP), (i) Micro

[3] Downloaded from https://temu.bsc.es/mesinesp/index.php/resources/

Recall (MiR) and (j) Micro F-Measure (MiF), but we only include in this section the Micro F-measure, since it is the official evaluation measure for this task.

Our results are shown in Table 2 including the first position and the baseline results of the task.

System	MiF	EBF	MaF	Acc	Position (MiF)
X-BERT BioASQ F1	0,1071	0,1051	0,0008	0,0575	1
Graph matching (ours)	0,0836	0,0846	0,001	0,0451	15
Exact matching (ours)	0,0826	0,0829	0,001	0,0442	18
BioASQ_Baseline	0,0161	0,0217	0,0022	0,0116	22

Table 2. This table shows the results or the two methods (Graph matching and Exact matching), as well as the best and the baseline ones

A total of 25 methods have been submitted to the MESINESP competition. Our work has focused on the efficiency and robustness of the method and executes the whole process in less than 30 minutes without requiring a training process. We have used the DeCS ontology and our new hypothesis is that the results will improve using another ontology-based similarity measure to the concept alignment without losing efficiency of the system.

The main difficulty in this task is derived from the use of a purely statistical method that prioritizes the most frequent terms and considers neither the ontology hierarchy for avoiding the annotation of redundant child-parent terms nor the less frequent codes that the experts annotate in the gold standard. For example, the terms ”tumor de mediastino” and ”mediastino” are annotated using our approximation for the document ID ”biblio-1000005”, but the experts only annotate the terms ”mediastino” and ”neoplasias del timo”. But, it happens that the term ”tumor de mediastino” is explicitly written in the title of the document and, for this reason, it is considered as relevant for our algorithm. On the other hand, the term ”pesar” is wrongly considered as relevant for our algorithm in most of the documents, because no semantics is considered in the selection of candidates.

4 Conclusions and Future Work

In this work, we describe a statistical graph matching method for semantically index documents from large-scale biomedical repositories in Spanish language provided at the MESINESP 2020 task [15]). The execution time and computational requirements have been priority factors in our approximation, giving us a first approach that is efficient and sufficiently robust to improve the results in the future.

Addressing the task, we understand that the Spanish language has not been thoroughly studied in a semantic indexing task, and there are only a few available tools. For example, there are some Name Entity Recognizers (NER) that find

UMLS concepts in Spanish biomedical documents, such as QuickUMLS [14] or IXAMedTagger [8]. But, as far as we know, there is not a NER tool for aligning DeCS codes with texts. Even more, the code sets tend to follow biased, unbalanced, and scattered distributions, as shown in a similar task of indexing CIE-10 codes for Spanish clinical documents [5].

In the future work, we are going to focus on the integration of the parser for the DeCS ontology in HESML. We will try to prove that our proposal will overcome the problems with the running time of the experiments based on sentence similarity measures by allowing the use of different ontology-based measures. Finally, we want also to test a new model that recognizes co-occurrence patterns beyond the basic measure of the frequency of occurrence of terms.

References

1. The bioasq challenge — bioasq.org. `http://www.bioasq.org/`, accessed: 2020-6-8
2. DeCS - health sciences descriptors. `http://decses.bvsalud.org/I/homepagei.htm`, accessed: 2020-6-10
3. IBECS. `https://www.isciii.es/QueHacemos/Servicios/Biblioteca/Paginas/IBECS.aspx`, accessed: 2020-6-10
4. . Manning, C.D., Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
5. Almagro, M., Unanue, R.M., Fresno, V., Montalvo, S.: ICD-10 coding of spanish electronic discharge summaries: An extreme classification problem. IEEE Access **8**, 100073–100083 (2020)
6. Benavent, J., Benavent, X., de Ves, E., Granados, R., Garcia-Serrano, A.: Experiences at imageclef 2010 using cbir and tbir mixing information approaches. In: CEUR Proceedings. 2010 CLEF September, Padua, Italy. vol. 1176 (2010)
7. BIREME (http://www.bireme.br/). LILACS Unity (http://metodologia.lilacs.bvs.br/): LILACS database) :. `http://metodologia.lilacs.bvsalud.org/php/level.php?&component=19`, accessed: 2020-6-10
8. Gojenola, K., Oronoz, M., Pérez, A., Casillas, A., Taldea, I.X.A.: IxaMed: Applying freeling and a perceptron sequential tagger at the shared task on analyzing clinical texts. In: SemEval@ COLING. pp. 361–365. ixa.eus (2014)
9. Lastra-Diaz, J.J., Garcia-Serrano, A., Batet, M., Fernandez, M., Chirigati, F.: Hesml: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. Information Systems **66**, 97–118 (2017)
10. Lastra-Díaz, J.J., Goikoetxea, J., Taieb, M.A.H., others: A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Applications of Artificial . . . (2019)
11. Lastra-Díaz, J.J., Goikoetxea, J., Hadj Taieb, M.A., García-Serrano, A., Aouicha, M.B., Agirre, E.: Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity. Data Brief **26**, 104432 (Oct 2019)
12. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60. aclweb.org (2014)

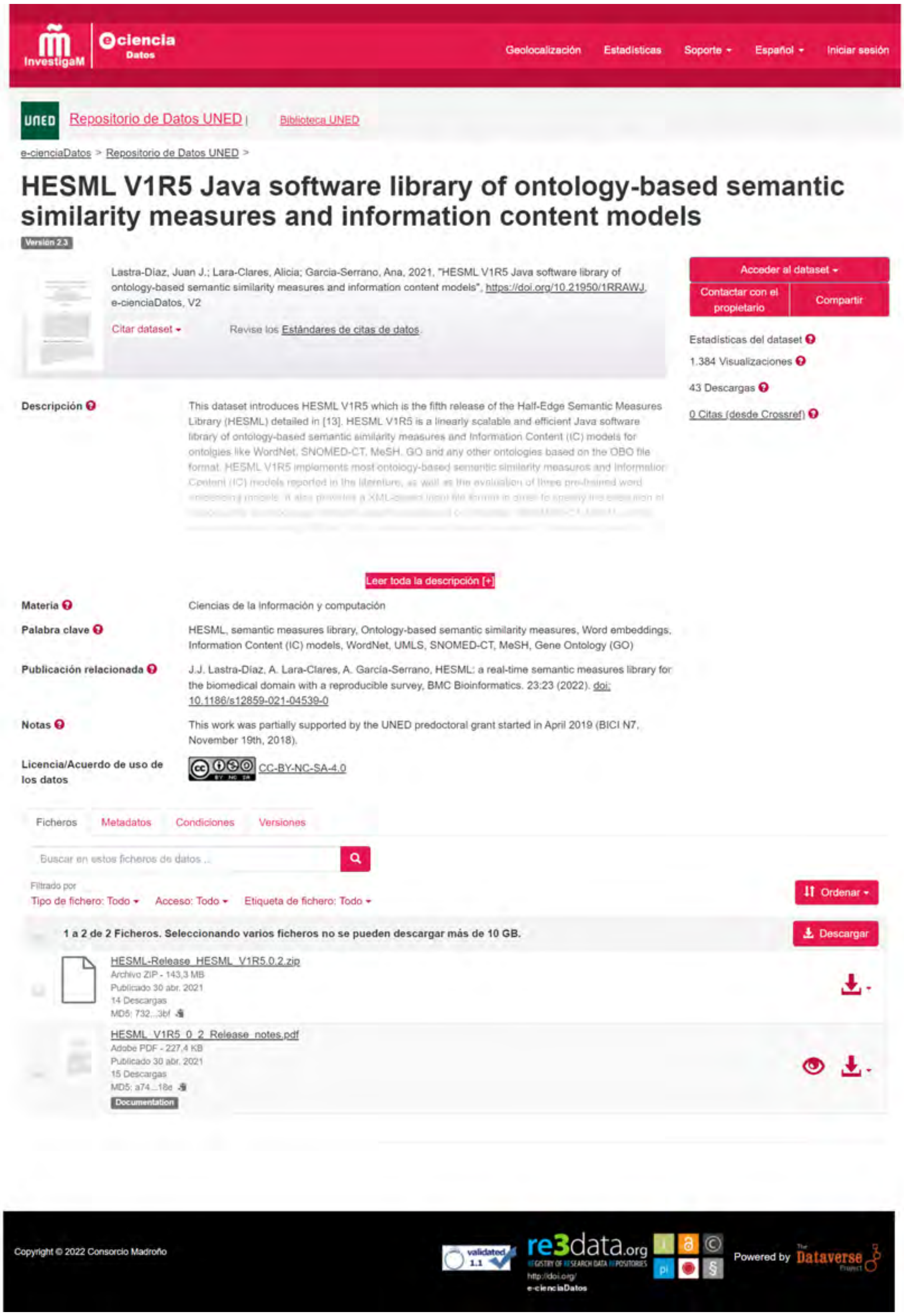
13. P. Jaccard: Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. sci. nat. **44**, 223–270 (1908)
14. Soldaini, L., Goharian, N.: Quickumls: a fast, unsupervised approach for medical concept extraction. In: MedIR workshop, sigir. pp. 1–4. ir.cs.georgetown.edu (2016)
15. Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M.R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., Almirantis, Y., Pavlopoulos, J., Baskiotis, N., Gallinari, P., Artiéres, T., Ngomo, A.C.N., Heino, N., Gaussier, E., Barrio-Alvers, L., Schroeder, M., Androutsopoulos, I., Paliouras, G.: An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics **16**, 138 (Apr 2015)

Part III

Software Libraries, Protocols and Datasets

Chapter 13

HESML V1R5 Semantic Measure Library



Repositorio de Datos UNED | Biblioteca UNED

e-cienciaDatos > Repositorio de Datos UNED >

HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models

Versión 2.3

Lastra-Díaz, Juan J.; Lara-Clares, Alicia; Garcia-Serrano, Ana, 2021, "HESML V1R5 Java software library of ontology-based semantic similarity measures and information content models", https://doi.org/10.21950/1RRAWJ, e-cienciaDatos, V2

Citar dataset ▾ Revise los Estándares de citas de datos.

Estadísticas del dataset

1.384 Visualizaciones

43 Descargas

0 Citas (desde Crossref)

Descripción This dataset introduces HESML V1R5 which is the fifth release of the Half-Edge Semantic Measures Library (HESML) detailed in [13]. HESML V1R5 is a linearly scalable and efficient Java software library of ontology-based semantic similarity measures and Information Content (IC) models for ontolgies like WordNet, SNOMED-CT, MeSH, GO and any other ontologies based on the OBO file format. HESML V1R5 implements most ontology-based semantic similarity measures and Information Content (IC) models reported in the literature, as well as the evaluation of three pre-trained word embedding models. It also provides a XML-based input file format in order to specify the execution of [illegible]

Materia Ciencias de la información y computación

Palabra clave HESML, semantic measures library, Ontology-based semantic similarity measures, Word embeddings, Information Content (IC) models, WordNet, UMLS, SNOMED-CT, MeSH, Gene Ontology (GO)

Publicación relacionada J.J. Lastra-Díaz, A. Lara-Clares, A. García-Serrano, HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey, BMC Bioinformatics. 23:23 (2022). doi: 10.1186/s12859-021-04539-0

Notas This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018).

Licencia/Acuerdo de uso de los datos CC-BY-NC-SA-4.0

1 a 2 de 2 Ficheros. Seleccionando varios ficheros no se pueden descargar más de 10 GB.

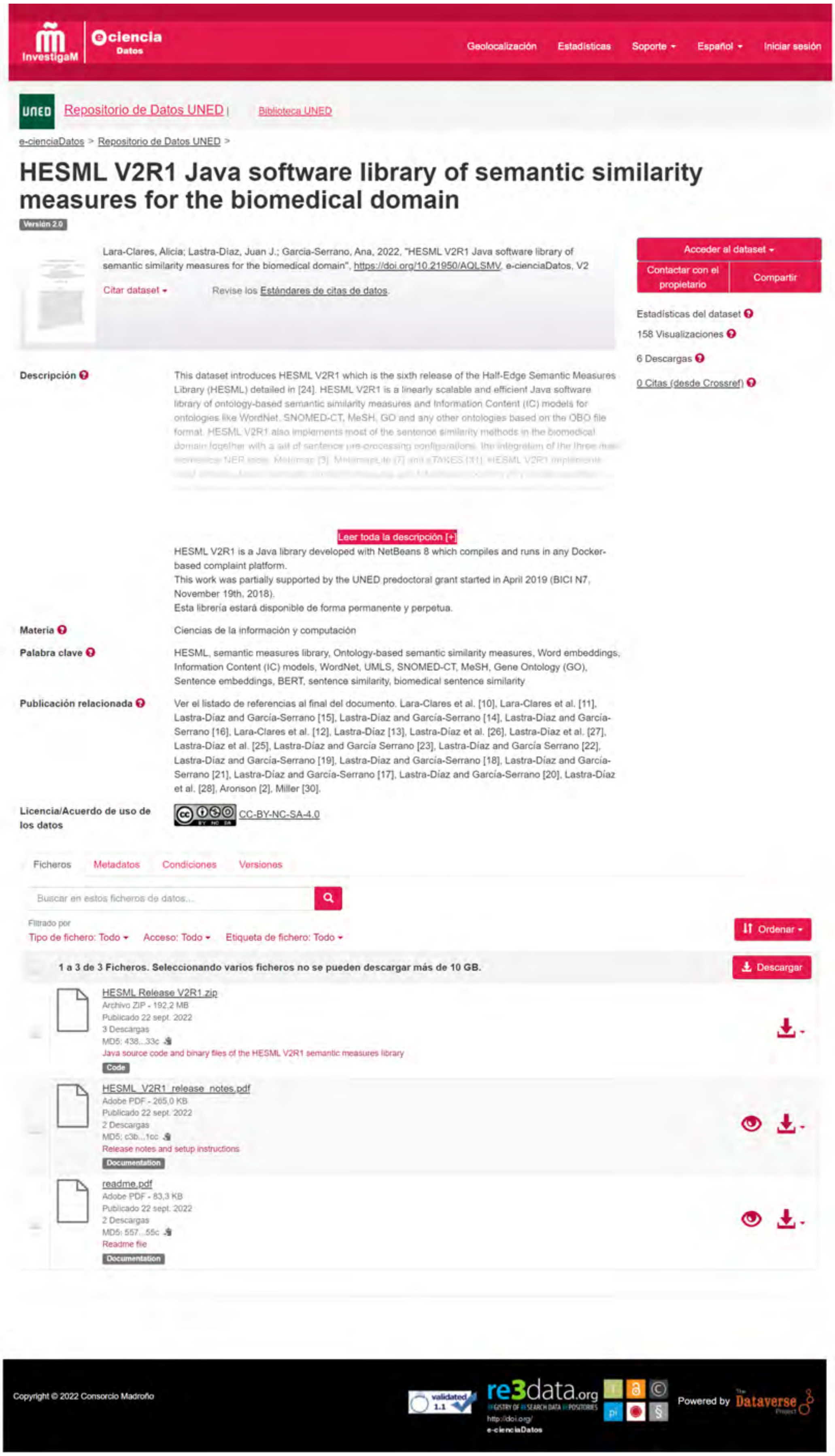
HESML-Release_HESML_V1R5.0.2.zip
Archivo ZIP - 143,3 MB
Publicado 30 abr. 2021
14 Descargas
MD5: 732...3bf

HESML_V1R5_0_2_Release_notes.pdf
Adobe PDF - 227,4 KB
Publicado 30 abr. 2021
15 Descargas
MD5: a74...18e
Documentation

validated 1.1
re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES
http://doi.org/
e-cienciaDatos
Powered by The Dataverse Project

Chapter 14

HESML V2R1 Semantic Measure Library



HESML V2R1 Java software library of semantic similarity measures for the biomedical domain

Versión 2.0

Lara-Clares, Alicia; Lastra-Díaz, Juan J.; Garcia-Serrano, Ana, 2022, "HESML V2R1 Java software library of semantic similarity measures for the biomedical domain", https://doi.org/10.21950/AQLSMV, e-cienciaDatos, V2

Citar dataset ▾ Revise los Estándares de citas de datos.

Estadísticas del dataset

158 Visualizaciones

6 Descargas

0 Citas (desde Crossref)

Descripción This dataset introduces HESML V2R1 which is the sixth release of the Half-Edge Semantic Measures Library (HESML) detailed in [24]. HESML V2R1 is a linearly scalable and efficient Java software library of ontology-based semantic similarity measures and Information Content (IC) models for ontologies like WordNet, SNOMED-CT, MeSH, GO and any other ontologies based on the OBO file format. HESML V2R1 also implements most of the sentence similarity methods in the biomedical domain together with a set of sentence pre-processing configurations, the integration of the three main biomedical NER tools: Metamap [3], MetamapLite [7] and cTAKES [11]. HESML V2R1 implements [illegible]

Leer toda la descripción [+]

HESML V2R1 is a Java library developed with NetBeans 8 which compiles and runs in any Docker-based complaint platform.

This work was partially supported by the UNED predoctoral grant started in April 2019 (BICI N7, November 19th, 2018).

Esta librería estará disponible de forma permanente y perpetua.

Materia Ciencias de la información y computación

Palabra clave HESML, semantic measures library, Ontology-based semantic similarity measures, Word embeddings, Information Content (IC) models, WordNet, UMLS, SNOMED-CT, MeSH, Gene Ontology (GO), Sentence embeddings, BERT, sentence similarity, biomedical sentence similarity

Publicación relacionada Ver el listado de referencias al final del documento. Lara-Clares et al. [10], Lara-Clares et al. [11], Lastra-Díaz and García-Serrano [15], Lastra-Díaz and García-Serrano [14], Lastra-Díaz and García-Serrano [16], Lara-Clares et al. [12], Lastra-Díaz [13], Lastra-Díaz et al. [26], Lastra-Díaz et al. [27], Lastra-Díaz et al. [25], Lastra-Díaz and García Serrano [23], Lastra-Díaz and García Serrano [22], Lastra-Díaz and García-Serrano [19], Lastra-Díaz and García-Serrano [18], Lastra-Díaz and García-Serrano [21], Lastra-Díaz and García-Serrano [17], Lastra-Díaz and García-Serrano [20], Lastra-Díaz et al. [28], Aronson [2], Miller [30].

Licencia/Acuerdo de uso de los datos CC-BY-NC-SA-4.0

1 a 3 de 3 Ficheros. Seleccionando varios ficheros no se pueden descargar más de 10 GB.

HESML_Release_V2R1.zip
Archivo ZIP - 192,2 MB
Publicado 22 sept. 2022
3 Descargas
MD5: 438...33c
Java source code and binary files of the HESML V2R1 semantic measures library
Code

HESML_V2R1_release_notes.pdf
Adobe PDF - 265,0 KB
Publicado 22 sept. 2022
2 Descargas
MD5: c3b...1cc
Release notes and setup instructions
Documentation

readme.pdf
Adobe PDF - 83,3 KB
Publicado 22 sept. 2022
2 Descargas
MD5: 557...55c
Readme file
Documentation

validated 1.1
re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES
http://doi.org/
e-cienciaDatos
Powered by The Dataverse Project

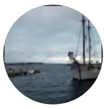
Chapter 15

Reproducibility protocol at protocols.io

A reproducibility protocol and dataset on the biomedical sentence similarity V.4

Alicia Lara Clares[1], Juan J. Lastra-Díaz[1], Ana Garcia-Serrano[1]

[1]UNED



Alicia Lara Clares

VERSION 4

SEP 20, 2022

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.36wgq429xvk5/v4

External link:
http://hesml.lsi.uned.es/

Protocol Citation: Alicia Lara Clares, Juan J. Lastra-Díaz, Ana Garcia-Serrano 2022. A reproducibility protocol and dataset on the biomedical sentence similarity. **protocols.io** https://dx.doi.org/10.17504/protocols.io.36wgq429xvk5/v4 Version created by Alicia Lara Clares

MANUSCRIPT CITATION:
Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. A reproducible experimental survey on biomedical sentence similarity: a string-based method sets the state of the art. Submitted to PLoS One. 2022.

Protocol status: Working
We use this protocol and it's working

Created: Sep 19, 2022

Last Modified: Sep 20, 2022

PROTOCOL integer ID: 70259

Keywords: hesml, sentence similarity, semantic similarity, ontologies, bert

ABSTRACT

This protocol introduces a set of reproducibility resources with the aim of allowing the exact replication of the experiments introduced by our main paper [1], which introduces the largest and for the first time reproducible experimental survey on biomedical sentence similarity. HESML V2R1 [2] is the sixth release of our Half-Edge Semantic Measures Library (HESML), which is a linearly scalable and efficient Java software library of ontology-based semantic similarity measures and Information Content (IC) models for ontologies like WordNet, SNOMED-CT, MeSH and GO.
This protocol sets a self-contained reproducibility platform which contains the Java source code and binaries of our main benchmark program, as well as a Docker image which allows the exact replication of our experiments in any software platform supported by Docker, such as all Linux-based operating systems, Windows or MacOS. All the necessary resources for executing the experiments are published in the permanent repository [3]

Our benchmark program is distributed with the UMLS SNOMED-CT and MeSH ontologies by courtesy of the US National Library of Medicine (NLM), as well as all needed software components with the aim of making the setup process easier. Our Docker image provides an exact virtual replica of the machine in which we ran our experiments, thus removing the need to carry-out any tedious setup process, such as the setup of the Named Entity Recognizer tools and other software components. (2022-02-20)

[1] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. A reproducible experimental survey on biomedical sentence similarity: a string-based method sets the state of the art. Submitted to PLoS One. 2022.

[2] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. HESML V2R1 Java software library of semantic similarity measures for the biomedical domain. e-cienciaDatos; 2022. doi:10.21950/DOI

[3] Lara-Clares, Alicia; Lastra-Díaz, Juan J.; Garcia-Serrano, Ana, 2021, "Reproducible experiments on word and sentence similarity measures for the biomedical domain",https://doi.org/10.21950/EPNXTR, e-cienciaDatos, V2

GUIDELINES

The Docker image provides all software pre-installed, which means that it is not necessary to install them to reproduce the results of this paper.

MATERIALS

All the required materials to reproduce the experiments in this protocol are published in our reproducibility dataset

CITATION

Lara-Clares, Alicia; Lastra-Díaz, Juan J.; Garcia-Serrano, Ana (2022). Reproducible experiments on word and sentence similarity measures for the biomedical domain. e-cienciaDatos, V2.

LINK

https://doi.org/10.21950/EPNXTR

BEFORE START INSTRUCTIONS

Our benchmarks can be reproduced in any Docker-complaint platform, such as Windows, MacOS or any Linux-based system by following a similar setup to that introduced herein.

In order to obtain a decrypt password for downloading the required files, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file. Likewise, you should obtain and sign a Data User Agreement from the Mayo Clinic to use the MedSTS dataset by sending the authors the Data User Agreement form, https://n2c2.dbmi.hms.harvard.edu/data-use-agreement.

Installing Docker on Ubuntu

1 If Docker is not installed in your machine, instructions below install latest version of Docker CE. For further details, we refer the reader to the official Docker setup page https://docs.docker.com/install/linux/docker-ce/ubuntu/ 5m

First, we update the system:

Command

```
sudo apt-get update
```

We install the dependencies:

Command

```
sudo apt-get install ca-certificates curl gnupg lsb-release && curl -fsSL
https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
```

We set stable Docker release

Command

```
echo  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

We install Docker engine

Command

```
sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Note

If the installation detailed below fails, you can install Docker for Ubuntu:

Command

```
sudo apt  install docker.io
```

Downloading resources from the repository

2 Now, we download and decrypt the external resources such as pre-trained models and dependencies. 10h

First, we create a data directory which will contain all the datasets, pre-trained models and dependencies for executing the experiments

Command

```
cd /home/[user]/Desktop && mkdir HESML_DATA && cd HESML_DATA
```

Now, we download extract the BERT pretrained models compressed file (20,2 GB) to the HESML_DATA

Command

```
wget https://doi.org/10.21950/BERTExperiments.tar.gz && tar xvf BERTExperiments.tar.gz
```

We also download and extract the pre-trained character and sentence embeddings models (20GB) in the same directory

Command

```
wget https://doi.org/10.21950/CharacterAndSentenceEmbeddings.tar.gz && tar xvf
CharacterAndSentenceEmbeddings.tar.gz
```

We download and extract the pre-trained word embedding models (40GB) in the same directory

Command

```
wget https://doi.org/10.21950/WordEmbeddings.tar.gz && tar xvf WordEmbeddings.tar.gz
```

And finally, we download, decrypt and extract the rest of dependencies (10GB), such as datasets, UMLS, Java libraries, cTAKES, Metamap and MetamapLite.

Command

```
sudo apt install -y ccrypt && wget https://doi.org/10.21950/Dependencies.tar.gz.cpt &&
ccrypt -d Dependencies.tar.gz.cpt
```

Safety information

In order to obtain a decrypt password for the Dependencies.tar.gz file, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file. Likewise, you should obtain and sign a Data User Agreement from the Mayo Clinic to use the MedSTS dataset by sending the authors the Data User Agreement form, https://n2c2.dbmi.hms.harvard.edu/data-use-agreement

Command

```
tar xvf Dependencies.tar.gz
```

Now, we can remove all the compressed files

Command

```
rm -r *.tar.gz
```

Expected result

At the end of this section, you should have a directory named HESML_DATA on your local machine with this file structure:

```
.
./ImportedLibs
./WordEmbeddings
./UMLS
./SentenceEmbeddings
./ReproducibleResults
./SentenceSimDatasets
./FlairEmbeddings
./public_mm_lite
./apache-ctakes-4.0.0.1-src
./BERTExperiments
./dist
./public_mm
```

Create and run a Docker container with HESML and depend...

3 In this step, we create and run a Docker container which have pre-installed all the necessary 15m

⚠ software for executing the experiments.

Command

We pull the HESML docker image from DockerHub, which contains all the pre-installed software for executing the experiments.

```
docker pull alicialara/hesml_v2r1:latest
```

Note

NOTE: Alternatively, the docker image can also be downloaded and extracted from our permanent repository:

CITATION

Lara-Clares, Alicia; Lastra-Díaz, Juan J.; Garcia-Serrano, Ana (2022). Reproducible experiments on word and sentence similarity measures for the biomedical domain. e-cienciaDatos, V2.

LINK

https://doi.org/10.21950/EPNXTR

In this case, you can import the Docker file by following the next command

Command

```
wget https://doi.org/10.21950/hesml_STS_dockerRelease.tar.gz && tar xvf
hesml_STS_dockerRelease.tar.gz && docker load --input hesml_STS_dockerRelease.tar.gz
```

Now, we create, run and attach to the Docker container named "HESMLV2R1" which will share a volume with the HESML_DATA directory.

Note

NOTE: you have to modify the variable [PATH_TO_HESML_DATA_DIRECTORY] using the path from your local machine.

Command

```
docker run --name=HESMLV2R1 -it -v
[PATH_TO_HESML_DATA_DIRECTORY]/HESML_DATA/:/home/user/HESML_DATA
alicialara/hesml_v2r1:latest /bin/bash
```

In the following, we will be working on the Docker container, which has been attached in the previous step.

Now, we clone the HESML repository from Github

Command

```
cd /home/user && git clone --branch HESML_STS_paper_experiments
https://github.com/jjlastra/HESML.git
```

And we copy the external libraries and dependencies into the HESMLSTSclient directory and we copy the last HESML core jar file into the client directory

Command

```
cd /home/user/HESML_DATA/ && cp -r dist/lib
/home/user/HESML/HESML_Library/HESMLSTSclient/dist && cd
/home/user/HESML/HESML_Library && cp HESML/dist/HESML-V2R1.0.1.jar
HESMLSTSclient/dist/lib
```

Expected result

At the end of this section, you should have the following directories in the /home/user directory of the Docker container:

.
./HESML
./HESML_DATA

- The HESML directory contains the sources from Github with all the necessary dependencies and libraries for executing the experiments.
- The HESML_DATA directory contains the pre-trained models, python virtual environments and the NER tools for executing the experiments

Launch the Metamap and cTAKES services

4 The experiments evaluated herein use the Metamap [4], MetamapLite [5] and cTAKES [6] external NER tools to annotate CUI codes on the sentences. Thus, we have to launch the NER tools services following the next steps. 5m

Note

[4] Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010;17: 229–236. doi:10.1136/jamia.2009.002733

[5] Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. J Am Med Inform Assoc. 2017;24: 841–844. doi:10.1093/jamia/ocw177

[6] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010;17: 507–513. doi:10.1136/jamia.2009.001560

First, we open the Metamap directory

Command

```
cd /home/user/HESML_DATA/public_mm
```

We start the Metamap dependency services

Command

We start the Metamap services. (Docker version 20.10.12)

```
./bin/skrmedpostctl start && ./bin/wsdserverctl start
```

Note

Note: Before executing the next step, wait until the following message appears (2-3 minutes): "WSD Server databases and disambiguation methods have been initialized." and press the "Enter" key.

Now, we start the Metamap service

Command

```
./bin/mmserver &
```

Then, press "Enter" key and execute the next step using your UMLS KEY.

Command

```
export ctakes_umls_apikey=[ENTER YOUR UMLS API KEY]
```

Safety information

In order to obtain a UMLS KEY, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html.

Expected result

At the end of this section, you should have initialized the NER tools services, and you can execute all the experiments evaluated in our primary paper:

CITATION

Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. (2022). A reproducible experimental survey on biomedical sentence similarity: a string-based method sets the state of the art. Submitted to PLoS One.

UBUNTU-based instructions to run our benchmarks on a Do...

5 The final step is the execution of the experiments evaluated in out primary paper. 1d

CITATION

Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. (2022). A reproducible experimental survey on biomedical sentence similarity: a string-based method sets the state of the art. Submitted to PLoS One.

To run the experiments, first step into the HESMLSTSclient directory

Command

```
cd /home/user/HESML/HESML_Library/HESMLSTSclient/
```

Before running the experiments, remove previous results and temporal files:

Command

```
rm -r ../ReproducibleExperiments/BioSentenceSimilarity_paper/BioSentenceSimFinalRawOutputFiles/* && rm -r ../ReproducibleExperiments/BioSentenceSimilarity_paper/BioSentenceSimFinalProcessedOutputFiles/* && rm Execution_times_* && rm -r tmp* && rm -r /tmp/tmp*
```

Now, execute the HESMLSTSclient with the default options

Command

```
java -jar -Xms30g dist/HESMLSTSclient.jar
```

Note

Note that this experiment take more than 24 hours of execution time in a desktop computer with an AMD Ryzen 7 5800x CPU (16 cores) with 64 Gb RAM and 2TB Gb SSD disk

Expected result

At the end of this section, you should find all the raw output files in your HESML_DATA directory

```
[PATH_TO_HESML_DATA_DIRECTORY]/HESML_DATA/ReproducibleResults/BioSentenceSimilarity_paper/BioSentenceSimFinalRawOutputFiles
.
├── raw_similarity_BIOSSES_BESTCOMBS.csv
├── raw_similarity_BIOSSES_COMBestWorst.csv
├── raw_similarity_BIOSSES_LiBlockNER.csv
├── raw_similarity_BIOSSES_NERexperiment.csv
├── raw_similarity_CTR_BESTCOMBS.csv
├── raw_similarity_CTR_COMBestWorst.csv
├── raw_similarity_CTR_LiBlockNER.csv
├── raw_similarity_CTR_NERexperiment.csv
├── raw_similarity_MedSTSFull_BESTCOMBS.csv
├── raw_similarity_MedSTSFull_COMBestWorst.csv
├── raw_similarity_MedSTSFull_LiBlockNER.csv
└── raw_similarity_MedSTSFull_NERexperiment.csv
```

These raw output files will be used in the post-processing stage to create the tables 8, 10-17, figure 5 and appendix A detailed in our primary paper [1].

5.1 [OPTIONAL] Running the pre-processing experiments 2w

In our primary paper [1], we also evaluate the pre-processing configurations of each method, which are detailed in tables 7 and 9, as well as the appendix B of the same paper. This pre-processing experiments are evaluated using the HESMLSTSImpactEvaluationclient software included in the HESML V2R1 software release [6].

[6] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. HESML V2R1 Java software library of semantic similarity measures for the biomedical domain. e-cienciaDatos; 2022. doi:10.21950/DOI

Safety information

It is important to note that the execution of the pre-processing experiments requires high computational requirements and running times (more than 2 weeks), since they perform more than 1100 pre-processing combinations in total.

To execute the pre-processing experiments, run the following commands

Command

```
cd /home/user/HESML_DATA/ && cp -r dist/lib
/home/user/HESML/HESML_Library/HESMLSTSImpactEvaluationclient/dist && cd
/home/user/HESML/HESML_Library && cp HESML/dist/HESML-V2R1.0.1.jar
HESMLSTSImpactEvaluationclient/dist/lib
```

Command

```
cd /home/user/HESML/HESML_Library/HESMLSTSImpactEvaluationclient/ && java -jar -
Xms30g dist/HESMLSTSImpactEvaluationclient.jar
```

Post-processing the experiments

20m

6 The post-processing stage use the RStudio software installed in the local machine to create the final latex tables and CSV files. 20m

Note

NOTE: Now, the post-processing experiments are evaluated in the local machine, under the HESML_DATA directory. You can detach the HESMLV2R1 docker container by clicking the key sequence: CTRL+p, CTRL+q

In our experiments, we use the last release of RStudio software (Version 1.4) with R version 4.1.2 (2021-11-01). We also install the following packages for executing the post-processing scripts:

- collections

- kableExtra
- knitr
- readr
- stringr
- xtable
- dplyr
- ggpubr
- ggqqplot
- ggpubr
- ggplot2

After executing the experiments, the raw output files, as well as the R post-processing scripts are automatically copied into the HESML_DATA directory, in a new directory named "ReproducibleResults". Before executing the post-processing scripts, it is necessary to modify the file permissions following the next step:

Command

```
cd [PATH_TO_HESML_DATA_DIRECTORY]/HESML_DATA && sudo chmod -R 777
ReproducibleResults/
```

The tables 8, 10-17, figure 5 and appendices A and B are created executing the following R scripts marked in **bold** as follows:

.[PATH_TO_HESML_DATA_DIRECTORY]/HESML_DATA/ReproducibleResults/Post-scripts
├── **bio_sentence_sim_tables.R**
├── **bio_analytics_biosses.R**
├── **bio_analytics_ctr.R**
├── **bio_analytics_medsts.R**
├── **bio_sentence_sim_allExperiments_analyzingtablesPreprocessing.R**
├── **bio_sentence_sim_pvaluesLiBlock.R**
├── **bio_sentence_sim_pvaluesNER.R**
├── **bio_sentence_sim_pvalues.R**
├── bio_sentence_sim_scripts
│ ├── readBERT.R
│ ├── readBESTCOMBS.R
│ ├── readFlair.R
│ ├── readLiBlockNERExperiment.R

```
│   ├── readNERexperiment.R
│   ├── readOurWE.R
│   ├── readSent2Vec.R
│   ├── readString.R
│   ├── readSWEM.R
│   ├── readTest.R
│   ├── readUBSM.R
│   ├── readUSE.R
│   └── readWBSM.R
```

- **bio_sentence_sim_tables.R** : Creates the tables 8,10,11 and 12 in our primary paper [1] as well as all the tables from appendix B. It is also used to extract the best and worst pre-processing configuration in the table 9 of the same paper
- **bio_sentence_sim_pvalues.R** : Creates the tables of the appendix A in our primary paper [1].
- **bio_sentence_sim_allExperiments_analyzingtablesPreprocessing.R** : Creates the tables with all the p-values of the pre-processing experiments using the HESMLSTSImpactEvaluationclient, which are used in the table 9 of our main paper.
- **bio_sentence_sim_pvaluesLiBlock.R** : Creates a table with the LiBlock NER experiments which is used to detail the p-values in table 12 of the main paper [1].
- **bio_sentence_sim_pvaluesNER.R** : Creates a table with the NER experiments which is used to detail the p-values in table 11 of the main paper [1].
- **bio_analytics_biosses.R, bio_analytics_medsts.R and bio_analytics_ctr.R**: Creates the figure 5 and is used to create the tables 13-17 of our primary paper [1].

Note

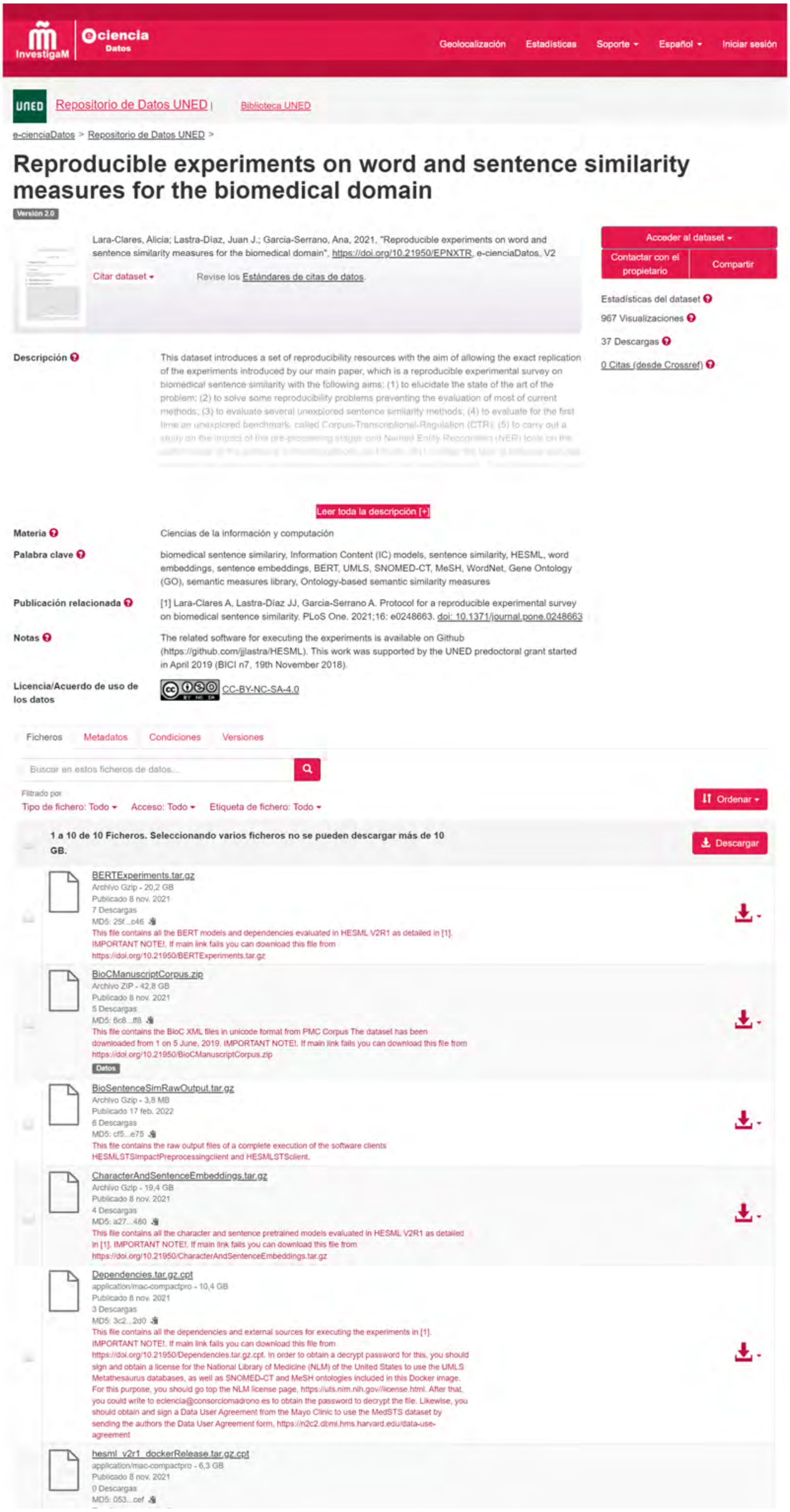
The "bio_sentence_sim_scripts" directory contains a set of R scripts to parse the output raw files created by the execution of HESMLSTSclient and HESMLSTSImpactEvaluationclient.

Expected result

The execution of all the R scripts listed below produces a ser of TXT and CSV files containing all the post-processed results, which are used to create tables 8, 10-17, figure 5 and appendices A and B of our primary paper [1].

Chapter 16

Benchmarks between Semantic Measure Libraries


InvestigaM
e-ciencia Datos
Geolocalización
Estadísticas
Soporte
Español
Iniciar sesión
UNED
Repositorio de Datos UNED |
Biblioteca UNED
e-cienciaDatos > Repositorio de Datos UNED >

Reproducible experiments on word and sentence similarity measures for the biomedical domain

Versión 2.0

Lara-Clares, Alicia; Lastra-Díaz, Juan J.; García-Serrano, Ana, 2021, "Reproducible experiments on word and sentence similarity measures for the biomedical domain", https://doi.org/10.21950/EPNXTR, e-cienciaDatos, V2

Citar dataset
Revise los Estándares de citas de datos.
Acceder al dataset
Contactar con el propietario
Compartir

Estadísticas del dataset

967 Visualizaciones

37 Descargas

0 Citas (desde Crossref)

Descripción This dataset introduces a set of reproducibility resources with the aim of allowing the exact replication of the experiments introduced by our main paper, which is a reproducible experimental survey on biomedical sentence similarity with the following aims: (1) to elucidate the state of the art of the problem; (2) to solve some reproducibility problems preventing the evaluation of most of current methods; (3) to evaluate several unexplored sentence similarity methods; (4) to evaluate for the first time an unexplored benchmark, called Corpus-Transcriptional-Regulation (CTR); (5) to carry out a study on the impact of the pre-processing stages and Named Entity Recognizers (NER) tools on the [illegible]

Materia Ciencias de la información y computación

Palabra clave biomedical sentence similariry, Information Content (IC) models, sentence similarity, HESML, word embeddings, sentence embeddings, BERT, UMLS, SNOMED-CT, MeSH, WordNet, Gene Ontology (GO), semantic measures library, Ontology-based semantic similarity measures

Publicación relacionada [1] Lara-Clares A, Lastra-Díaz JJ, Garcia-Serrano A. Protocol for a reproducible experimental survey on biomedical sentence similarity. PLoS One. 2021;16: e0248663. doi: 10.1371/journal.pone.0248663

Notas The related software for executing the experiments is available on Github (https://github.com/jjlastra/HESML). This work was supported by the UNED predoctoral grant started in April 2019 (BICI n7, 19th November 2018).

Licencia/Acuerdo de uso de los datos CC-BY-NC-SA-4.0

1 a 10 de 10 Ficheros. Seleccionando varios ficheros no se pueden descargar más de 10 GB.

BERTExperiments.tar.gz
Archivo Gzip - 20,2 GB
Publicado 8 nov. 2021
7 Descargas
MD5: 25f...c46
This file contains all the BERT models and dependencies evaluated in HESML V2R1 as detailed in [1]. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/BERTExperiments.tar.gz

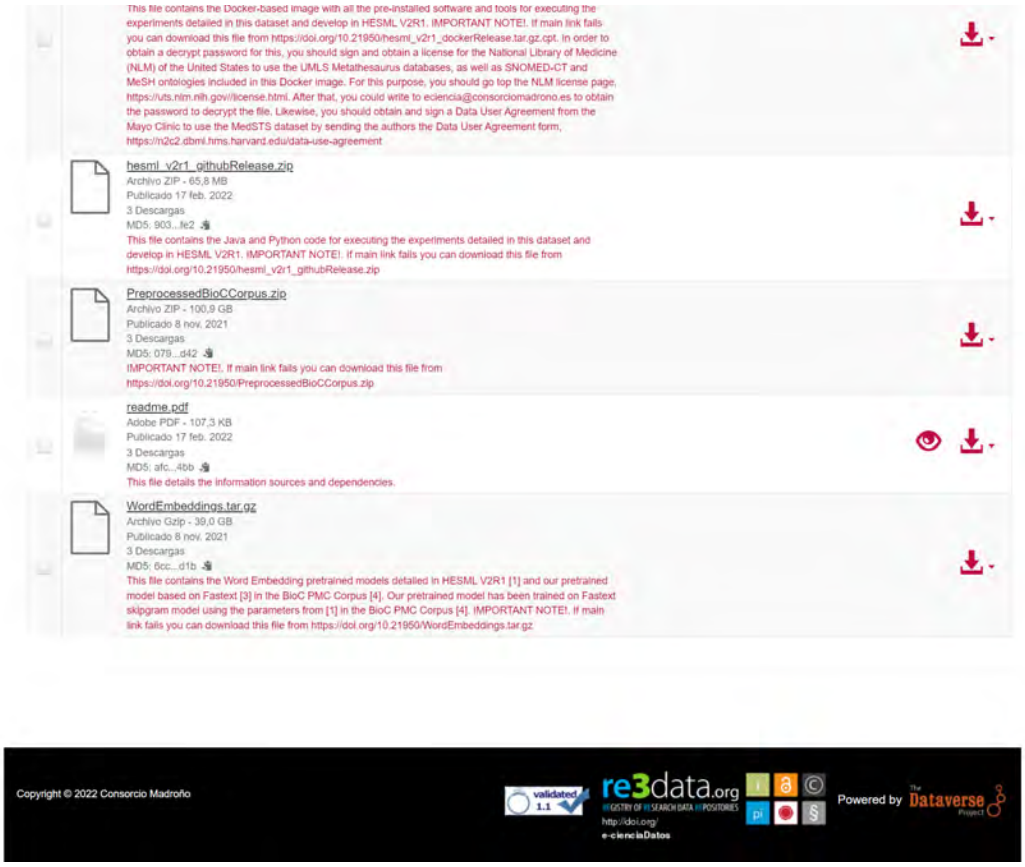
BioCManuscriptCorpus.zip
Archivo ZIP - 42,8 GB
Publicado 8 nov. 2021
5 Descargas
MD5: 6c8...ff8
This file contains the BioC XML files in unicode format from PMC Corpus The dataset has been downloaded from 1 on 5 June, 2019. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/BioCManuscriptCorpus.zip
Datos

BioSentenceSimRawOutput.tar.gz
Archivo Gzip - 3,8 MB
Publicado 17 feb. 2022
6 Descargas
MD5: cf5...e75
This file contains the raw output files of a complete execution of the software clients HESMLSTSImpactPreprocessingclient and HESMLSTSclient.

CharacterAndSentenceEmbeddings.tar.gz
Archivo Gzip - 19,4 GB
Publicado 8 nov. 2021
4 Descargas
MD5: a27...480
This file contains all the character and sentence pretrained models evaluated in HESML V2R1 as detailed in [1]. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/CharacterAndSentenceEmbeddings.tar.gz

Dependencies.tar.gz.cpt
application/mac-compactpro - 10,4 GB
Publicado 8 nov. 2021
3 Descargas
MD5: 3c2...2d0
This file contains all the dependencies and external sources for executing the experiments in [1]. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/Dependencies.tar.gz.cpt. In order to obtain a decrypt password for this, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file. Likewise, you should obtain and sign a Data User Agreement from the Mayo Clinic to use the MedSTS dataset by sending the authors the Data User Agreement form, https://n2c2.dbmi.hms.harvard.edu/data-use-agreement

hesml_v2r1_dockerRelease.tar.gz.cpt
application/mac-compactpro - 6,3 GB
Publicado 8 nov. 2021
0 Descargas
MD5: 053...cef



This file contains the Docker-based image with all the pre-installed software and tools for executing the experiments detailed in this dataset and develop in HESML V2R1. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/hesml_v2r1_dockerRelease.tar.gz.cpt. In order to obtain a decrypt password for this, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file. Likewise, you should obtain and sign a Data User Agreement from the Mayo Clinic to use the MedSTS dataset by sending the authors the Data User Agreement form, https://n2c2.dbmi.hms.harvard.edu/data-use-agreement

hesml_v2r1_githubRelease.zip
Archivo ZIP - 65,8 MB
Publicado 17 feb. 2022
3 Descargas
MD5: 903...fe2

This file contains the Java and Python code for executing the experiments detailed in this dataset and develop in HESML V2R1. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/hesml_v2r1_githubRelease.zip

PreprocessedBioCCorpus.zip
Archivo ZIP - 100,9 GB
Publicado 8 nov. 2021
3 Descargas
MD5: 079...d42

IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/PreprocessedBioCCorpus.zip

readme.pdf
Adobe PDF - 107,3 KB
Publicado 17 feb. 2022
3 Descargas
MD5: afc...4bb

This file details the information sources and dependencies.

WordEmbeddings.tar.gz
Archivo Gzip - 39,0 GB
Publicado 8 nov. 2021
3 Descargas
MD5: 6cc...d1b

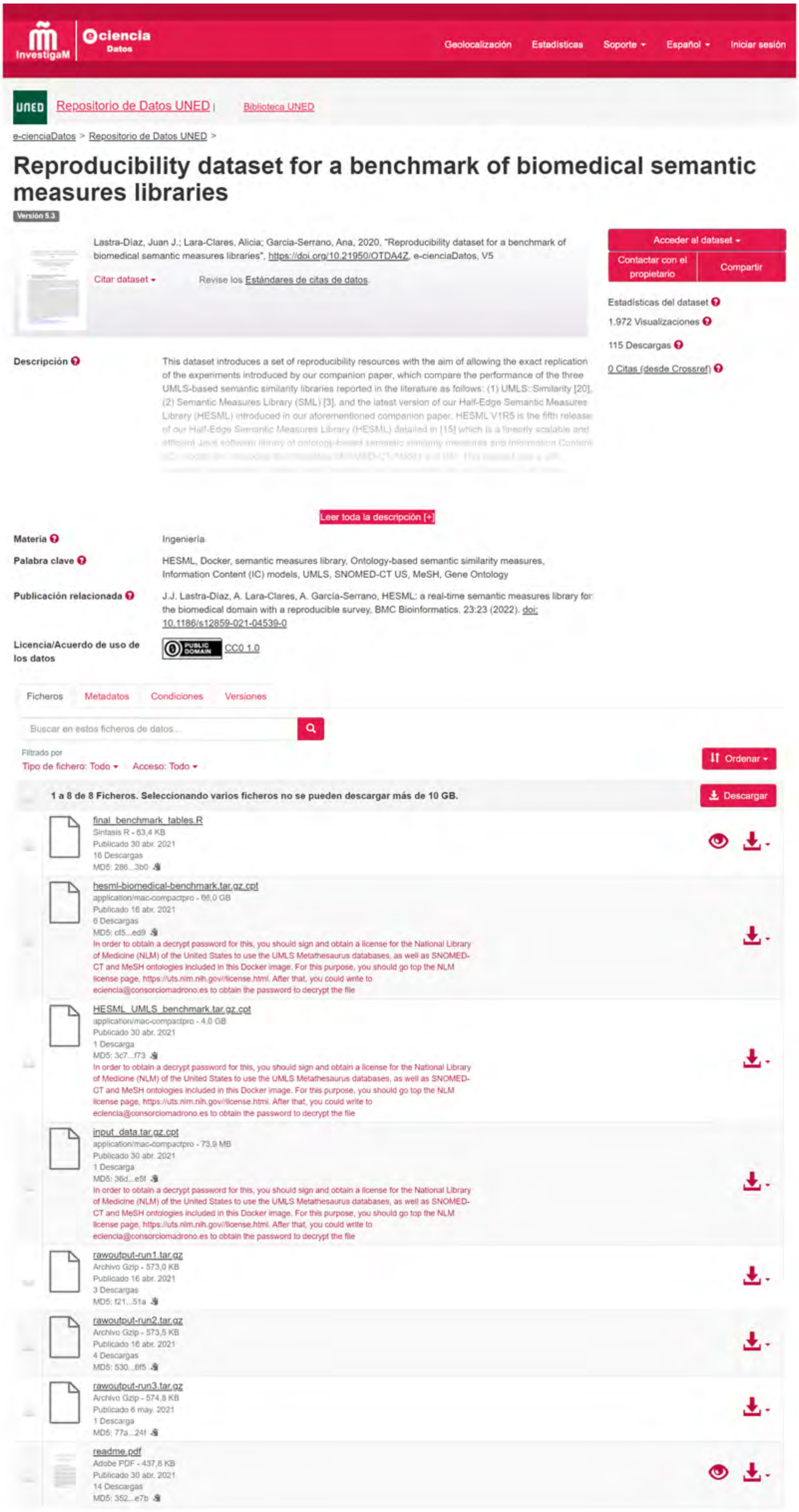
This file contains the Word Embedding pretrained models detailed in HESML V2R1 [1] and our pretrained model based on Fastext [3] in the BioC PMC Corpus [4]. Our pretrained model has been trained on Fastext skipgram model using the parameters from [1] in the BioC PMC Corpus [4]. IMPORTANT NOTE!. If main link fails you can download this file from https://doi.org/10.21950/WordEmbeddings.tar.gz

Copyright © 2022 Consorcio Madroño

validated 1.1
re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES
http://doi.org/
e-cienciaDatos
Powered by The Dataverse Project

Chapter 17

Benchmarks between Semantic Sentence Similarity Measures



Repositorio de Datos UNED | Biblioteca UNED

e-cienciaDatos > Repositorio de Datos UNED >

Reproducibility dataset for a benchmark of biomedical semantic measures libraries

Versión 5.3

Lastra-Díaz, Juan J.; Lara-Clares, Alicia; Garcia-Serrano, Ana, 2020, "Reproducibility dataset for a benchmark of biomedical semantic measures libraries", https://doi.org/10.21950/OTDA4Z, e-cienciaDatos, V5

Citar dataset ▾ Revise los Estándares de citas de datos.

Estadísticas del dataset

1.972 Visualizaciones

115 Descargas

0 Citas (desde Crossref)

Descripción This dataset introduces a set of reproducibility resources with the aim of allowing the exact replication of the experiments introduced by our companion paper, which compare the performance of the three UMLS-based semantic similarity libraries reported in the literature as follows: (1) UMLS::Similarity [20], (2) Semantic Measures Library (SML) [3], and the latest version of our Half-Edge Semantic Measures Library (HESML) introduced in our aforementioned companion paper. HESML V1R5 is the fifth release of our Half-Edge Semantic Measures Library (HESML) detailed in [15] which is a linearly scalable and efficient Java software library of ontology-based semantic similarity measures and Information Content [illegible]

Materia Ingeniería

Palabra clave HESML, Docker, semantic measures library, Ontology-based semantic similarity measures, Information Content (IC) models, UMLS, SNOMED-CT US, MeSH, Gene Ontology

Publicación relacionada J.J. Lastra-Díaz, A. Lara-Clares, A. García-Serrano, HESML: a real-time semantic measures library for the biomedical domain with a reproducible survey. BMC Bioinformatics. 23:23 (2022). doi: 10.1186/s12859-021-04539-0

Licencia/Acuerdo de uso de los datos PUBLIC DOMAIN CC0 1.0

1 a 8 de 8 Ficheros. Seleccionando varios ficheros no se pueden descargar más de 10 GB.

final_benchmark_tables.R
Sintasis R - 63,4 KB
Publicado 30 abr. 2021
16 Descargas
MD5: 286...3b0

hesml-biomedical-benchmark.tar.gz.cpt
application/mac-compactpro - 66,0 GB
Publicado 16 abr. 2021
6 Descargas
MD5: cf5...ed9
In order to obtain a decrypt password for this, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file

HESML_UMLS_benchmark.tar.gz.cpt
application/mac-compactpro - 4,0 GB
Publicado 30 abr. 2021
1 Descarga
MD5: 3c7...f73
In order to obtain a decrypt password for this, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file

input_data.tar.gz.cpt
application/mac-compactpro - 73,9 MB
Publicado 30 abr. 2021
1 Descarga
MD5: 36d...e5f
In order to obtain a decrypt password for this, you should sign and obtain a license for the National Library of Medicine (NLM) of the United States to use the UMLS Metathesaurus databases, as well as SNOMED-CT and MeSH ontologies included in this Docker image. For this purpose, you should go top the NLM license page, https://uts.nlm.nih.gov//license.html. After that, you could write to eciencia@consorciomadrono.es to obtain the password to decrypt the file

rawoutput-run1.tar.gz
Archivo Gzip - 573,0 KB
Publicado 16 abr. 2021
3 Descargas
MD5: f21...51a

rawoutput-run2.tar.gz
Archivo Gzip - 573,5 KB
Publicado 16 abr. 2021
4 Descargas
MD5: 530...6f5

rawoutput-run3.tar.gz
Archivo Gzip - 574,8 KB
Publicado 6 may. 2021
1 Descarga
MD5: 77a...24f

readme.pdf
Adobe PDF - 437,8 KB
Publicado 30 abr. 2021
14 Descargas
MD5: 352...e7b

validated 1.1
re3data.org
REGISTRY OF RESEARCH DATA REPOSITORIES
http://doi.org/
e-cienciaDatos
Powered by The Dataverse Project