UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

WEB PEOPLE SEARCH

TESIS DOCTORAL

Javier Artiles Picón
Licenciado en Lingüística Computacional
2009

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

WEB PEOPLE SEARCH

Javier Artiles Picón
Licenciado en Lingüística Computacional por la Universidad Complutense de Madrid

Directores:

Julio Gonzalo Arroyo

Profesor Titular de Universidad del Departamento de Lenguajes y Sistemas Informáticos
de la Universidad Nacional de Educación a Distancia

Enrique Amigó Cabrera

Profesor Ayudante Doctor del Departamento de Lenguajes y Sistemas Informáticos
de la Universidad Nacional de Educación a Distancia

A mis padres, Julio y Meryam;
a mi hermana, Solange;
a mi tío, Jaime

Acknowledgements

This section would exceed the actual length the book if I were to thank my supervisors, Julio Gonzalo and Enrique Amigó, for everything they have done for my thesis. I am deeply indebted to Julio, who put a lot of confidence in me when I first came to UNED and who offered me the unique chance to work in this enthralling field of research. Julio has provided me with patient guidance in the art of scientific research and he has never missed the opportunity to remind me I still have a lot to learn. I am indebted to Enrique for sharing both his tireless curiosity and genuine scientific perseverance with me. His innumerable contributions to the methodology of evaluation have been crucial for the completion of this work. I want to thank both for instilling their passion for research, endless creativity and positive vibes in me. I would never have been able to write this thesis without their help.

I want to express my humble gratitude to Felisa Verdejo, who is responsible for creating and managing such a favorable research environment as the one we enjoy in the PLN group at UNED.

I want to thank Satoshi Sekine for harboring me in his research group at NYU. His deep experience and proficient aid have been indispensable while organising both evaluation campaigns recorded in this thesis.

Thanks to Anselmo Peñas for having supervised and coordinated my first research project. He has always had enough time to listen to me and to offer me his experience as researcher through the years.

I would also like to thank my colleagues from the Languages and Computer Systems Department at UNED. I would like to thank Víctor Peinado, Juan Cigarrán, Fernando López-Ostenero and Víctor Fresno in particular for all the travels, projects and works we have shared together. I could not possibly have found better companions for this journey.

I would like to thank Eduardo Valls for sharing his command of the English Language with me in order to improve the accuracy and style of this text.

Last, but not least, I would most sincerely like to thank my family. I would like to thank my parents, Julio and Meryam, for instiling true critical and independent awareness in me as well as for supporting me through all these years. I would like to thank Solange for having met her obligations as older sister when I most needed them. And, finally, I would like to thank my uncle Jaime for all his encouragement and wholehearted love.

Agradecimientos

Esta sección empequeñecería el resto si escribiera todas las cosas que debo agradecer a mis directores de tesis, Julio Gonzalo y Enrique Amigó. Estoy en deuda con Julio, quien confió en mí cuando llegué por primera vez a la UNED y me ofreció la oportunidad de trabajar en un área apasionante. Julio me ha guiado con paciencia en el arte de la investigación científica y cada día me recuerda lo mucho que me queda por aprender. A Enrique le debo el compartir conmigo su infatigable curiosidad y su constancia de auténtico científico. Sus aportaciones a la metodología de evaluación han sido esenciales para este trabajo. A los dos les agradezco el contagiarme su pasión por la investigación, su creatividad y su energía positiva. Sin su ayuda esta tesis nunca se habría escrito.

Quiero expresar mi gratitud a Felisa Verdejo, quien ha creado y dirige un entorno de investigación tan favorable como el que tenemos en el grupo PLN de la UNED.

Le agradezco a Satoshi Sekine el haberme acogido en su grupo de investigación en la Universidad de Nueva York. Su experiencia y ayuda han sido imprescindibles para la organización de las dos campañas de evaluación que se presentan en este trabajo.

A Anselmo Peñas le agradezco el haber dirigido mi primer trabajo de investigación. Durante todos estos años siempre tuvo tiempo para escucharme y ofrecerme su experiencia como investigador.

Quisiera también agradecer a mis compañeros del Departamento de Lenguajes y Sistemas Informáticos de la UNED por el buen ambiente de trabajo y el conocimiento compartido durante este tiempo. En particular, quiero agradecer su ayuda a Víctor Peinado, Juan Cigarrán, Fernando López-Ostenero y Víctor Fresno con quienes he compartido proyectos y trabajos. No podría haber deseado mejores compañeros para este viaje.

Agradezco muy especialmente a mi familia: a mis padres Julio y Meryam, por enseñarme a pensar con espíritu crítico e independiente y por su apoyo durante todos estos años; a Solange, por haber ejercido de hermana mayor cuando más lo necesitaba; y a mi tío Jaime, por sus palabras de ánimo y su cariño incondicional.

Abstract

In this thesis we have addressed the problem of name ambiguity while searching for people on the Web. At the beginning of our work, in 2004, there were very few research papers on this topic, and no commercial web search engine would provide this type of facility. For this reason, our research methodology initially focused on the design and organisation (together with Prof. Sekine from New York University) of a competitive evaluation campaign for Web People Search systems. Once the campaign had been run for two years, we used the standard test suites built to perform our own empirical studies on the nature and challenges of the task.

The evaluation campaign, WePS, was organized in 2007 (as a SemEval 2007 task) and in 2009 (as a WWW 2009 workshop). WePS was crucial in the process to lay the foundations of a proper scientific study of the Web People Search problem. These were the main accomplishments:

- Standardisation of the problem: now a majority of researchers focus on the problem as a search results mining task (clustering and information extraction), as it has been defined in WePS.

- Creation of standard benchmarks for the task: since the first WePS campaign in 2007, the number of publications related to Web People Search has grown substantially, and most of them use the WePS test suites as a de-facto standard benchmark. As of summer 2009, there were already more than 70 research papers citing WePS overviews; this not only suggests that WePS has indeed become a standard reference for the task, but also that it has contributed to arouse the interest in this kind of research problems.

- Design of evaluation metrics for the task:

  1. We have performed a careful formal analysis of several extrinsic clustering evaluation metrics based on formal constraints, to conclude that BCubed metrics are the most suitable for the task. We have also extended the original BCubed definition to allow for overlapping clusters, which is a practical requirement of the task. Our results are general enough to be employed in other clustering tasks.
  2. We have introduced a new metric combination function, Unanimous Improvement Ratio (UIR), which, unlike Van Rijsbergen's F, does not

require an a-priori weighting of metrics (in our case, BCubed Precision and Recall). In an extensive empirical study we have shown that UIR provides rich information to compare the performance of systems, which was impossible with previous existing metric combinations functions (most prominently F). Using the results of the WePS-2 campaign, we have shown that F and UIR provide complementary information and, altogether, constitute a powerful analytical tool to compare systems. Although we have tested UIR only in the context of our task, it could be potentially useful in any task where several evaluation metrics are needed to capture the quality of a system, as it happens in several Natural Language Processing problems.

Using the test suites produced in the two WePS evaluation campaigns, we have then performed a number of empirical studies in order to enhance a better understanding and comprehension of both the nature of the task involved and the way to solve it:

- First, we have studied the potential effects of using (interactive) query refinements to perform the Web People Search task. We have discovered that, although in most occasions there is an expression that can be used as a near-perfect refinement to retrieve all and only those documents referring to an individual, the nature of these ideal refinements is unpredictable and very unlikely to be hypothesized by the user. This confirms the need for search results clustering, and also suggests that looking for an optimal refinement may be a strategy of automatic systems to accomplish the task (and one that has not been used by any participant in the WePS campaigns).

- Second, we have studied the usefulness of linguistic (computationally intensive) features as compared to word n-grams and other cheap features to solve our clustering problem. Notably, named entities, which are the most popular feature immediately after bag-of-words approaches, does not seem to provide a direct competitive advantage to solve the task. We have reached this conclusion abstracting from a particular choice of Machine Learning and Text Clustering algorithms, by using a Maximal Pairwise Accuracy estimator introduced in this thesis.

- As a side effect of our empirical study, we have built a system which, using the confidence of a binary classifier (whether two pages are coreferent or not) as a similarity metric between document pairs to feed a Hierarchical Agglomerative Clustering algorithm, provides the best results for the task known to us ($F_{0.5} = 0.83$ vs. 0.82 for the best WePS-2 system), without using computationally intensive linguistic features.

Resumen

En esta tesis hemos abordado el problema de la ambigüedad de nombres en la búsqueda de personas en la Web. Al inicio de nuestro trabajo, en 2004, había muy pocos artículos de investigación sobre este tema, y ningún buscador web comercial ofrecía este tipo de servicio. Por esta razón, nuestra metodología de investigación se enfocó inicialmente en el diseño y la organización (junto con el Dr. Satoshi Sekine de la Universidad de Nueva York) de una campaña de evaluación competitiva para sistemas de Búsqueda de Personas en la Web. Tras la celebración de dos campañas de evaluación, utilizamos las colecciones de prueba elaboradas para realizar nuestros propios estudios empíricos sobre la naturaleza y los desafíos de la tarea.

La campaña de evaluación, WePS, tuvo lugar en 2007 (como una tarea de Semeval 2007) y en 2009 (como un *workshop* de la conferencia WWW 2009). Esta campaña fue crucial para sentar las bases para un estudio científico del problema de la Búsqueda de Personas en la Web. Éstos fueron los principales logros:

- Estandarización del problema: ahora la mayoría de investigadores enfocan el problema como una tarea de minería de resultados de busqueda (clustering – agrupación – y extracción de información), tal y como lo definimos en WePS.

- Creación de un estándar para la comparación de sistemas: desde la primera campaña WePS en 2007, el número de publicaciones relacionadas con la Búsqueda de Personas en la Web ha crecido sustancialmente, y la mayoría usa colecciones de prueba desarrolladas en WePS. En el verano de 2009, ya más de 70 artículos de investigación referenciaban la tarea. Ésto no sólo sugiere que WePS se ha convertido en un estándar de referencia para la tarea, sino también que ha contribuido a aumentar el interés en este tema de investigación.

- Diseño de métricas de evaluación para la tarea:

    1. Hemos realizado un cuidadoso análisis, basado en restricciones formales, de varias métricas extrínsecas de evaluación de sistemas de clustering, y hemos concluído que las métricas BCubed son las más adecuadas para la tarea. También hemos extendido la definición original de BCubed para permitir clusters solapados, que es un requisito práctico de la tarea. Nuestros resultados son suficientemente generales como para ser empleados en otras tareas de agrupación.

2. Hemos introducido una nueva función para la combinación de métricas, el *Unanimous Improvement Ratio* (UIR) o Ratio de Mejora Unánime, el cual, al contrario que la función F de Van Rijsbergen, no requiere un pesado *a-priori* de las métricas (en nuestro caso, BCubed Precision y Recall). En un extenso estudio empírico hemos mostrado que UIR proporciona información muy valiosa para la comparación de sistemas, información que no proporcionan las funciones de combinación de métricas existentes (principalmente F). Utilizando los resultados de la campaña WePS-2, hemos mostrado que F y UIR proporcionan informacion complementaria y, en conjunto, constituyen una potente herramienta de análisis para comparar sistemas. Aunque hemos probado UIR sólo en el contexto de nuestra tarea, podría ser útil en cualquier tarea donde se necesiten varias métricas de evaluación para capturar la calidad de los sistemas, como ocurre en muchos problema de Procesamiento del Lenguaje Natural.

Utilizando las colecciones de prueba desarrolladas en las dos campañas de evaluación WePS, hemos realizado una serie de estudios empíricos orientados a obtener una mejor comprensión tanto de la naturaleza de la tarea como de la manera de resolverla:

- En primer lugar, hemos estudiado los efectos potenciales de usar refinamientos de consultas (interactivos) para realizar la tarea de Búsqueda de Personas en la Web. Hemos descubierto que, aunque en la mayoría de las ocasiones existe una expresión que puede ser utilizada como refinamiento casi perfecto para recuperar todos y sólo los documentos que refieren a una persona, la naturaleza de esto refinamientos ideales es impredecible y son muy difíciles de encontrar por un usuario. Esto confirma la necesidad de agrupar los resultados de búsqueda, y también sugiere que buscar un refinamiento óptimo puede ser una estrategia para sistemas que intentan resolver la tarea (una estrategia que aun no ha sido utilizada por los participantes en las campañas WePS).

- En segundo lugar, hemos estudiado la utilidad de los rasgos lingüísticos (computacionalmente costosos) en comparación con n-gramas de palabras y otros rasgos “baratos” para resolver nuestro problema de agrupación. Sorprendentemente, las entidades nombradas, que son son el tipo de rasgo más popular después de las aproximaciones basadas en ”bolsas de palabras”, no parecen aportar una ventaja competitiva directa para resolver la tarea. Hemos alcanzado esta conclusión con independencia de la elección del algoritmo de aprendizaje automático y del algoritmo de clustering usados, mediante el uso del estimador *Maximal Pairwise Precision* (Precisión Máxima de Pares) presentado en esta tesis.

- Como efecto secundario de nuestro estudio empírico, hemos construido un sistema que, utilizando la confianza de un clasificador binario (el cual detecta si dos documentos son o no correferentes) como métrica de similitud entre pares de documentos para alimentar al algoritmo de Agrupación Aglomerativa Jerárquica, aporta los mejores resultados para la tarea que conocemos (($F_{0,5} = 0,83$ frente a 0,82 del mejor sistema en WePS-2), sin utilizar rasgos lingüísticos computacionalmente costosos.

Contents

Contents 17

List of Figures 21

List of Tables 23

I Background 25

1 Introduction 27

1.1 The Ambiguity of People Names on the Web 29

1.2 Web People Search Services 30

1.3 Web People Search and Other Related NLP Tasks 33

1.4 Goals 35

1.5 How we Addressed the Web People Search Problem 36

1.6 Structure of the Thesis 37

2 State of the Art 41

2.1 Test Collections 41

2.1.1 Manually Annotated Test Collections 42

2.1.2 Pseudo-ambiguity Test Collections 46

2.2 Document Representation 50

2.3 Similarity Metrics 51

2.4 Clustering Methods 51

2.4.1 Clustering Process 52

2.4.2 Stopping Criteria 54

2.4.3 Linkage 54

2.5 Evaluation Metrics 56

II Benchmarking 59

3 The WePS-1 Campaign 61

3.1 Preliminary Testbed 61

3.2 WePS-1 Testbed 63

3.2.1 Training Data 64

3.2.2 Test Data 66
3.2.3 Annotation Process 66
3.3 Evaluation Methodology 68
3.4 Campaign Design 69
3.5 Results of the Evaluation Campaign 70
3.6 Conclusions 71

4 Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints 73
4.1 Formal Constraints on Evaluation Metrics for Clustering Tasks . . 74
4.1.1 Constraint 1: Cluster Homogeneity 74
4.1.2 Constraint 2: Cluster Completeness 75
4.1.3 Constraint 3: Rag Bag 75
4.1.4 Constraint 4: Clusters Size vs. Quantity 76
4.2 Comparison of Evaluation Metrics 77
4.2.1 Evaluation by Set Matching 77
4.2.2 Metrics Based on Counting Pairs 78
4.2.3 Metrics based on entropy 80
4.2.4 Evaluation Metrics Based on Edit Distance 81
4.2.5 BCubed: a Mixed Family of Metrics 81
4.3 Related Work: Other Proposed Formal Constraints 84
4.3.1 Dom's Constraints 84
4.3.2 Meila's Constraints 86
4.4 Evaluation of Overlapping Clustering 86
4.4.1 Extending Standard Metrics for Overlapping Clustering . 87
4.4.2 Extending BCubed Metrics 87
4.4.3 Extended BCubed: Example of Usage 90
4.4.4 Extended BCubed Applied to WePS-1 93
4.5 Conclusions 95

5 Combining Evaluation Metrics via the Unanimous Improvement Ratio 97
5.1 The Effects of Metric Weighting in Clustering Tasks 98
5.2 Proposal 101
5.2.1 Unanimous Improvements 101
5.2.2 Unanimous Improvement Ratio 102
5.3 Experiments 103
5.3.1 Rewarding Robustness Across α Values 103
5.3.2 Reflecting Improvement Ranges 105
5.3.3 UIR Threshold 105
5.4 Conclusions 107

6 The WePS-2 Campaign 109
6.1 Testbed 109
6.1.1 Development Data 109
6.1.2 Test Data 110

6.1.3 Manual Annotation 111
6.2 Evaluation Methodology 113
6.2.1 Campaign Design 114
6.3 Participants 114
6.4 Results of the Evaluation 115
6.4.1 Comparison of Systems 116
6.4.2 Robustness of F Results with Different α Values 117
6.5 Conclusions 120

III Empirical Studies 121

7 The Scope of Query Refinement in the WePS Task 123
7.1 Dataset 123
7.2 Experiments 124
7.3 Results 127
7.4 Conclusions 129

8 The Role of Named Entities in WePS 131
8.1 Experimental Settings 132
8.1.1 Features 132
8.1.2 Reformulating WePS as a Classification Task 133
8.2 Analysis of Individual Features 134
8.2.1 Feature Performance 134
8.2.2 Redundancy 135
8.3 Analysis of Feature Combinations 137
8.4 Results on the Clustering Task 139
8.5 Conclusions 141

9 Conclusions 143
9.1 Contributions 143
9.1.1 The Need for Name Disambiguation Systems 143
9.1.2 Development of Reference Test collections 144
9.1.3 Development of Improved Clustering Evaluation Metrics . 145
9.1.4 The Relevance of the Clustering Stopping Criterion 145
9.1.5 The Role of Named Entities 146
9.1.6 Testbeds and Software Released 147
9.2 Further Directions 147

Bibliography 149

A Publications 159

List of Figures

1.1 Search results for a name shared by many people 28
1.2 Celebrities sharing the name “Michael Moore” according to Wikipedia 30
1.3 Zoominfo: sample person profile 31
1.4 Zoominfo: error disambiguating mentions of the same person . . . 31
1.5 Name disambiguation error in Spock 32
1.6 Name disambiguation error in ArnetMiner 32
1.7 Thesis iterative process 37

2.1 Example of name conflation for generating a pseudo-ambiguous name 46
2.2 A dendrogram of a clustering of 10 web search results for the query “John Smith” 53
2.3 Single-link: maximum similarity 55
2.4 Complete-link: minimum similarity 55
2.5 Group-average: average of all similarities 55
2.6 Centroid: average of inter-similarity 55
2.7 Example of a cross-document coreference chain 57

3.1 WePS-1 baseline systems 69

4.1 Clustering evaluation metric constraint 1: Cluster Homogeneity . 75
4.2 Clustering evaluation metric constraint 2: Cluster Completeness . 75
4.3 Clustering evaluation metric constraint 3: Rag Bag 76
4.4 Clustering evaluation metric constraint 4: Size vs. Quantity 77
4.5 Clustering constraint 2 example 78
4.6 Satisfaction of Formal Constraints: Examples 79
4.7 Example of computing the BCubed precision and recall for one item 82
4.8 More noise clusters implies less quality 85
4.9 Clustering evaluation metric constraint 5: Item Multiplicity 88
4.10 Computing the multiplicity recall and precision between two items for extended BCubed metrics 89
4.11 BCubed computing example 1 (ideal solution): Precision=1 Recall=1 90
4.12 BCubed computing example 2 (duplicating clusters): Precision=0.86 Recall=1 90
4.13 BCubed computing example 3 (duplicating clusters): Precision=0.81 Recall=1 91

4.14 BCubed computing example 4 (removing item occurrences): Precision=1 Recall=0.68 91
4.15 BCubed computing example 5 (splitting clusters): Precision=1 Recall=0.74 91
4.16 BCubed computing example 6 (joining clusters): Precision=0.88 Recall=0.94 91
4.17 F average over BCubed Precision and Recall satisfies all formal constraints: $F(P,R) = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$ 92
4.18 Clustering output of a cheat system 93

5.1 Evaluation results for clustering systems in WePS-1 98
5.2 WePS-1 system evaluation results for several F parametrisations . 99
5.3 F measure vs. UIR: rewarding robustness 102
5.4 F vs. UIR: reflecting improvement ranges 104
5.5 UIR vs. the improvement according to the less improved metric. . 105
5.6 Improvement detected across UIR thresholds 106

6.1 WePS-2 clustering annotation GUI 113

8.1 Precision/Recall curve of token-based features 135
8.2 Precision/Recall curve of word n-grams 136
8.3 Precision/Recall curve of NEs obtained with the Stanford NER tool 137
8.4 Precision/Recall curve of NEs obtained with the OAK NER tool . 138
8.5 Independence of similarity criteria with respect to the token based feature 139
8.6 Estimated PWA upper bound versus the real PWA of decision trees trained with feature combinations 140
8.7 Maximal Pairwise Accuracy vs. results of a Decision Tree 141

List of Tables

2.1 Example of concept-instance pairs 43
2.2 Summary of manually annotated test collections 45
2.3 Summary of pseudo-ambiguous test collections 49

3.1 WePS-1 preliminary testbed: annotation statistics 62
3.2 WePS-1 trial testbed: number of people for each name 63
3.3 WePS-1 training data 65
3.4 WePS-1 test data 67
3.5 WePS-1 team ranking 71

4.1 WePS-1 system ranking according to Purity, Inverse Purity and F(Purity, Inverse Purity) 94
4.2 WePS-1 system ranking according to Extended BCubed Precision, Extended BCubed Recall, and its F combination. 94

5.1 WePS-1 rankings for $F_{0.5}$ and $F_{0.2}$ using Purity and Inverse Purity 100
5.2 Statistical significance of improvements: $F_{0.5}$ vs. $F0.2$ 100
5.3 UIR and $F_{0.5}$ increase when F increases for all α values 104
5.4 UIR and $F_{0.5}$ increases vs. statistical significance tests 104

6.1 WePS-2 test data 111
6.2 WePS-2 official team ranking using B-Cubed measures 116
6.3 WePS-2 team ranking using Purity/Inverse Purity metrics 118
6.4 WePS-2 results with BCubed Precision and Recall, F and UIR measures. 119

7.1 Query refinement results for clusters of size 1 124
7.2 Query refinement results for clusters of size 2 125
7.3 Query refinement results for clusters of size >=3 126
7.4 Query refinement results for clusters of size 1 126
7.5 Query refinement results for clusters of size 2 126
7.6 Query refinement results for clusters of size >=3 127
7.7 Distribution of the person attributes used for the ”best-ae“ query refinement strategy 127

8.1 Distribution of instances in the WePS classification testbed 134

8.2 Evaluation of Decision Tree models on the WePS-2 clustering task 141

Part I

Background

Chapter 1

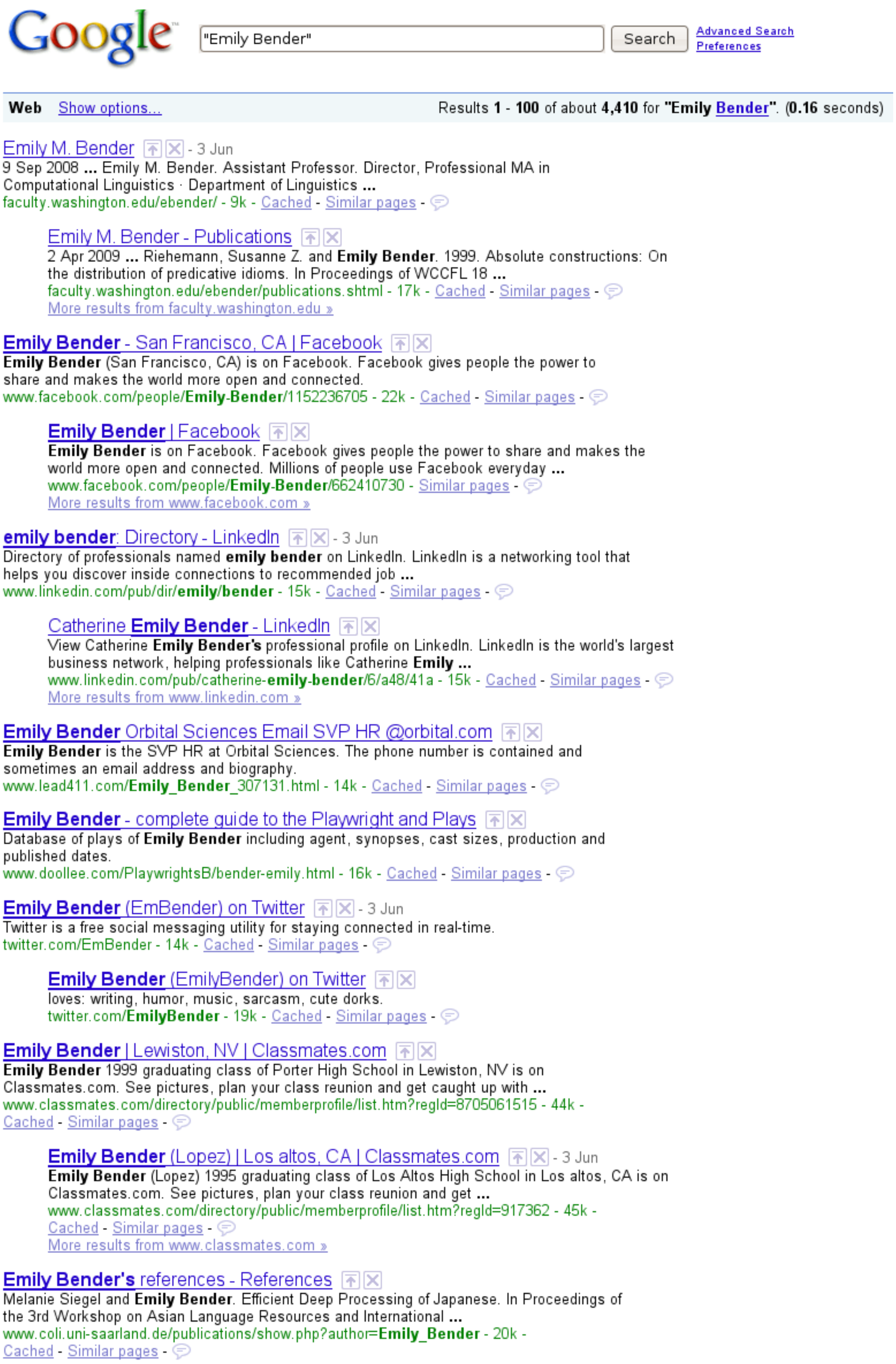
Introduction

Let us suppose that an Information Technology company has opened a project manager position and hundreds of people submit their resumé. The position needs to be filled by a reliable worker, and only few candidates can be interviewed. In view of this circumstance, the company decides to supplement the resumé with personal information available on the Web.

The employee in charge of collecting all candidates' data uses a popular web search engine, querying for each person's name and browsing the first ten pages of search results (100 documents) for documents with information regarding that particular candidate. Unfortunately, many documents refer to other people who share the same name.

Thus, the employee is forced to read all documents to make sure he is not missing anything. At the end of the day, only a small fraction of the pages he has read was relevant and a lot of time has been lost deciding which pages actually refer to the candidate. The problem is that standard web search services do not take the ambiguity of person name queries into account, and consequently the ranked list of documents contains a bias towards the most popular people, making the access to documents about other people quite difficult. For instance, in Figure 1.1 we show the top web search results for a candidate's name ("Emily Bender"), who is an assistant professor at the University of Washington. Although the first two results (unfolded from the same domain) are relevant, the next documents are quite hard to assess. There are results from social networks like Facebook and LinkedIn, but it is not clear whether they refer to the same person or not. Actually, the last result in the page does refer to the job candidate, but this is only recognisable if one infers that it is an academic reference of a related research field to the one mentioned in the first relevant result, as there is no clear reference to the person affiliation either.

Now, imagine the next day: our employee learns about a new type of web search engines which are specialised on "people search". He might find a search service like *spock.com* or *zoominfo.com* and try searching for a candidate's name. This time, the results comprise a list of different people sharing the same name and the documents associated to each one of them. Even some biographical details, like the affiliation and location, are displayed on each result, making it easier to spot the person he is looking for. This kind of search results, where documents are grouped



Results **1** - **100** of about **4,410** for **"Emily Bender"**. (**0.16** seconds)

Emily M. Bender - 3 Jun
9 Sep 2008 **...** Emily M. Bender. Assistant Professor. Director, Professional MA in Computational Linguistics · Department of Linguistics **...**
faculty.washington.edu/ebender/ - 9k - Cached - Similar pages -

Emily M. Bender - Publications
2 Apr 2009 **...** Riehemann, Susanne Z. and **Emily Bender**. 1999. Absolute constructions: On the distribution of predicative idioms. In Proceedings of WCCFL 18 **...**
faculty.washington.edu/ebender/publications.shtml - 17k - Cached - Similar pages -
More results from faculty.washington.edu »

Emily Bender - San Francisco, CA | Facebook
Emily Bender (San Francisco, CA) is on Facebook. Facebook gives people the power to share and makes the world more open and connected.
www.facebook.com/people/**Emily-Bender**/1152236705 - 22k - Cached - Similar pages -

Emily Bender | Facebook
Emily Bender is on Facebook. Facebook gives people the power to share and makes the world more open and connected. Millions of people use Facebook everyday **...**
www.facebook.com/people/**Emily-Bender**/662410730 - Similar pages -
More results from www.facebook.com »

emily bender: Directory - LinkedIn - 3 Jun
Directory of professionals named **emily bender** on LinkedIn. LinkedIn is a networking tool that helps you discover inside connections to recommended job **...**
www.linkedin.com/pub/dir/**emily**/**bender** - 15k - Cached - Similar pages -

Catherine **Emily Bender** - LinkedIn
View Catherine **Emily Bender's** professional profile on LinkedIn. LinkedIn is the world's largest business network, helping professionals like Catherine **Emily ...**
www.linkedin.com/pub/catherine-**emily-bender**/6/a48/41a - 15k - Cached - Similar pages -
More results from www.linkedin.com »

Emily Bender Orbital Sciences Email SVP HR @orbital.com
Emily Bender is the SVP HR at Orbital Sciences. The phone number is contained and sometimes an email address and biography.
www.lead411.com/**Emily_Bender**_307131.html - 14k - Cached - Similar pages -

Emily Bender - complete guide to the Playwright and Plays
Database of plays of **Emily Bender** including agent, synopses, cast sizes, production and published dates.
www.doollee.com/PlaywrightsB/bender-emily.html - 16k - Cached - Similar pages -

Emily Bender (EmBender) on Twitter - 3 Jun
Twitter is a free social messaging utility for staying connected in real-time.
twitter.com/EmBender - 14k - Cached - Similar pages -

Emily Bender (EmilyBender) on Twitter
loves: writing, humor, music, sarcasm, cute dorks.
twitter.com/**EmilyBender** - 19k - Cached - Similar pages -

Emily Bender | Lewiston, NV | Classmates.com
Emily Bender 1999 graduating class of Porter High School in Lewiston, NV is on Classmates.com. See pictures, plan your class reunion and get caught up with **...**
www.classmates.com/directory/public/memberprofile/list.htm?regId=8705061515 - 44k - Cached - Similar pages -

Emily Bender (Lopez) | Los altos, CA | Classmates.com - 3 Jun
Emily Bender (Lopez) 1995 graduating class of Los Altos High School in Los altos, CA is on Classmates.com. See pictures, plan your class reunion and get **...**
www.classmates.com/directory/public/memberprofile/list.htm?regId=917362 - 45k - Cached - Similar pages -
More results from www.classmates.com »

Emily Bender's references - References
Melanie Siegel and **Emily Bender**. Efficient Deep Processing of Japanese. In Proceedings of the 3rd Workshop on Asian Language Resources and International **...**
www.coli.uni-saarland.de/publications/show.php?author=**Emily_Bender** - 20k - Cached - Similar pages -

Figure 1.1: Search results for a name shared by many people

according to the person they refer to, are achieved by automatic methods and help our employee to finish, quite quickly, a task that otherwise would take many hours to a human. We will call *Web People Search* the task of grouping and mining search results for a person name according to the individuals sharing that name.

In this thesis, we formalise the Web People Search task (WePS) and study its relationship with the previous work in Natural Language Processing (NLP). We develop an evaluation framework for the empirical study of this task and apply it on competitive evaluation campaigns in order to compare approaches from different research groups. This framework will allow us to justify empirically the need for automatic methods in this task, test different evaluation metrics, compare systems and evaluate the impact of different document representations.

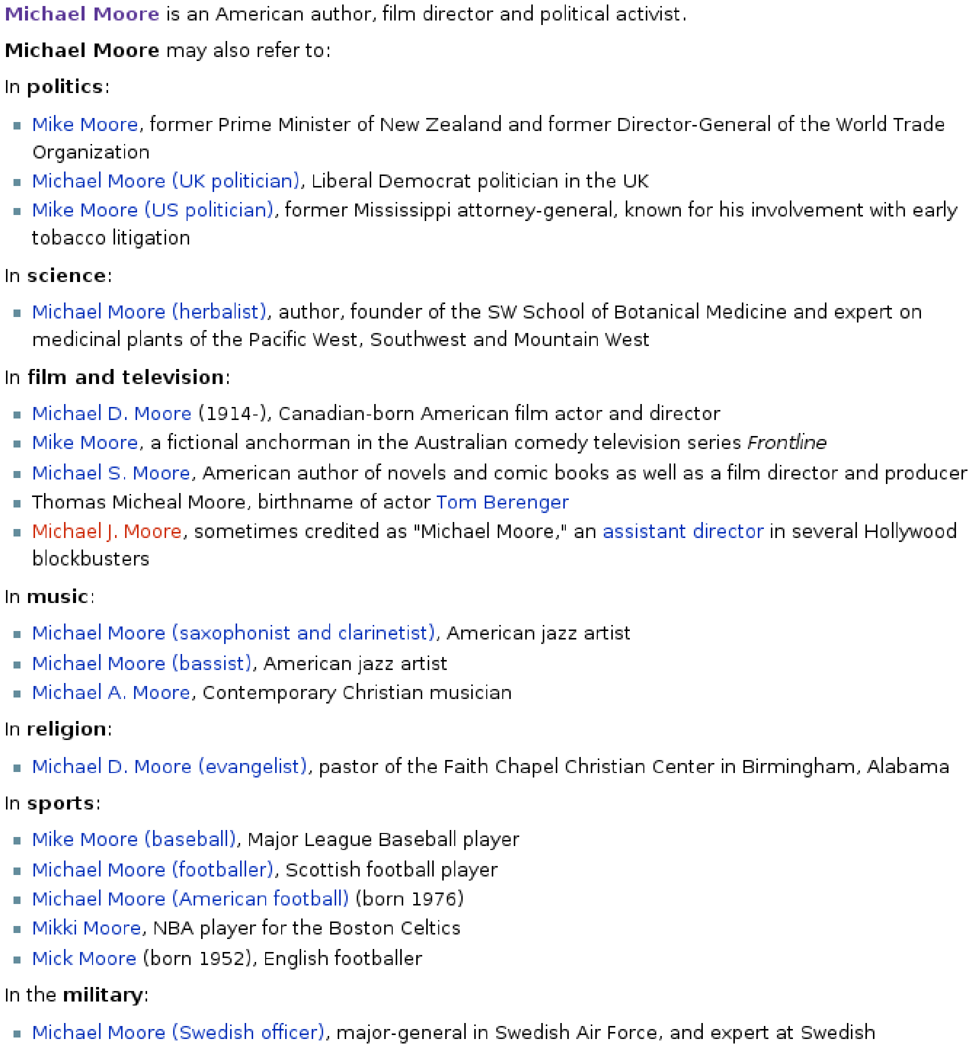
1.1 The Ambiguity of People Names on the Web

A study of the query log of the AllTheWeb and Altavista search sites [SJP04] gives an idea of the relevance of the people search task: 11-17% of the queries were composed of a person name with additional terms and 4% were identified simply as people names. All in all, three in four users were seeking information on non-celebrities.

In addition, ambiguity represents a characteristic feature of most people's names. According to the data available from 1990 U.S. Census Bureau, only 90,000 different names were shared by 100 million people [AGV05]. As the amount of information available in the WWW grows, a higher number of people is mentioned in different web pages. This means that a search for a person name will most likely return a large amount of documents, mentioning different people with the same name.

Different ambiguity scenarios can be found on web search results. On the one hand, the list of search results can contain many different people. For instance, in search results for "Emily Bender" (Figure 1.1) the top 13 documents contain roughly 8 different people which are difficult to discriminate and have a sparse distribution in the documents ranking. In view of these factors, retrieving all the available information for a particular person becomes quite a difficult task. On the other hand, a search for a name shared by a celebrity can make it difficult to retrieve information about less popular people. In a search for "Sharon Goldwater", 165 results are monopolised by a recognised researcher, while a music critic is relegated to the position 166 in the ranking. Even common names can be shared by many celebrities or historical figures at the same time. In those cases, large amounts of information will be available for each individual, but they will also be spread across search results for many different people. In Figure 1.2 we show a page of the online encyclopedia Wikipedia which features 20 celebrities or historical figures with the name "Michael Moore". It is even more troubling that many of these celebrities share similar occupations (three football players, three politicians, etc).

These characteristics of people names, in conjunction with the ranking visualisation in the main web search engines, delegates in the user the burden of finding the pages relevant to the particular person he is interested in. The user might refine the original query with additional terms, but this usually implies filtering out relevant documents in the process. Surprisingly, in Spink's study [SJP04], it was noted that few person name searches included query reformulation.



Michael Moore is an American author, film director and political activist.

Michael Moore may also refer to:

In **politics**:

- Mike Moore, former Prime Minister of New Zealand and former Director-General of the World Trade Organization
- Michael Moore (UK politician), Liberal Democrat politician in the UK
- Mike Moore (US politician), former Mississippi attorney-general, known for his involvement with early tobacco litigation

In **science**:

- Michael Moore (herbalist), author, founder of the SW School of Botanical Medicine and expert on medicinal plants of the Pacific West, Southwest and Mountain West

In **film and television**:

- Michael D. Moore (1914-), Canadian-born American film actor and director
- Mike Moore, a fictional anchorman in the Australian comedy television series *Frontline*
- Michael S. Moore, American author of novels and comic books as well as a film director and producer
- Thomas Micheal Moore, birthname of actor Tom Berenger
- Michael J. Moore, sometimes credited as "Michael Moore," an assistant director in several Hollywood blockbusters

In **music**:

- Michael Moore (saxophonist and clarinetist), American jazz artist
- Michael Moore (bassist), American jazz artist
- Michael A. Moore, Contemporary Christian musician

In **religion**:

- Michael D. Moore (evangelist), pastor of the Faith Chapel Christian Center in Birmingham, Alabama

In **sports**:

- Mike Moore (baseball), Major League Baseball player
- Michael Moore (footballer), Scottish football player
- Michael Moore (American football) (born 1976)
- Mikki Moore, NBA player for the Boston Celtics
- Mick Moore (born 1952), English footballer

In the **military**:

- Michael Moore (Swedish officer), major-general in Swedish Air Force, and expert at Swedish

Figure 1.2: Celebrities sharing the name “Michael Moore” according to Wikipedia

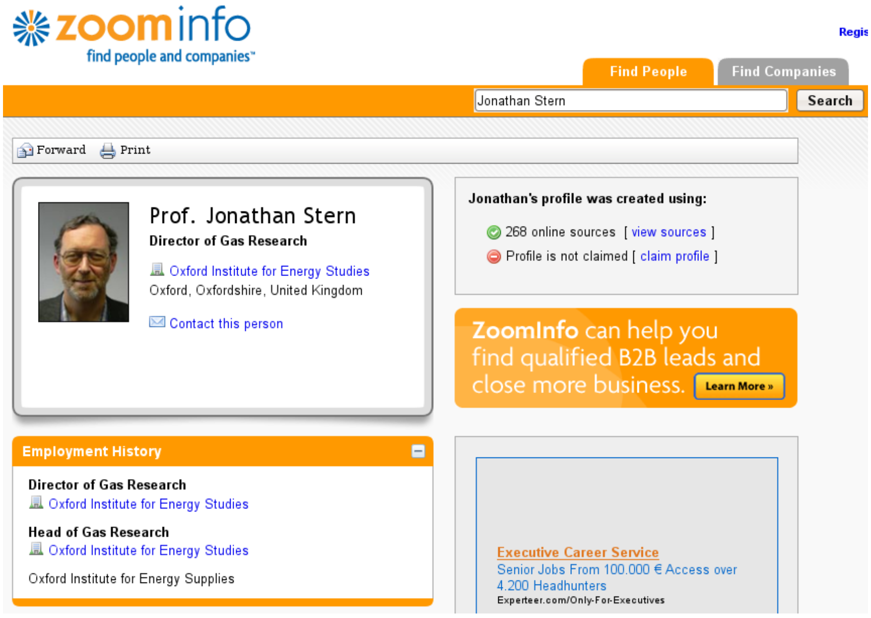
1.2 Web People Search Services

At the time of starting this research (late 2004) there were no people search services online searching on web data. The situation has changed dramatically since then.

The ambiguity of people names has recently become both an active research topic and a relevant application domain for web search services. Zoominfo.com, Spock.com, ArnetMiner, 123people.com are but a few examples of sites which perform Web People Search, albeit with limited disambiguation capabilities. The following examples will show that, although an important effort has been recently made by most Information Retrieval companies, name disambiguation in web search results is still an unsolved problem.

ZoomInfo

In 2005, ZoomInfo was launched as a Web People Search service. In broad terms this service focuses on business related people[1]. From the standpoint of our research, Zoominfo can be considered the first people search commercial initiative. Submitting a query to Zoominfo results in a list of people profiles, each one (fig. 1.3) containing information extracted from various web pages in which the person is mentioned.



Jonathan Stern

Prof. Jonathan Stern
Director of Gas Research
Oxford Institute for Energy Studies
Oxford, Oxfordshire, United Kingdom
Contact this person

Jonathan's profile was created using:
268 online sources [view sources]
Profile is not claimed [claim profile]

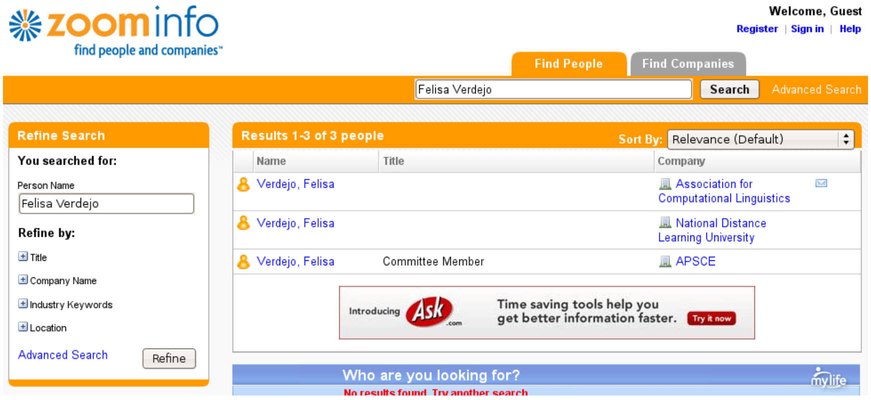
Employment History

Director of Gas Research
Oxford Institute for Energy Studies

Head of Gas Research
Oxford Institute for Energy Studies

Oxford Institute for Energy Supplies

Figure 1.3: Zoominfo: sample person profile



Felisa Verdejo

You searched for:
Person Name
Felisa Verdejo

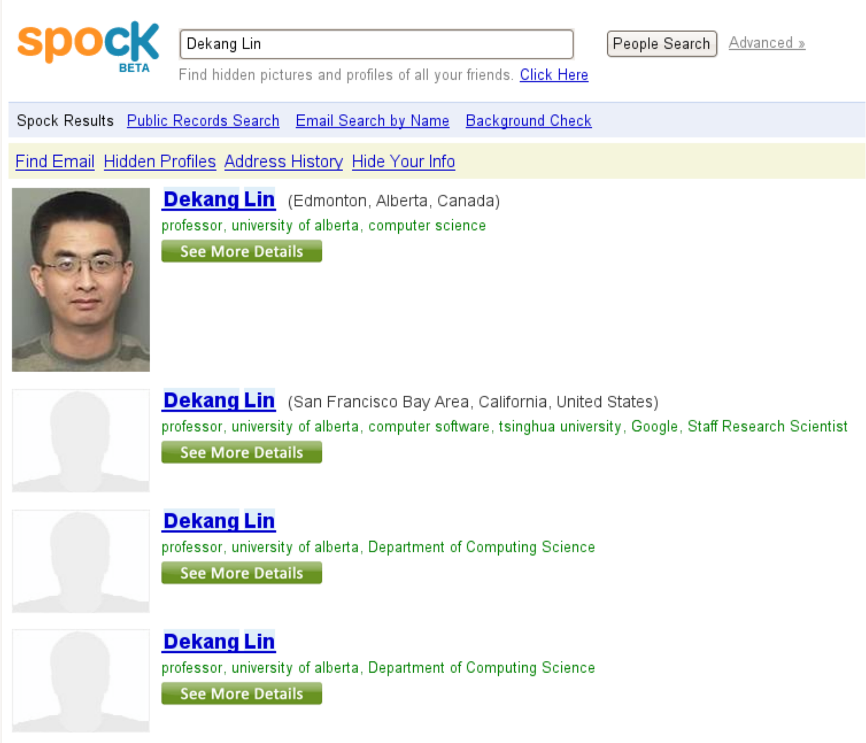
Refine by:
⊞ Title
⊞ Company Name
⊞ Industry Keywords
⊞ Location

Results 1-3 of 3 people   Sort By: Relevance (Default)

Name	Title	Company
Verdejo, Felisa		Association for Computational Linguistics
Verdejo, Felisa		National Distance Learning University
Verdejo, Felisa	Committee Member	APSCE

Figure 1.4: Zoominfo: error disambiguating mentions of the same person

[1] Zoominfo also powers people searches for Business Week.



Dekang Lin

People Search | Advanced »

Find hidden pictures and profiles of all your friends. Click Here

Spock Results | Public Records Search | Email Search by Name | Background Check

Find Email | Hidden Profiles | Address History | Hide Your Info

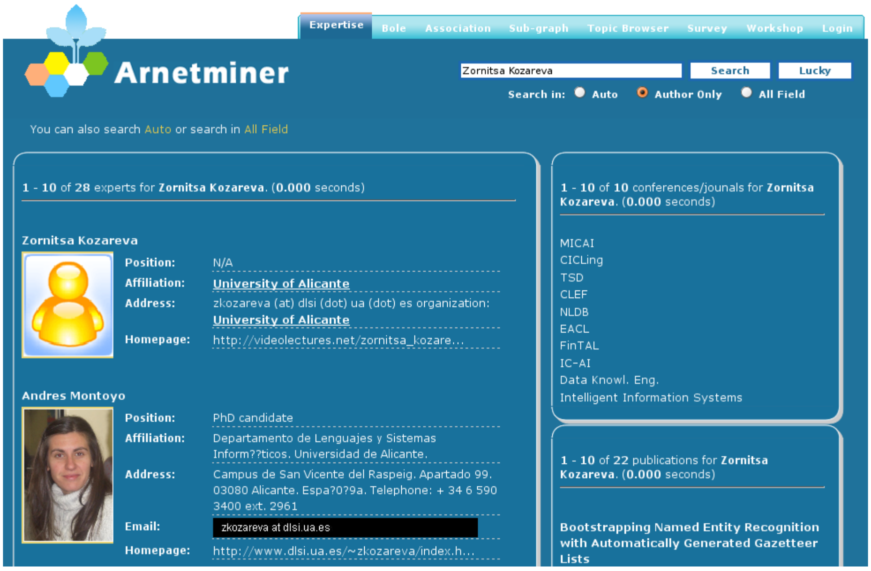
Dekang Lin (Edmonton, Alberta, Canada)
professor, university of alberta, computer science
See More Details

Dekang Lin (San Francisco Bay Area, California, United States)
professor, university of alberta, computer software, tsinghua university, Google, Staff Research Scientist
See More Details

Dekang Lin
professor, university of alberta, Department of Computing Science
See More Details

Dekang Lin
professor, university of alberta, Department of Computing Science
See More Details

Figure 1.5: Name disambiguation error in Spock



Zornitsa Kozareva | Search | Lucky

Search in: ○ Auto ● Author Only ○ All Field

You can also search Auto or search in All Field

1 - **10** of **28** experts for **Zornitsa Kozareva**. (**0.000** seconds)

Zornitsa Kozareva

Position: N/A
Affiliation: University of Alicante
Address: zkozareva (at) dlsi (dot) ua (dot) es organization: University of Alicante
Homepage: http://videolectures.net/zornitsa_kozare...

Andres Montoyo

Position: PhD candidate
Affiliation: Departamento de Lenguajes y Sistemas Inform??ticos. Universidad de Alicante.
Address: Campus de San Vicente del Raspeig. Apartado 99. 03080 Alicante. Espa?0?9a. Telephone: + 34 6 590 3400 ext. 2961
Email: zkozareva at dlsi.ua.es
Homepage: http://www.dlsi.ua.es/~zkozareva/index.h...

1 - **10** of **10** conferences/jounals for **Zornitsa Kozareva**. (**0.000** seconds)

MICAI
CICLing
TSD
CLEF
NLDB
EACL
FinTAL
IC-AI
Data Knowl. Eng.
Intelligent Information Systems

1 - **10** of **22** publications for **Zornitsa Kozareva**. (**0.000** seconds)

Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists

Figure 1.6: Name disambiguation error in ArnetMiner

Although ZoomInfo management does not publish details on the methods they use to solve ambiguity, we might suppose it is based on document clustering techniques. Whichever method is used, it is relatively easy to find errors in the documents grouping and information extraction results. In a test of the query by the name ”Felisa Verdejo” (fig. 1.4) three profiles that actually belong to the same person were returned. Each profile relate the person to one organisation (one is related to the UNED University, another to the Asia-Pacific Society for Computers in Education and, finally, a third one to the Association for Computational Linguistics). In this case, the search service was not able to link all this information to the same person.

Spock

In 2007, the start-up company Spock launched a new people search service. This service combines information from structured sources (including Wikipedia, IMDB, ESPN, LinkedIN, Hi5, Myspaces, Friendster, Facebook, Youtube, Flickr, etc.) with information extracted from general pages on the Web. As it happens in Zoominfo, it is not difficult to find examples of erroneous document groupings in Spock. In Figure 1.5 we can see information about professor Dekang Lin spread across four different profiles. One year after starting their people search service, Spock offered a 50,000$ prize to a team that could automatically solve the ambiguity of people names on a large testbed with the highest accuracy. The challenge was held from April to December 2007, and met over 1500 participants from around the world. A six-person team of researchers from Germany’s Bauhaus University Weimar were awarded the prize. Unfortunately, neither the evaluation methodology nor approach of the best team were made public.

ArnetMiner

In the domain of Computer Science researchers, ArnetMiner [TZZ+07] gathers different search functionalities: finding experts on a specific field, browsing the social network of authors, exploring events by topic, etc. In many of these applications, the problem of name ambiguity is present, and ArnetMiner provides built-in automatic methods to tackle it. Given the highly structured information available for this domain, ArnetMiner is able to use, for instance, the author’s publications metadata (e.g. title, conference, year, abstract, authors, references, etc.) to perform name disambiguation. For instance, as shown in Figure 1.6, we search for the name “Zortnisa Kozareva”. Although these results show an excellent performance in terms of information extraction (email, affiliation, address, position, etc), the two profiles are actually referring to the same person, and, furthermore, the second profile has been misnamed as “Andres Montoyo” instead of “Zortnisa Kozareva”.

1.3 Web People Search and Other Related NLP Tasks

In this thesis we define Web People Search (WePS) as the task of clustering a set of web pages, which are the result of a Web search for a person name, in as many groups as entities sharing that name.

This work is focused on the problem of name homonymy, meaning that the same name can refer to multiple people. A related but different problem is name synonymy resolution (or name variation), where different names refer to the same person [RK99, WRC97]. In its most traditional formulation, entity coreference tries to solve both problems at the same time, including the problem of linking noun phrases and pronouns, and considering different types of ambiguous entities (people names, but also locations and organisations).

Cross-document coreference

Name homonymy is not as frequent inside the same document as it is when considering a whole collection of texts. In this sense Cross-document Coreference Resolution (CDC) is a task strongly related with WePS. The objective of CDC is to reconstruct the coreference chain (i.e., a set of expressions referring to one individual) of an entity mentioned in a collection of documents. Unlike CDC, the WePS task focuses on a single person name and does not require building a complete coreference chain linking every mention of entities in the collection, but just grouping documents that mention the same person with that name. The CDC task varies different research works: in some cases it only deals with people names [BB98b, GA04], it includes in other cases other entity types [LMR05, PPK05]; and in other works it includes name variation [Blu05, BF08]. As we will show in Chapter 2, CDC evolution has set the ground for the methods and evaluation methodology that will be used in the WePS task.

Word Sense Disambiguation

WePS has close links with Word Sense Disambiguation (WSD). WSD is the task assigning a sense to a word in a given context [AE06]. In both cases, the addressed problem is the resolution of the ambiguity in a natural language expression. However, we should consider some differences:

- WSD can rely on dictionaries to define the number of possible senses in a word. In the case of name ambiguity, no such dictionary is available, even though, in plain theory there is an exact number of people sharing the same name.

- WSD typically focuses on the disambiguation of common words (nouns, verbs, adjectives) for which a relatively small number of senses exist compared to the hundreds or thousands of people that might share the same name. However, word senses in dictionaries often bear subtle differences which make them hard to distinguish in practice, while person name ambiguity can be considered as a homograph-level type of ambiguity.

- Boundaries between word senses in a dictionary are often subtle or even conflicting, making binary decisions is harder and sometimes even useless, depending on the application. In the case of person name disambiguation, distinctions can be either easier to establish (radically different people sharing the same name) or very difficult (for instance, when namesakes share the same occupation), but there is always an objective reality behind the different "senses" of a person name.

Word Sense Induction

An interesting variation of WSD is presented by the Word Sense Induction task (WSI) [PL02b], also known as Word Sense Discrimination. In this case, the goal consists of discovering the senses of a word in a given a set of contexts. This task is closer to WePS in that no dictionary is used to guide the disambiguation process and that semantic clustering techniques (see Chapter 2.4) typically have a predominant role both in WePS and WSI [PL02b, Nei02, Rap03]. Still, WSI maintains the other essential differences we have mentioned between word and name sense disambiguation.

Citation disambiguation

A particular instance of the WePS task takes place in online communities where each individual is explicitly linked to others. This is the case, for example, of authors mentioned in scholar citations [HZG05, TKL06, KNL^{+}09]. As we saw in the previous section, ArnetMiner provides a disambiguation solution for this domain restricted scenario. The case of author ambiguity in scholar citations presents a major problem for the integration and search of bibliographic sources. Citation Disambiguation (CD) tries to solve this issue, usually exploiting the meta-information available in most publications (co-authorship, author affiliation, email, etc).

With the advent of social networks in the WWW, name disambiguation has become major focus of research in those cases too. Malin [Mal05] studied methods for the disambiguation of entities in the Internet Movie Database. Bekkerman [BM05] approaches the problem of disambiguating web appearances of a group of people.

Expert finding

WePS can also be considered as an intermediate task in fields like Expert finding (EF), which consists of finding people that have a certain type of knowledge [Bal08]. This task has recently received increased attention, especially since the launch of an expert finding task as part of the enterprise track at TREC in 2005 [CdVS05]. Given a query (describing the area in which a particular expertise is being sought), participating systems have to return a ranked list of people names in response. In this case, WePS could serve as means of providing more accurate and complete information about people in a particular area of expertise.

Results re-ranking

Name disambiguation has been also formulated as the task of, given one relevant document about a person, finding other documents that talk about the same person [Guh04]. The input of this task is not person name, but a document mentioning a particular person and a ranking of search results. The goal here is to bring documents that are relevant for that particular user information need on top. A drawback of this interaction model is that users must browse search results in order to find at least one document that matches the person they are looking for. To our knowledge, this approach to the name disambiguation problem has not been pursued in other works.

1.4 Goals

The main goal of this thesis is to create the appropriate resources and to establish a well grounded methodology for the evaluation of name disambiguation systems. This goal in turn is divided in three main objectives:

1. To formalise the name disambiguation problem in web search results. The following goals are derived from this objective:

    - Review the treatment of the name disambiguation problem in related Natural Language Processing areas (Word Sense Disambiguation, Cross-document Coreference, Record Linkage, Expert Finding, Citation Disambiguation, etc).

- To motivate empirically the need for automatic methods in order to assist Web People Search.

2. To create an evaluation framework for Web People Search systems. This aim implies the following sub-goals:

    - To define a name disambiguation task that allows the empirical comparison of different approaches.
    - To create a testbed corpus, based on real web results for ambiguous people names. The testbed should provide a representative sample of the problem.
    - To adopt an evaluation methodology and quality measures for WePS systems.

3. To analyse the most prominent features of the name disambiguation problem as well as the most promising research directions in the development of WePS systems. Specifically, we focus on the document representation phase. To validate the impact of different document representations empirically, independently of the choice if clustering algorithm, term weighting criteria, etc. We aim to compare the performance of features requiring an in-depth linguistic processing (mainly, named entities) versus features like word n-grams and terms in the document.

1.5 How we Addressed the Web People Search Problem

This thesis has followed an iterative development process. It started with an initial planning in which preliminary studies were carried out and the name disambiguation task was formalised. The iterative process featured the implementation of two consecutive evaluation campaigns and, in between, the refinement of evaluation metrics of the task. Finally, empirical studies based on the resources and knowledge developed in the previous research steps (test collections, evaluation methodology, baseline approaches, systems results, etc.) were performed. From a chronological perspective, our work has comprised the following broad steps (Figure 1.7):

Task formalisation. In first place, we studied the previous work related with the name disambiguation problem. Based on this survey of the state of the art, we formalised the task of people search in the World Wide Web.

Preliminary studies. The study of previous work showed us the need for a more representative test collection for web name disambiguation. We developed a preliminary test collection based on these premises and then we evaluated baseline approximations on it [AGV05].

First evaluation campaign. Based on the methodology developed for the preliminary test collection, we extended the test collection with a larger set of names and web pages [AGS07]. A competitive evaluation campaign was organised, with this extended testbed as its central resource (WePS-1 [AGS07]). For this campaign, we used standard document clustering evaluation metrics, as well as the baseline approximations defined in the previous step.

Evaluation methodology refinement. This first evaluation campaign led to a more comprehensive study of clustering evaluation metrics and the proposal of a metric adapted to the task evaluation needs [AGAV08]. Both this testbed and the data obtained from the participant systems allowed us to study the role of quality metrics weighting when comparing systems in a clustering evaluation campaign like WePS-1 [AGA09a].

Second evaluation campaign. Based on the knowledge acquired during the first edition and the research on evaluation metrics, a second evaluation campaign was organised (WePS-2 [AGS09]). Furthermore, the related problem of personal information extraction from the Web was introduced as a pilot task [SA09, ASG08].

Empirical studies. With the wealth of approaches developed so far, we carried out a study to measure the impact of the different features commonly used to represent documents [AAG09]. The performance of manual query refinement strategies was empirically evaluated on different people search scenarios (using test collections and manual annotations generated in previous steps). We studied the conditions in which query refinement strategies are useful for this task [AGA09b].

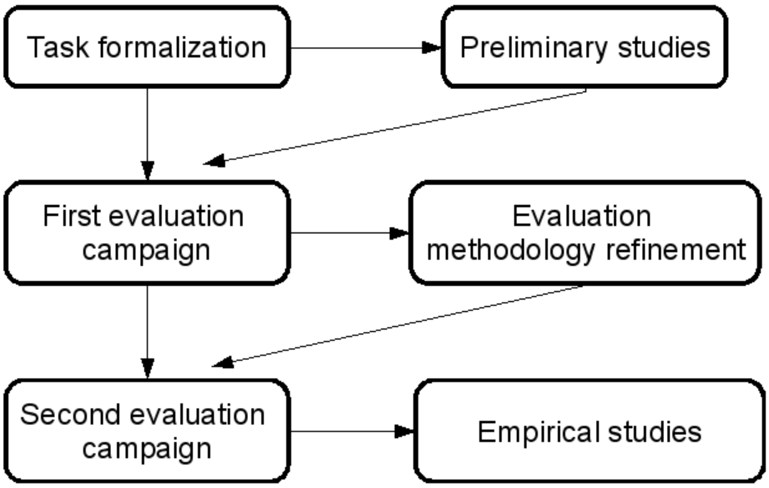

Task formalization
Preliminary studies
First evaluation campaign
Evaluation methodology refinement
Second evaluation campaign
Empirical studies

Figure 1.7: Thesis iterative process

1.6 Structure of the Thesis

This thesis is organised in three parts, namely: background, benchmarking and empirical studies. The first part introduces the problem of person name ambiguity and provides a survey of the work related to this question, mainly in the area of cross-document coreference. The second part explains the development of an evaluation framework for name disambiguation systems and its implementation in two competitive evaluation campaigns. Finally, the third part presents empirical studies on the impact of query refinement strategies and the impact of different feature representations. These main parts comprise the following individual chapters:

I: Background

- **Chapter 1** presents the motivation for the study of automatic approaches to the name disambiguation problem. We formalise this problem as the "Web People Search task". We review current commercial initiatives and present a brief survey of the related research areas.
- **Chapter 2** provides a survey of approaches to the problem prior to our work. This survey reviews the test collections, document representation approaches, similarity metrics, clustering methods and evaluation metrics employed in the literature.

II: Benchmarking

- **Chapter 3** presents the task definition, resources, participation, and comparative results for the Web People Search task, which was organised as part of the SemEval-2007 evaluation exercise. This task consists of, given the first 100 documents retrieved from a web search engine using an ambiguous person name as query, cluster them according to the actual entities that are mentioned in each document.
- In **Chapter 4** we define a few intuitive formal constraints which shed light on those aspects of the quality of a clustering that are captured by different metric families. These formal constraints are compared with other constraints proposed in the literature. Considering the characteristics of Web People Search, we also extend the analysis to the problem of overlapping clustering, where items can simultaneously belong to more than one cluster. BCubed metrics [BB98b] are chosen as the only ones that both satisfy all formal constraints and can be adapted to cover the overlapping clustering task.
- **Chapter 5** presents the *Unanimous Improvement Ratio* (UIR), a measure that allows to compare systems using two evaluation metrics without dependencies on relative metric weights. For clustering tasks, this kind of measure becomes necessary given the trade-off between precision and recall oriented metrics which usually depends on a clustering threshold parameter stated in the algorithm.
- **Chapter 6** describes the second WePS (Web People Search) Evaluation campaign. This chapter presents the definition, resources, methodology and evaluation metrics, participation and comparative results for the clustering task.

III: Empirical studies

- In **Chapter 7** we study whether it is reasonable to assume that pages about the desired person can be interactively filtered by the user by adding query terms. We justify the need for automatic methods that solve the person name ambiguity on web search results empirically.
- In **Chapter 8** we compare the coverage, reliability and independence of a number of features constitute potential information sources for this clustering task, paying special attention to the role of named entities in the texts to be clustered. Although named entities are used in most approaches, our results show that, regardless of the Machine Learning or Clustering algorithm used, named entity recognition and classification by themselves only make a small contribution to solve the problem.
- **Chapter 8** discusses our conclusions and contributions. We also present the future research lines of this work.

Chapter 2

State of the Art

In this chapter, we will review previous work related to the resolution and evaluation of person name ambiguity. We will present the main features of test collections that have been used in order to study the performance of different systems, as well as the evaluation metrics that have been used to measure it. We will also review the methods and document representations different researchers have chosen in previous works about the subject.

2.1 Test Collections

Test collections constitute an essential tool for comparing different approaches to the same Natural Language Processing (NLP) task. Typically, an NLP test collection comprises a textual corpus and data that is used as ground truth to be compared to the systems output.

Systems for person name disambiguation have been tested initially on cross-document coreference (see Section 1.3) test collections. In these collections, each mention of an ambiguous name is disambiguated. Note that, strictly speaking, for the cross-document coreference task, each mention of an ambiguous name has to be annotated, while in the name disambiguation task, as we have defined it (see Introduction), it suffices to group the documents containing at least one mention referring to the same person with the ambiguous name. Many cross-document coreference test collections have been built upon newswire corpora [BB98b, WL02, FH04, GA04, PPK05, Ped06]. It has not been until recently that web collections have become predominant [MY03, AKE04, WGLD05, Man06, CM07a]. These collections are usually obtained by querying a search engine for an ambiguous name and retrieving a certain number of documents from the top results. Web collections are characterised by noisy contexts, in which well-formed sentences are not as abundant as they are in news articles and little or no information is provided to disambiguate a name.

Only few of these collections have been reused by different researchers. Usually, they are created ad-hoc for each particular research work. In addition to that, the evaluation methodology, quality measures and task definitions vary among research works. These conditions have prevented a consistent comparison of different approaches.

We will classify test collections in two types, according to the method employed for its creation: (i) manually annotated testbeds and (ii) testbeds created using automatic methods (pseudo-ambiguity).

2.1.1 Manually Annotated Test Collections

Manually annotated collections represent the most straightforward method of creating a testbed for a task. In manually annotated collections the first step is to select a text corpora together with one or more ambiguous names in it. Then, each mention of the ambiguous names is manually tagged according to the individual it refers to. This method of creating test collections requires a substantial amount of time as well as human resources to annotate even a relatively small amount of text. For instance, Gideon Mann reports an average 3-4 hours of work to group a set of 100 documents mentioning only one ambiguous name [Man06].

John Smith corpus

In 1998, Bagga and Balwin [BB98b] created the first name disambiguation testbed with a single ambiguous name. It gathered 197 news articles with the name "John Smith" from the 1996 and 1997 editions of the New York Times. Articles which either contained the name "John Smith" or some variation with a middle/initial name were selected. The answer keys consisted of manually created cross-document coreference chains[1]. 35 different "John Smiths" were found in the collection. 24 out of these had only one article which mentioned them. The other 173 articles referred to the 11 remaining "John Smiths". As stated by the authors, there is a great variability both on the background of these people and on the number of articles that mention each individual.

Mann 2003

Since Baggas work, many researchers decided to use automatic methods based on automatic pseudo-ambiguity to create larger test collections. We will describe this type of collections in the next section. Nevertheless, in some cases, manual and pseudo-ambiguous test collections were used to complement each other. Mann [MY03] made a small manual collection of only 4 naturally ambiguous names (as compared to 28 pseudo-names in his main test collection). For each name, 100 web search results were downloaded and hand labelled, obtaining an average of 60 different people for each name. Also, Bollegala [BMI06] evaluated both manual and automatically annotated collections. The manual dataset was composed of over 1000 web pages retrieved from Google, using three people names as queries. A significantly lower number of individuals were identified in this case (8, 3 and 10 individuals, respectively for the three names).

Fleischman 2004

Fleischman [FH04] acknowledged the limitations of automatic annotation methods and used a manually annotated collection exclusively. In this case, the annotation effort was reduced by the fact that a very specific and short context had to be disambiguated. His work focused on the task of solving name ambiguity when populating an ontology with concept/instance (noun phrase/person name) pairs

[1] In this case, one coreference chain is a set of expressions in the text collection referring to a particular individual with the name "John Smith".

automatically extracted from text (Table 2.1). Pairs sharing the exact same instance (person name) were considered ambiguous. Rather than from full text documents, these pairs are extracted from a large corpus of newspaper articles (described in [FH04]). A set of 31 names and their corresponding pairs were extracted from the collection. 11 of these names turned out to refer to more than one person whereas the remaining 20 only referred to a single person. This testbed assumes that all names are previously linked to their corresponding noun-phrases. This in itself is a difficult coreference problem, and such informative phrases might not be available in all documents. Also, they might not be the only relevant information in documents for the disambiguation process.

instance	concept	referent
Paul Simon	pop star	1
Paul Simon	singer	1
Paul Simon	politician	2

Table 2.1: Example of concept-instance pairs

Al-Kamha 2004

Al-Kamha [AKE04] created one of the first name disambiguation collections based on web documents. In his collection 19 ambiguous people names were selected, then each name was used as a query for Google and the top 50 web results were collected and annotated. Unfortunately, no further information is provided about the number of individuals found for each name or the annotation process.

Wan 2005

A larger web test collection was produced a year later by Wan [WGLD05], who selected the 200 most-frequent person queries from the search log of Microsoft's portal (MSN). The top 100 search results were collected for each name. This method of selecting the people names might be responsible for the introduction of a bias for famous people in the collection, since popular person queries will most likely be about celebrities, and the ranking function of a search engine will tend to show the results for these celebrities on top. Before the annotation, 2% of the pages were filtered out: (i) pages in which there was no occurrence of the person name is used to mention a specific person and, (ii) pages that mention to two or more referents with the same ambiguous name[2]. All remaining person pages were grouped into different clusters, obtaining an average of 6.88 referents per ambiguous name. This is significantly lower than the 60 average individuals obtained in Mann's 2003 web testbed [MY03], and seems to confirm the predominance of popular people in Wan's collection.

Boulder Name corpus

Multilingual corpora has been used in name ambiguity also [CM07a, Ped06]. For instance, Ying Chen [CM07a] created a small testbed (Boulder Name corpus) for both English and Chinese web news documents[3]. Four data sets were created for English (James Jones, John Smith, Michael Johnson, Robert Smith) and four data sets for Chinese (Li Gang, Li Hai, Liu Bo, Zhang Yong). For each person name the first non-duplicated 100 search results were retrieved from Google (Chinese) or

[2] This means one individual per ambiguous name/document was assumed.

[3] Google News (http://news.google.com) was used to retrieve the results for each ambiguous name.

Google news (English). The ambiguity found in this collection is characterised by a few popular individuals and a long tail of “singletons” (people mentioned in only one document). This effort for multilingual collections was preceded by Pedersen [Ped06], who used pseudoambiguity in the annotation process (see next section).

Web03 corpus

The Web03 corpus [Man06] [4] is the test collection which best fits our definition of the Web People Search task. It comprises web documents retrieved from a search engine. The selected ambiguous names are common, yet not necessarily associated to a celebrity, broadening the types of ambiguity scenarios to be found in the collection. For the name selection first and last names were sampled independently from the U.S. Census distribution. Each name was searched in Google, and then the top 100 search results (at maximum) were downloaded. The resulting collection is composed of 882 web pages, 32 names and 212 people sharing those names. A conservative approach was used to group the pages: two pages were considered to refer to the same person only when certain information appearing in both pages could be used to relate them (e.g. same affiliation, city of residence, etc.). If no matching information was found between the two pages, they remained separate. As in the Boulder Name Corpus, a long tail of infrequent individuals is found in this collection: 155 (73%) out of all 212 referents were mentioned in just one page. The large number of people that only appears in one document is due to the usage of very common names from the U.S. Census. The average number of people sharing a name is 6.6. This might seem a low degree of ambiguity for such common names, but it must be noted that in average, the actual number of documents annotated for each name is also very low (27.5).

Table 2.2 summarises some of the characteristics of the manual collections we have reviewed [5]. Most of the collections reviewed are not appropriate for the WePS task. Newswire datasets like the “John Smith” corpus and Fleischman’s collection [BB98b, FH04] are discarded, because we are focused on web search results. The criteria to sample people names also proves to be conflictive. Using only popular names as in Wan’s collection [WGLD05] does not relate to search scenarios such as the one described in the introduction to this thesis. In cases like Mann 2003, Bollegala 2006 and the Boulder corpus [MY03, BMI06, CM07a], the number of ambiguous names sampled is too small.

All in all, Mann’s “Web03” collection [Man06] provides the best testbed for the evaluation of the name disambiguation task[6]. And yet, there are aspects of the testbed design that could be improved. For instance: (i) providing a wider variety of name sources (both common names and names of celebrities), (ii) increasing the number of samples of ambiguous names in the collection and (iii) increasing the number of documents for each name.

[4] http://www.cs.jhu.edu/ gsm/publications/Web03.tar.gz

[5] Note that not all the information is available for each collection.

[6] [AKE04] is a similar collection but has a smaller number of ambiguous names.

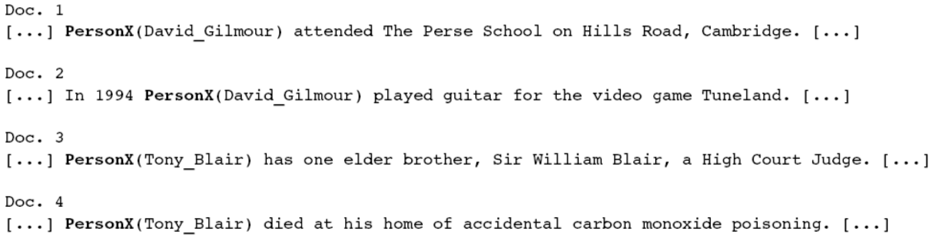
year	name	# docs	# names	name type	# people	source	languages
1998	John Smith corpus	173	1	one very common name	11	newswire	English
2003	Fleishman 2004	2675 (concept - instance pairs)	31	names mentioned in news	11 names referred to multiple people, 20 names had only a single referent	newswire	English
2003	Mann 2003	up to 100 per name	4	-	60 in average	Web	English
2004	Al-Kamha 2004	50 per name	19	random names from a phone book	-	Web	English
2005	Wan 2005	up to 100 per name	200	most popular names in a query log	6.8 in average	Web	English
2006	Bollegala 2006	1000	3	-	8, 3 and 10	Web	English
2006	Mann’s “Web03”	882	32	common names from US Census	6.6 in average	Web	English
2007	Boulder corpus	100 per name	8	-	-	Web	English, Chinese

Table 2.2: Summary of manually annotated test collections

2.1.2 Pseudo-ambiguity Test Collections

Name conflation

Pseudo-ambiguity has been used as an inexpensive way of creating Word Sense Disambiguation (WSD) testbeds[7]. In the case of people names ambiguity, pseudo-ambiguity consists of generating one artificial ambiguous name by replacing two or more names with a common string. This process is usually referred as *name conflation*. For instance, in Figure 2.1, two names (David Gilmour and Tony Blair) have been conflated in one pseudo-name (PersonX). This process is repeated for every occurrence of these two names in the collection of documents, and the information about the original names is saved as ground truth to evaluate disambiguation systems.



```
Doc. 1
[...] PersonX(David_Gilmour) attended The Perse School on Hills Road, Cambridge. [...]

Doc. 2
[...] In 1994 PersonX(David_Gilmour) played guitar for the video game Tuneland. [...]

Doc. 3
[...] PersonX(Tony_Blair) has one elder brother, Sir William Blair, a High Court Judge. [...]

Doc. 4
[...] PersonX(Tony_Blair) died at his home of accidental carbon monoxide poisoning. [...]
```

Figure 2.1: Example of name conflation for generating a pseudo-ambiguous name

Spurious errors

Pseudo-ambiguity provides a fast method for generating large ambiguity testbeds, but it can also present serious drawbacks. In the case of name ambiguity, this method assumes that names chosen to be conflated under a pseudo-name are not ambiguous themselves. In Figure 2.1, the name “Tony Blair” is used to create a pseudo-name, assuming that it will most likely refer to the British prime minister in all cases. But, in fact, the mentions in documents 3 and 4 actually refer to different people (the former British prime minister and a Governor of Missouri, respectively). A system might detect ambiguity has not been recognised previously in the gold standard, thus the evaluation can penalise spurious errors. A common strategy to avoid this problem is to conflate only names of famous people, so that all mentions are most likely to refer to the same celebrity [MY03, PPK05, Ped06, BMI06]. This is usually true for small news collections and the top search results from a web search engine. However, errors can still occur, and the bigger the collection the more likely errors might appear (e.g. the top results for the query “Michael Jackson” will probably be monopolised by the singer, but as we go further down in the ranking, less popular people will start to appear).

“Senses” frequency distribution

Furthermore, pseudo-names do not necessarily recreate the “senses” frequency distribution, nor the amount of ambiguity to be found in naturally ambiguous names. Actually, one of the difficulties of using pseudoambiguity is that the number of individuals per name has to be fixed manually. This is specially difficult for people names since there is a wide range of ambiguity that depends on many factors (number of documents, presence of a celebrity, frequency of the name in the

[7] The use of pseudo-words for the creation of WSD test collections was introduced in [GCY92] and [Sch92].

population, etc).

“person-x” corpus

The John Smith corpus created by Bagga [BB98b] (see previous section) was used six years later by Gooi and Allan [GA04] to compare their system with Bagga’s original approach. The small size of the collection and the fact that it only represented the ambiguity of one name motivated the creation of a larger testbed using automatic annotation. A single pseudo-ambiguous name was generated from the conflation of a large list of people names. First, a named entity recognition tool was used to detect people names in the Text Retrieval Conference (TREC) newswire collections. Then, all occurrences of a random set of people names was replaced by the string “person-x”, keeping the information about the original names. A total of 14,767 different people names mentioned 34,404 times were replaced with the pseudo-name “person-x”. Although the annotation process is automatic, thorough manual work had to be carried out in order to avoid conflating ambiguous names and to include variations of the chosen names in each document. The result was a single, but enormously ambiguous, pseudo-name. The primary objective of this work was to compare Bagga’s approach with others, and thus the corpus design was intended to be similar in that it contained only one ambiguous name. Leaving aside the downsides related to pseudo-ambiguous names, it seems unrealistic to evaluate only one ambiguous name at the time. Names can provide radically different ambiguity scenarios and, as we will see in Chapter 3, this circumstance has consequences in the performance of name disambiguation systems.

Mann 2003

In Mann [MY03], 28 pseudo-names were made by conflating two presumably unambiguous names. These pseudo-names were created combining 8 real people names of celebrities, historical figures, etc. In the same work, Mann used a smaller manually annotated corpus (see previous section). It is noteworthy that there is quite a wide gap between the ambiguity assigned to pseudo-names (2 individuals per name) and the one found in naturally ambiguous names (an average of 60 different people reported for 4 different names). Actually, Mann does not evaluate all clusters in the hand labelled collection, but rather a three way partition of the clustering (the two biggest clusters, plus one containing everything else in the collection). The evaluation is only performed over the two biggest clusters. This choice removes the difficulty of clustering infrequent people, for which little information is available, and makes results found in the two collections more comparable. On the other hand, the reality of the hand labelled data is largely ignored, which, in a way, seems to spoil the interest of evaluating on naturally occurring ambiguity.

Pedersen 2005

Another example of fixed ambiguity is Pedersen’s 2005 testbed [PPK05], where pseudo-names are limited to only two “senses”. All the contexts associated with each pair were extracted from a large corpus of newswire text. Each context consisted of approximately 25 words to the left and right of the ambiguous name. The corpus employed in these experiments was the Agence France Press English Service (AFE) portion of the GigaWord English Corpus, as distributed by the Linguistic Data Consortium. The AFE corpus consists of 170,969,000 words of English text which appeared in the AFE newswire from May 1994 to May 1997, and from December 2001 to June 2002. Overall this represents approximately 1.2 GB of text (uncompressed).

Pedersen 2006

A multilingual test collection was created by Pedersen [Ped06] from a large newswire corpora in four languages, Bulgarian, English, Romanian and Spanish. Evaluation contexts were created by conflating together pairs of popular names of people and places (likely to be unambiguous). The following pairs of names were conflated in all four languages: George Bush-Tony Blair, Mexico-India, USA-Paris, Ronaldo-David Beckham, Diego Maradona-Roberto Baggio, and NATO-USA. As in its previous work, these pairs were conflated creating pseudo-names with two "senses".

Bollegala 2006

Bollegala [BMI06] carried out an evaluation using fixed ambiguity pseudo-names as well as naturally ambiguous names (see previous section). 50 documents were obtained from a web search engine for three different people names, and then merged on a single pseudo-name. The names correspond to Maria Sharapova, Bill Gates and Bill Clinton.

Spock Challenge corpus

One of the largest pseudo-ambiguity test collections for name disambiguation was released between April - December 2007 by the startup company Spock. This collection was part of a competition for automatic person name disambiguation systems[8]. The wining team was awarded a monetary price. Unfortunately, details about the test collection, evaluation methodology and winning strategy were not published. The test collection was available only during the development period of the contest.

In Table 2.3, we have summed up the main features of automatically generated test collections. Surprisingly, these collections have been generated for a small number of names (with the exception of [Man06]). Furthermore, the amount of "senses" assigned to pseudo-names has been either extremely low or over the top (14767 people for one name in the "person-x" corpus), but nobody has tried to emulate the distribution of people names ambiguity that can be found on naturally ambiguous names. In this thesis we will show that estimating the number of people mentioned with the ambiguous name is one of the main challenges for system designers. For this reason, the generation of test collections through pseudo-names does not seem an optimal choice.

[8] http://challenge.spock.com/

year	name	# docs	# names	name type	# people	source	language/s
2003	Mann2003	up to 100 per name	28	famous people with similar backgrounds	2 individuals for each name	Web	English
2004	“person-x” corpus	34404 mentions	1	names in news articles	14767	news articles	English
2005	Pedersen2005	-	6	famous people	2 individuals for each name	news articles	English
2006	Pedersen2006	-	3	famous people	2 individuals for each name	news articles	Bulgarian, English, Romanian, and Spanish
2006	Bollegala 2006	150	1	famous people	3	Web	English
2007	Spock challenge corpus	-	-	-	-	Web	English

Table 2.3: Summary of pseudo-ambiguous test collections

2.2 Document Representation

Bag of Words

Many different features have been used to represent documents in which an ambiguous name is mentioned. The most basic is the Bag of Words (BoW) representation, where the document text is processed as an unordered collection of words. In most systems, words from the full document are used either as the only feature [BB98b, GA04] or in combination with others [KCN$^+$07]. Other works have used smaller portions of the document to produce BoW representations. For instance, Bagga [BB98b] produced within-document coreference chains in order to extract all sentences that refer to the same entity in the document (a similar approach is followed by Gooi and Bollegala [GA04, BMI06]). A simpler approach consists of selecting the text on a window of words around each occurrence of the ambiguous name. In Mann's work [Man06], context windows of 50 or 100 words obtained significantly better results than the use of the entire document in a pseudo-ambiguity test collection, but no significant difference in the naturally ambiguous names was ascertained. Finally, word n-grams have been also employed. Document representation consisting of statistically significant bigrams that occur in the same context as the ambiguous name have also been used [PPK05].

Wikipedia information

Researchers like Cucerzan and Nguyen [Cuc07, NC08] have explored the use of Wikipedia information to improve the disambiguation process. Wikipedia provides candidate entities that are linked to specific mentions in a text. The obvious limitation of this approach lies in the fact that only celebrities and historical figures can be identified in this way. These approaches are yet to be applied to the specific task of grouping search results.

Biographical features

Biographical features are strongly related to NEs and have been also proposed for this task due to its high level of precision. Mann [MY03] extracted these features using lexical patterns to group pages about the same person. Al-Kamha [AKE04] used a simpler approach, based on hand coded features (e.g. email, zip codes, addresses, etc). In Wan's work [WGLD05], biographical information (person name, title, organisation, email address and phone number) is shown to improve the clustering results when combined with lexical features (words from the document) and NE (person, location, organisation).

Named entities

NEs are a frequently used feature for name disambiguation. Ravin [RK99] introduced a rule-based approach that tackles both variation and ambiguity while analysing the structure of names. In most recent research, NEs (person, location and organisations) are extracted from the text and used as a source of evidence to calculate the similarity between documents (see for instance [Blu05, KCN$^+$07]). However, the advantages of using NE have not yet been clarified. For instance, Blume [Blu05] uses NEs coocurring with the ambiguous mentions of a name as a key feature for the disambiguation process. Saggion [Sag08] compared the performace of NEs versus BoW features. In his experiments only one of several representations based on organisation NEs outperformed the word based approach. Furthermore, this result is highly dependent on the choice of metric weighting (NEs achieve high precision at the cost of a low recall and viceversa for BoW).

In summary, with the exception of representations that use the link structure

[BM05, Mal05] or graph representations [KCN$^+$07], the most common document representations for the problem include BoW and NEs, and in some cases biographical features retrieved from the text.

2.3 Similarity Metrics

The next step after choosing the set of features that represent documents mentioning an ambiguous name is to select a method of measuring the semantic similarity between those mentions. It is commonly assumed that similar contexts tend to refer to the same people. Hence, the similarity between documents is measured in the task of grouping documents about the same person.

Cosine

Cosine [MS99] is the most frequently used similarity metric in previous work [BB98b, WL02, MY03, AKE04, NLS04, WGLD05, BM05, Mal05, BMI06]. The distance between two documents is computed as the cosine of the angle between their corresponding vectors. Each component of a document vector represents a certain feature. Only the presence/absence of the feature might be indicated (for binary vectors), or a certain weight might be assigned in order to account for its importance in that specific document (real-valued vectors). For the general case of two n-dimensional vectors $\vec{x}$ and $\vec{y}$ in a real-valued space, the cosine measure can be calculated as follows:

$$cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Kullback-Leibler Divergence

Kullback-Leibler Divergence [MS99] has been also used in name disambiguation [GA04]. The KL divergence measures how different two probability distributions are. The more dissimilar the distributions are, the higher the KL divergence. If the distance is 0, then both distributions are identical.

Classification for metrics combination

Finally, classification has been used in the name disambiguation problem to obtain a value represeting the confidence that a pair of documents are coreferent [FH04, AKE04, Tiw05]. A document-to-document matrix is generated with these confidence values and then fed to a standard clustering algorithm. This approach has the advantage of optimising the combination of many similarity metrics according to a training collection. For instance, Fleischman [FH04] trained a Maximum Entropy model to give the probability that two names referred to the same individual.

2.4 Clustering Methods

Once the similarity between documents has been calculated, the next step consists of grouping documents mentioning the ambiguous name according to the actual individual they refer to. Name disambiguation systems use clustering methods to perform this step. Clustering algorithms group a set of elements (e.g. documents) into subsets or clusters. Their goal is to create internally coherent clusters that are clearly different from each other. In other words, elements within a cluster should be as similar as possible; and elements in different clusters should be as dissimilar as possible.

Unsupervised learning

Clustering is the most common form of unsupervised learning. It differs from supervised learning in that the learner is given only unlabelled examples. In clustering, it is both the distribution and makeup of the data which determine cluster membership. In the case of the name disambiguation problem, it is not feasible to learn how to classify documents for every individual. Thus, the unsupervised quality of clustering methods provides an appropriate approach to the problem.

Supervised learning

As we saw in the previous section, there are systems that, conversely, train a boolean classifier to decide whether a pair of documents is coreferent in order to obtain a reliable similarity criterion [FH04, NLS04, AKE04, Tiw05]. It is also common to train certain variables like the weight assigned to different features or the stopping criteria. So, it is not uncommon for systems in this task to use a development/training set to choose a value for those variables.

Overlapping clustering

In general, the assignment models employed in the previous work are disjunctive (*hard clustering*), meaning that one object can belong to one and only one cluster. This is related to the assumption that the occurrences of a person name in the same document always refer to the same entity (Yarowsky's "one sense per discourse"). In reality, it is not infrequent to find an ambiguous name referring to multiple people in the same document (*overlapping clustering*). As we will see in Chapters 3 and 6, our task definition does allow the same document to be assigned to multiple clusters to account for this fact in the data.

2.4.1 Clustering Process

Incremental Vector Space

In its foundational work of 1998 Bagga [BB98b] proposed a cross-document coreference system that uses the Incremental Vector Space clustering algorithm. The system works as follows: First, a cluster is created with one document. Then, the next document is compared against the first cluster. If the similarity computed is above a predefined threshold, then both documents are considered to be coreferent and clustered together. Otherwise, a new cluster is created for the document. At each step, a new document is compared to all existing clusters and then is subsequently merged into the cluster with the highest similarity if it is above the predefined threshold. The same process continues until all documents have been clustered.

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC), also called *bottom-up hierarchical clustering*, is frequently employed in name disambiguation literature [MY03, GA04, WGLD05, BMI06]. In this algorithm the first step is to create one cluster for each document in the collection. Then, for each cluster the similarity to all other clusters is calculated. If the highest similarity computed is above a predefined threshold, the two clusters are merged together (agglomerated). If any merging was performed during the last iteration, the process keeps looking for the most similar pairs and merging. The algorithm stops when no more merging is done (no pair of clusters has a similarity above the predefined threshold).

A HAC clustering can be visualised as a dendrogram. Each merge is represented by a horizontal line. The y-coordinate of the horizontal axis is the similarity of the two clusters that were merged, where documents are viewed as singleton clusters. For instance, Figure 2.2 shows a dendrogram of a clustering of 10 web search results for the query "John Smith". The name disambiguation task requires to output a flat partition of the documents. Consequently a number of criteria is available to select the most appropriate cutting point (see Section 2.4.2).

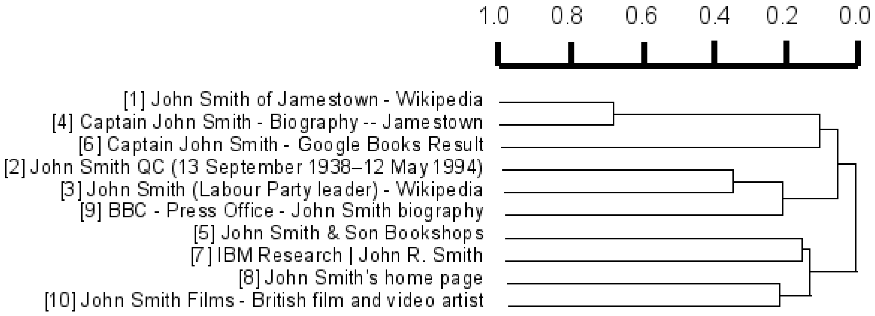

1.0
0.8
0.6
0.4
0.2
0.0
[1] John Smith of Jamestown - Wikipedia
[4] Captain John Smith - Biography -- Jamestown
[6] Captain John Smith - Google Books Result
[2] John Smith QC (13 September 1938–12 May 1994)
[3] John Smith (Labour Party leader) - Wikipedia
[9] BBC - Press Office - John Smith biography
[5] John Smith & Son Bookshops
[7] IBM Research | John R. Smith
[8] John Smith's home page
[10] John Smith Films - British film and video artist

Figure 2.2: A dendrogram of a clustering of 10 web search results for the query "John Smith"

IVS vs HAC

Bagga's IVS clustering approach and the HAC algorithm were empirically compared by Gooi in [GA04]. In his experiments Gooi noticed that early clusters formed by the IVS algorithm contained misplaced documents that, in turn, attracted yet other unwanted documents. HAC requires more comparisons but, it is order independent and, provided the appropriate choice of linkage[9], it can minimise problems caused by single noisy elements placed close to each other.

HAC with tree refactoring

A modified version of the HAC algorithm was presented by Mann [MY03]. Mann stops the clustering process before it finishes, relying on the percentage of documents clustered and the relative size of the clusters achieved. In HAC, the most similar documents are clustered first, therefore, at this intermediate stage, high-precision clusters can be obtained. These clusters are then used as seeds in a second stage, in which the unclustered documents are assigned to the seed with the closest distance measure. In these experiments, the number of individuals (only three for each pseudo-name) was known *a priori* and the information was used in order to determine the number of clusters within the output of the system.

Other approaches

Although HAC is the predominant clustering approach employed in the previous work on name disambiguation, other clustering methods have been applied too. An early work by Ravin [RK99] proposed a rule based system that takes into account variations of the name and the similarity between contexts in which the ambiguous name is mentioned. Although experimentation is performed on two news collections (NYT and WSG), it does not provide a systematic empirical evaluation based on a ground truth of the data. In [PPK05], the different individuals referred to by a person name are discriminated by clustering the context vectors of each name occurrence with the method of Repeated Bisections [ZK02]. The cluster is bisected

[9] See Section 2.4.3 for an explanation of the "chaining effect" in single-linkage clustering and its alternatives.

using standard K-means method with $K = 2$, while the criteria function maximises the similarity between each instance and the centroid of the cluster to which it is assigned.

2.4.2 Stopping Criteria

Similarity threshold

In a realistic name disambiguation scenario, the number of individuals sharing a name is not known in advance. In these cases, the most common approach consists of setting a similarity threshold on a hierarchical clustering algorithm. A similarity threshold is a stop criterion that tells the clustering algorithm which flat clustering solution must output. For instance, HAC can be used to produce a dendrogram that is then cut at a pre-specified level of similarity [GA04, WGLD05]. Gooi [GA04] observes that even though there are similarity thresholds that provide good results, clustering methods can be very sensitive to changes in this variable and therefore rapidly deteriorate. This threshold can be obtained by averaging the optimal thresholds on a training collection and applying this value to the test data [WGLD05]. An optimal threshold can be defined as the similarity merging threshold that obtains the best score according to an extrinsic clustering quality metric (i.e. a metric that compares a clustering solution with a gold standard). For this reason, a training corpus is necessary if we want to apply this method.

Internal clustering quality

Another approach consists of using internal clustering quality metrics that evaluate a clustering solution without comparing it to a gold standard [PK06, BMI06]. In [PK06] Pedersen presents a clustering system that supports four cluster stopping measures. The first three measures (PK1, PK2, PK3) look at the successive values of a criterion functions as K (the number of clusters) increases, thus trying to identify the point at which the criterion function stops improving significantly. There is also an adaptation of the Gap Statistic [TWH01], which compares the criterion function from the clustering of the observed data with the clustering of a null reference distribution in order to select the value of K for which the difference between them is greatest. Bollegala [BMI06] defines an internal cluster quality measure based on the internal and external correlation of the clusters. Internal correlation is a measure of how much the similarity of items within a cluster is maximised (i.e. the degree of similarity of documents within clusters) whereas external correlation constitutes a measure of how much the similarity of items between clusters is minimised.

2.4.3 Linkage

The linkage method [MS99] defines the way in which clusters are compared during the clustering process. There are three main linkage options.

Single linkage

In single-link clustering, the similarity of two clusters equals the similarity of their most similar members (see Figure 2.3). This type of linkage focuses on the area where both clusters come closest to each other. A drawback of this method (known as the "chaining effect") arises as clusters may be merged due to single noisy elements being close to each other, even though many of the elements in each cluster may be very distant to each other. Single linkage has been used in name disambiguation (see [BB98b] and [Man06]).

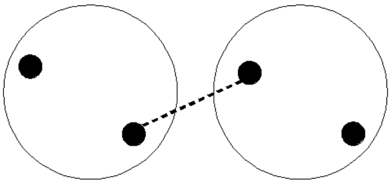


Figure 2.3: Single-link: maximum similarity

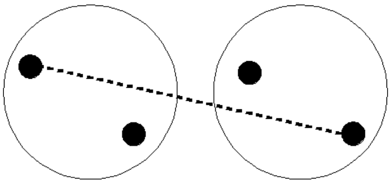


Figure 2.4: Complete-link: minimum similarity

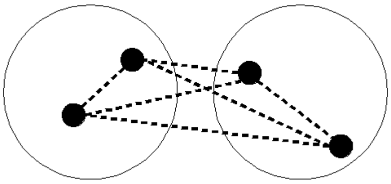


Figure 2.5: Group-average: average of all similarities

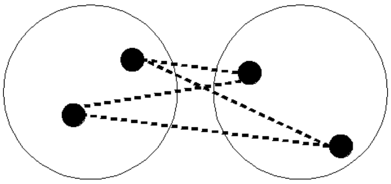


Figure 2.6: Centroid: average of inter-similarity

Complete linkage

Another type is complete linkage. In complete-link clustering (used in [Man06]), the similarity of two clusters equals the similarity of their most dissimilar members (see Figure 2.4). In this type of linkage, a single document far from the center of a cluster can prevent the merge with other clusters, having a decisive effect in the final result of the clustering.

Group average

In group average clustering, the criterion for merges is the average similarity of the cluster members (see Figure 2.5). Group average linkage has been used in name disambiguation systems like [Man06, GA04, WGLD05, BMI06].

Centroid

In centroid clustering, the similarity of two clusters is defined as the similarity of their centroids (see Figure 2.6). It has been used in [MY03, Man06].

Mann [Man06] compared the performance of single link, complete link and group-average clustering for name disambiguation using HAC. Overall group-average and centroid clustering methods yielded a better performance in experiments with pseudo-ambiguous and naturally ambiguous names.

2.5 Evaluation Metrics

Measuring the performance of a system is an essential aspect of Natural Language Processing research. The score for a particular system is the single measure of how well a system is performing and it can strongly determine directions for further improvements. In this section, we provide an overview of the evaluation metrics used in the coreference task that are related with the person name disambiguation problem. Even though name disambiguation systems have mostly used clustering techniques, coreference metrics have been applied predominantly[10].

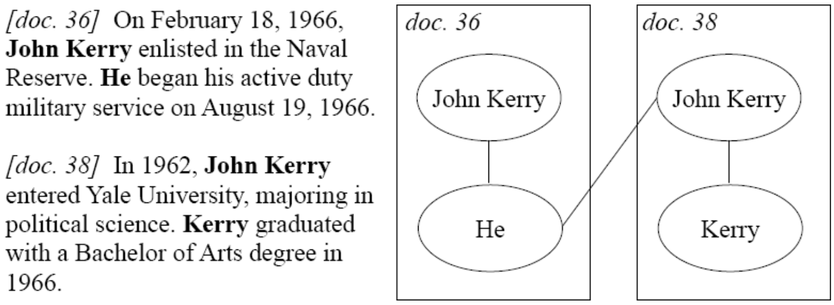
Coreference chains

In the field of cross-document coreference, the output expected from a system is a “coreference chain” for each entity. A cross-document coreference chain is the list of all expressions referring to the same individual (Figure 2.7) in the scope of a document collection. It can contain both occurrences of an ambiguous name as well as variations of that name and other expressions used to mention an individual. Note that these metrics can be applied to the name disambiguation clustering task we have defined. In our case. we only require systems to group documents containing at least one occurrence of an ambiguous name without taking into consideration the actual occurrences of that name (or variations) inside each document. Systems we have reviewed in this chapter focus on the name ambiguity, in some cases considering each occurrence of the name in the text (e.g. [BB98b, GA04]), and in others considering only the documents grouping (e.g. [Man06, BMI06]), as we have proposed.

MUC-6 coreference metric

According to Bagga [BB98b], cross-document coreference was identified as one of the potential tasks for the Sixth Message Understanding Conference (MUC-6) but was not included as a formal task because it was considered too ambitious [Gri94]. Instead, the simpler intra-document coreference task was included. The MUC-6 algorithm [VBA$^+$95] computes precision and recall by looking at the number of

[10] In Chapter 4 we give an survey of the families of metrics used to evaluate clustering systems and study their properties.



[doc. 36] On February 18, 1966, **John Kerry** enlisted in the Naval Reserve. **He** began his active duty military service on August 19, 1966.

[doc. 38] In 1962, **John Kerry** entered Yale University, majoring in political science. **Kerry** graduated with a Bachelor of Arts degree in 1966.

Figure 2.7: Example of a cross-document coreference chain

links identified by a system compared to the links in an answer key.

In a study of evaluation metrics for the coreference task, Bagga [BB98a] showed two shortcomings of the MUC-6 metric, namely: (i) it does not reward the ability of identifying coreference chains made of only one element and (ii) precision errors that intuitively are different in their importance receive the same penalty. Conversely he proposed a new scoring metric (B-Cubed) designed to overcome these problems.

B-Cubed metric

The B-Cubed scoring algorithm [BB98a, BB98b] focuses on the presence or absence of elements relative to each other in the coreference chains. This metric has been used in many cross-document coreference works, particularly in works focused on person name ambiguity [BB98b, WL02, NLS04, GA04, PNH06, Man06]. In Chapter 4, we give a detailed explanation of this metric and show that it meets several interesting properties.

Clustering metrics

Clustering metrics provide a straightforward method for evaluating name disambiguation systems. Given a similarity metric between objects, clustering evaluation metrics can be intrinsic, i.e., based on how close elements from one cluster are to each other, and how distant from elements in other clusters. Extrinsic metrics, on the other hand, are based on comparisons between the output of the clustering system and a *gold standard* usually built using human assessors. In this work we will focus on extrinsic measures, which are the most commonly used in text clustering problems.

When doing extrinsic evaluation, determining the distance between both clustering solutions (the system output and the *gold standard*) is non-trivial and still subject to discussion. Many different evaluation metrics have been proposed, such as Purity and Inverse Purity (usually combined via Van Rijsbergen's F measure), Clusters and class entropy, VI measure, Q_0, V-measure, Rand Statistic, Jaccard Coefficient, Mutual Information, etc. In Chapter 4 we will study and compare in detail these metrics.

Purity and Inverse Purity

Two standard extrinsic clustering evaluation metrics are Purity [ZK01] and Inverse Purity. Purity relates to the Precision measure and Inverse Purity to the Recall measure, which are well known in Information Retrieval. This measure focuses on the frequency of the most common category in each cluster, and rewards the clustering solutions that introduce less noise in each cluster. C being the set of clusters to be evaluated, L being the set of categories (manually annotated) and

n the number of clustered elements, Purity is computed by taking the weighted average of maximal precision values:

$$\text{Purity} = \sum_i \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j)$$

where the precision of a cluster C_i for a given category L_j is defined as:

$$\text{Precision}(C_i, L_j) = \frac{|C_i \bigcap L_j|}{|C_i|}$$

Inverse Purity focuses on the cluster with maximum recall for each category, rewarding the clustering solutions that gather more elements of each category in a corresponding single cluster. Inverse Purity is defined as:

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)$$

Works on name disambiguation [PPK05, AKE04] are closer to our task definition and evaluation methodology because they evaluate their disambiguation approach using extrinsic clustering metrics. In [PPK05] precision and recall of a clustering were evaluated using a matching metric. Al-Kamha [AKE04] used split and merge measures, which belong to the family of edit distance clustering metrics.

Other types of evaluation have been used for the evaluation of particular research approaches. Some works base their evaluation in classification results rather than coreference chains [FH04, MY03]. Fleischman [FH04] trained a classifier to decide wether a pair of contexts mentioning and ambiguous name are coreferent or not. The results are expressed in terms of the percentage of correct predictions made by the classifier (classification accuracy). Mann [MY03] used accuracy to measure the performance over two-class pseudo-ambiguous names.

As we have shown, there are many options to evaluate clustering algorithms, each one with different properties, and it is unclear which is the optimal metric for our problem. We will come back and solve this issue in Chapter 4

Part II

Benchmarking

Chapter 3

The WePS-1 Campaign

Developing of an evaluation framework that both provides the elements for a meaningful comparison of different approaches to the task and brings together the different research groups working on the problem constitutes a crucial element of this thesis. In this chapter, we will describe the first Web People Search evaluation campaign that was organised as part of the SemEval-1 evaluation exercise[1]. 16 teams presented their systems and were evaluated using standard clustering quality metrics and a common test collection. This pilot experience confirmed the interest of the research community in the Web People Search problem. It also showed us the difficulties of building a reliable testbed and the need for further research in evaluation metrics for the task.

This chapter is organized as follows: Section 3.1 presents a preliminary testbed together with the lessons we drew from it. Section 3.2 provides a description of the training and test data provided to the participants. Section 3.3 presents the evaluation measures and baseline approaches used in this campaign. The campaign design is briefly presented in Section 3.4. Section 3.5 gives a description of the participant systems and provides the evaluation results. Finally, Section 3.6 presents some conclusions.

3.1 Preliminary Testbed

Testbed components

For this evaluation campaign we initially delivered a preliminary testbed that was used as trial data set for the potential participants. The goal pursued during the development of this collection was to improve our overall understanding of the main characteristics of the problem in order to refine the annotation methodology[2]. The trial data consisted of (i) a corpus of web pages retrieved using people names as queries to web search engines; (ii) a classification of pages according to all the different people (with the same name) they refer to; (iii) manual annotations of

[1] SemEval-1 took place in 2007, followed by a workshop held in conjunction with ACL in Prague. SemEval-1 included 18 different tasks targeting the evaluation of systems for the semantic analysis of text (http://www.senseval.org/).

[2] This collection was also used to carry a first study of baseline approaches [AGV05], comparing full text and snippet document representations.

relevant information – found in the web pages – describing them (e-mail, image, profession, phone number, etc.); (iv) the results of applying a general purpose clustering algorithm to that annotated data, which serve as a baseline for the ambiguity resolution problem.

category	instances	per name	per person
home page	28	2.8	0.06
part of h.p.	15	1.5	0.03
reference p.	412	41.2	1
other	532	53.2	1.2
tags	instances	per name	per person
name	5,374	537.4	11.60
job	2,105	210.5	4.55
author of	1,823	182.3	3.94
description	438	43.8	0.95
date birth	387	38.7	0.84
date death	256	25.6	0.55
image	232	23.2	0.50
place birth	386	38.6	0.83
email	282	28.2	0.61
location	185	18.5	0.40
phone num	136	13.6	0.29
address	86	8.6	0.19
place death	85	8.5	0.18
fax num	37	3.7	0.08
Total	11,812	1,181.2	25.51

Table 3.1: WePS-1 preliminary testbed: annotation statistics

Testbed creation

The creation of this test collection consisted of the following steps:

1. Generating ten English people names, using random combinations of the most frequent first and last names in the 1990 U.S. Census

2. Collecting the first 100 web pages retrieved by the Google search engine for every (quoted) person name.

3. Grouping documents according to the person they refer to, for every person name.

4. Classifying every web document in the collection as a (i) homepage entry (ii) part of a homepage (iii) reference page (exclusively containing information about the person) and (iv) other.

5. Annotating all the occurrences of certain types of descriptive information: name, job, person image, date of birth/death, place of birth/death, email address, postal address, fax/phone number, location (where the person lives in), author of (e.g. books, paintings, patents...) and description (a brief definition of the person).

name	people
Ann Hill	55
Angela Thomas	36
Brenda Clark	23
Christine King	29
Helen Miller	38
Lisa Harris	30
Mary Johnson	54
Nancy Thompson	47
Samuel Baker	38
Sarah Wilson	62
Mean	41

Table 3.2: WePS-1 trial testbed: number of people for each name

Annotation results

Table 3.1 summarises the results of this exhaustive annotation process. A total of 11,812 text fragments were semantically annotated, with an average of 25.51 annotations per person. The ambiguity of this testbed is very high, with an average of 41 different people sharing each person name (see Table 3.2). This ambiguity indicates that they are very common names, but also that, none of them corresponds to any web celebrity in general. The most common tags are *name* (which includes name variants), *job* and *author of* (mostly titles of books and other written materials). The high frequency of *name* tags suggests that named entities could be a good document representation feature (this aspect is further explored in Chapter 8).

Note that there are few pages classified as *home page*, and even less pages tagged as *part of a home page*. Nonetheless, each identified person has, in average, one explicit description (*reference page*).

From the point of view of the WePS-1 campaign, the predominant feature in this preliminary testbed corpus corresponds to a high number of individuals in each document set (see Table 3.2). Although this preliminary effort was a cost-effective way of releasing data to play around with, we identified four main points for improvement: (i) in order to match the WePS task specifications, annotators should consider documents with internal ambiguity (overlapping clusters); (ii) non-person entities should be taken into account (e.g., a public library named after a person); (iii) introducing name sources other than the Census should provide more varied ambiguity cases; and, last but not least, (iv) the annotation of personal attributes is highly expensive, hence, for this first campaign, resources should concentrate on the annotation for the clustering task.

3.2 WePS-1 Testbed

The WePS-1 testbed was developed from the experience acquired in the previous test collection. In this second test collection, more name sources were included to provide a wider variety of ambiguity scenarios. Also, within-document ambiguity of names was taken into account during the annotation process (see Section 3.2.3). For the WePS-1 testbed, we decided to favour the manual clustering of a larger amount of names, instead of including page types and personal information extracted from pages as we did in the preliminary collection. The testbed was developed in two stages: first a test collection was prepared and handed over to the participants so that they could develop their systems (training data). Then, a different collection was annotated to evaluate each system (test data).

3.2.1 Training Data

Name selection

During the process of generating the corpus, the selection of the names plays an important role, which potentially conditions the degree of ambiguity that will be found later in the web search results. The reasons for this variability in the ambiguity of names are diverse and do not always correlate with the straightforward census frequency. A much more decisive feature is, for instance, the presence of famous entities sharing the ambiguous name with less popular people. As we are considering top search results, these can easily be monopolised by a single popular entity in the Internet.

In order to provide different ambiguity scenarios, we selected person names from different sources:

US Census. We reused the Web03 corpus [Man06], which contains 32 names randomly picked from the US Census, and proved to be well suited for the task.

Wikipedia. Another seven names were sampled from a list of ambiguous people names in the English version of Wikipedia. These were expected to have a few predominant entities (popular or historical), and, therefore, a lower level of ambiguity than the previous set.

ECDL. Finally, ten additional names were randomly selected from the Program Committee listing of a Computer Science conference (ECDL 2006). This set offers a potentially low ambiguity scenario (computer science scholars usually have a stronger Internet presence than other professional fields) together with the added value of the *a priori* knowledge of a domain-specific type of entity (scholar) present in the data.

Retrieval process

All datasets consist of collections of web pages obtained from the 100 top results for a person name query to an Internet search engine[3]. Note that 100 is an upper bound, since the URL returned by the search engine in some occasions no longer exists. Datasets consist of 17 people names and 1685 associated documents in total, that is, 99 documents per name in average. Each web page was downloaded and stored for offline processing. We also stored the basic metadata associated to each search result, including the original URL, title, position in the results ranking and its corresponding snippet generated by the search engine.

Ambiguity

After the annotation of this data (see Section 3.2.3), we found our predictions about the average ambiguity of each dataset were not completely accurate. In Table 3.3, we see that the ECDL-06 average ambiguity is indeed relatively low, except for the documents for “Thomas Baker”, standing as the most ambiguous name

[3] We used the Yahoo! Search Web Services API (http://developer.yahoo.com/search/web/).

Name	**entities**	**documents**	**discarded**
Wikipedia names			
John Kennedy	27	99	6
George Clinton	27	99	6
Michael Howard	32	99	8
Paul Collins	37	98	6
Tony Abbott	7	98	9
Alexander Macomb	21	100	14
David Lodge	11	100	9
Average	23.14	99.00	8.29
ECDL-06 Names			
Edward Fox	16	100	36
Allan Hanbury	2	100	32
Donna Harman	7	98	6
Andrew Powell	19	98	48
Gregory Crane	4	99	17
Jane Hunter	15	99	59
Paul Clough	14	100	35
Thomas Baker	60	100	31
Christine Borgman	7	99	11
Anita Coleman	9	99	28
Average	15.30	99.20	30.30
WEB03 Corpus			
Tim Whisler	10	33	8
Roy Tamashiro	5	23	6
Cynthia Voigt	1	405	314
Miranda Bollinger	2	2	0
Guy Dunbar	4	51	34
Todd Platts	2	239	144
Stacey Doughty	1	2	0
Young Dawkins	4	61	35
Luke Choi	13	20	6
Gregory Brennan	32	96	38
Ione Westover	1	4	0
Patrick Karlsson	10	24	8
Celeste Paquette	2	17	2
Elmo Hardy	3	55	15
Louis Sidoti	2	6	3
Alexander Markham	9	32	16
Helen Cawthorne	3	46	13
Dan Rhone	2	4	2
Maile Doyle	1	13	1
Alice Gilbreath	8	74	30
Sidney Shorter	3	4	0
Alfred Schroeder	35	112	58
Cathie Ely	1	2	0
Martin Nagel	14	55	31
Abby Watkins	13	124	35
Mary Lemanski	2	152	78
Gillian Symons	3	30	6
Pam Tetu	1	4	2
Guy Crider	2	2	0
Armando Valencia	16	79	20
Hannah Bassham	2	3	0
Charlotte Bergeron	5	21	8
Average	5.90	47.20	18.00
Global average	10.76	71.02	26.00

Table 3.3: WePS-1 training data

in the whole training. This phenomenon happens due to the fact that researchers have a strong presence on the Web. Wikipedia names have an average ambiguity of 23.14 entities per name, which is higher than those corresponding to the ECDL set. The WEB03 Corpus has the lowest ambiguity (5.9 entities per name), for two reasons: first, randomly picked names belong predominantly to the long tail of infrequent people names which, *per se*, have low ambiguity. As they are infrequent to find names, it implies that there are fewer documents returned by the search engine in average (47.20 per name), which also reduces the possibilities to find ambiguity.

3.2.2 Test Data

Name selection

For the test data we followed the same process described for the training. In the name selection we tried to maintain a similar distribution of ambiguity degrees and scenarios. For that reason, we randomly extracted 10 people names from the English Wikipedia and another 10 names from participants in the ACL-06 conference. In the case of the US census names, we decided to focus on relatively common names, thus avoiding the explained above problems.

Ambiguity

It is most remarkably that, after the annotation was finished (once the submission deadline had expired), we found a major increase in ambiguity degrees (Table 3.4) in all data sets. While we expected a raise in the case of the US census names, the other two cases just show that there is a high (and unpredictable) variability, and that much larger data sets are required to achieve representative population samples. We suspect, however, that the distribution of ambiguity does not follow a normal distribution, and therefore average ambiguity is not particularly meaningful to describe this type of datasets.

This has made the task particularly challenging for participants, because naive learning strategies (such as empirical adjustment of distance thresholds to optimise standard clustering algorithms) might be misled by the training set.

3.2.3 Annotation Process

Annotation of the data was performed separately in each set of documents related to an ambiguous name. Given a set of approximately 100 documents that mention the ambiguous name, the annotation consisted of the manual clustering of each document according to the actual entity that is referred to on it.

Annotation guidelines

When non person entities were found (for instance, organisation or places named after a person), the annotation was performed without any special rule. Typically, the annotator browses documents following the original ranking in the search results; after reading a document he will decide whether the mentions of the ambiguous name refer to a new entity or to an entity previously identified. We asked the annotators to concentrate first on mentions that strictly contained the search string, and then to pay attention to the co-referent variations of the name. For instance "John Edward Fox" or "Edward Fox Smith" would be valid mentions. "Edward J. Fox", however, breaks the original search string, and since we do not

Name	entities	documents	discarded
Wikipedia names			
Arthur Morgan	19	100	52
James Morehead	48	100	11
James Davidson	59	98	16
Patrick Killen	25	96	4
William Dickson	91	100	8
George Foster	42	99	11
James Hamilton	81	100	15
John Nelson	55	100	25
Thomas Fraser	73	100	13
Thomas Kirk	72	100	20
Average	56.50	99.30	17.50
ACL06 Names			
Dekang Lin	1	99	0
Chris Brockett	19	98	5
James Curran	63	99	9
Mark Johnson	70	99	7
Jerry Hobbs	15	99	7
Frank Keller	28	100	20
Leon Barrett	33	98	9
Robert Moore	38	98	28
Sharon Goldwater	2	97	4
Stephen Clark	41	97	39
Average	31.00	98.40	12.80
US Census Names			
Alvin Cooper	43	99	9
Harry Hughes	39	98	9
Jonathan Brooks	83	97	8
Jude Brown	32	100	39
Karen Peterson	64	100	16
Marcy Jackson	51	100	5
Martha Edwards	82	100	9
Neil Clark	21	99	7
Stephan Johnson	36	100	20
Violet Howard	52	98	27
Average	50.30	99.10	14.90
Global average	45.93	98.93	15.07

Table 3.4: WePS-1 test data

consider name variation detection, it will only be considered valid if it is co-referent to a valid mention. In order to perform the clustering, the annotator was asked to pay attention to objective data (biographical dates, related names, occupations, etc.) as well as to be conservative when making decisions.

Ambiguity

The final result is a complete clustering of the documents, where each cluster contains the documents that refer to a particular entity. Following the previous example, when dealing with documents mentioning the name “Edward Fox” the annotator found 16 different entities bearing that name. Note that there is no *a priori* knowledge about the number of entities to be discovered in a document set. This makes the annotation task specially difficult when there are many different entities and a high volume of scattered biographical information to be taken into account.

Documents filtering

In cases where the document does not offer enough information to decide whether it belongs to a cluster or if it is a new entity, the document is discarded from the evaluation process (but not from the dataset). Another common reason for discarding documents was the absence of the person name in the document, usually due to a mismatch between the search engine cache and the downloaded URL.

Overlapping clusters

Following this method, we found that, in many cases, different entities were mentioned using the ambiguous name within a single document. This circumstance produces overlapped clusters where one document is associated to several individuals at the same time. This was the case when a document mentions relatives with names that contain the ambiguous string (for instance "Edward Fox" and "Edward Fox Jr."). Another common case of within-document ambiguity is that of pages containing database search results, such as book lists from Amazon, actors from IMDB, etc. A similar case occurs in pages that explicitly analyse the ambiguity of a person name (Wikipedia "disambiguation" pages). Also, large genealogy pages turned out to be a frequent and very difficult case. In these documents, a large number of individuals can be referred with the same name and with very sparse information to assess them. The way this situation was handled – in terms of the annotation – was to assign each document to as many clusters as entities were referred to in it with the ambiguous name.

Inter-annotator agreement

After the campaign was finished, a second *gold standard* of test data was completed by a different annotator. In order to check the consistency of the annotators assesments, we compared the results evaluating the systems output with both *gold standards*. The results showed no significant differences in the ranking of systems.

3.3 Evaluation Methodology

Evaluation was performed in each document set (web pages mentioning an ambiguous person name) of the data distributed as test. The human annotation was used as the gold standard for the evaluation.

Metrics combination

Each system was evaluated using the standard Purity and Inverse Purity clustering measures (see Section 2.5). For the final ranking of systems we used the harmonic mean of Purity and Inverse Purity $F_{\alpha=_{0.5}}$. The F measure [Rij74] is defined as follows:

$$\mathrm{F} = \frac{1}{\alpha\frac{1}{\text{Purity}} + (1-\alpha)\frac{1}{\text{Inverse Purity}}}$$

$F_{\alpha=_{0.2}}$ is included as an additional measure giving more relevance to the Inverse Purity aspect. The rationale is that, for a search engine user, it should be easier to discard a few incorrect web pages in a cluster containing all the information needed than having to collect the relevant information across many different clusters. Therefore, achieving a high Inverse Purity should be rewarded more than having high Purity.

Baselines

Two simple baseline approaches were applied to the test data (Figure 3.1). The *ALL-IN-ONE* baseline provides a clustering solution where all the documents are assigned to a single cluster. This has the effect of always achieving the highest score in the Inverse Purity measure, because all classes have their documents in a single cluster. On the other hand, the Purity measure will be equal to the *precision* of the predominant class in that single cluster. The *ONE-IN-ONE* baseline gives another extreme clustering solution, where every document is assigned to a different cluster. In this case, Purity always gives its maximum value, while Inverse Purity decreases with larger classes.

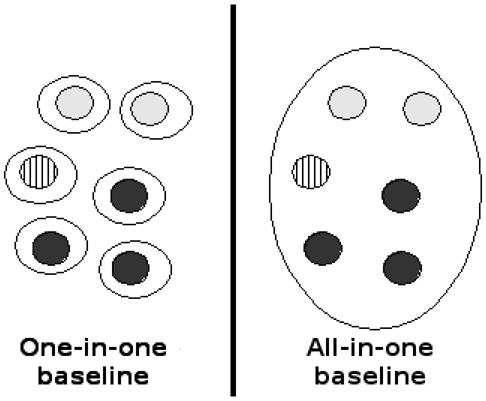


Figure 3.1: WePS-1 baseline systems

3.4 Campaign Design

Following the general SemEval guidelines, we prepared trial, training and test data sets for the task, which are described below.

Schedule

The schedule for the evaluation campaign was set by the SemEval organisation as follows: (i) release task description and trial data set[4]; (ii) release of training and test; (iii) participants send their answers to the task organisers; (iv) the task organisers evaluate the answers and send the results.

Evaluation period

The official evaluation period started with the simultaneous release of both training and test data, together with a scoring script with the main evaluation measures to be used. This period spanned five weeks, during which, teams were allowed to register and download the data. During that period, results for a given task had to be submitted no later than 21 days after downloading the training data and no later than 7 days after downloading the test data. Only one submission per team was allowed.

Data distribution

Training data included the downloaded web pages, their associated metadata and the human clustering of each document set, providing a development testbed for the participant's systems. We also specified the source of each ambiguous name in the training data (Wikipedia, ECDL conference and US Census). Test data only included the downloaded web pages and their metadata. This section of the corpus was used for the systems evaluation. Participants were required to send a clustering for each test document set.

[4] The task description and the initial trial data set were publicly released before the start of the official evaluation.

Finally, after the evaluation period was finished and all the participants sent their data, the task organisers sent the evaluation for the test data.

3.5 Results of the Evaluation Campaign

29 teams expressed their interest in the task. This number exceeded our expectations for this pilot experience, and confirmed the interest of the research community in this highly practical problem. 16 out of those teams submitted results within the deadline; their results are reported below.

Systems ranking

Table 3.5 presents the macro-averaged results obtained by the sixteen systems plus the two baselines on the test data. We found macro-average[5] preferable to micro-average[6] because it has a clear interpretation: if the evaluation measure is F, then we should calculate F for every test case (person name) and then average over all trials. The interpretation of micro-averaged F is less clear. The systems were ranked according to the scores obtained with the harmonic mean measure $F_{\alpha=_{0.5}}$ of Purity and Inverse Purity. If we consider only the participant systems, the average value for the ranking measure was 0.60 and its standard deviation 0.11.

Baselines

The good performance of the *ONE-IN-ONE* baseline system is indicative of the abundance of singleton entities (entities represented by only one document). This situation increases the Inverse Purity score for this system, thus giving a harmonic measure higher than the expected.

Metrics weighting effect

Results of systems are not substantially different considering $F_{\alpha=_{0.2}}$ or $F_{\alpha=_{0.5}}$, although there are some ranking swaps between close pairs of systems. Nevertheless, in the case of the baselines, not only they swap positions with the different F-measure α values, but the *ONE-IN-ONE* baseline moves from the middle of the table with $F_{\alpha=_{0.5}}$ to the end of it with $F_{\alpha=_{0.2}}$ (this effect will be analysed in detail in Chapter 5).

Features

Some common characteristics appear in the approaches presented by the participants. Regarding the features used to represent documents, the full document text is present in most systems, sometimes as the only feature [BAdR07, SO07] and sometimes in combination with others (see for instance [CM07b, PM07]). The most used features for the Web People Search task, however, are NEs [CM07b, PM07, ETY$^+$07, Sag07, KB07, RGY07, HN07, IXZ07, dVAdPSVD07]. The use of the link structure of web pages is less frequent [CM07b, LHF07, dVAdPSVD07]. Finally, biographical facts (like place/date of birth, etc.) are extracted by only one of the participants [LHF07].

Clustering algorithms

The most frequent clustering algorithm among participants was HAC [CM07b, LHF07, ETY$^+$07, Sag07, EE07, HN07, IXZ07, dVAdPSVD07] combined with single linkage [CM07b, Sag07, EE07]. As we saw in Section 2.4.3, this type of linkage tends to cluster many documents together, and hence provided better results on low ambiguity names. The similarity metric of choice in many systems was the

[5] Macro-average F consists of computing F for every test set (person name) and then averaging over all test sets.

[6] Micro-average F consists of computing the average P and IP (over all test sets) and then calculating F with these figures.

cosine [ETY$^+$07, Sag07, dVAdPSVD07]. Finally, regarding the clustering stopping criteria, most participants decided to train a value on the development datasets and apply it to the test data [CM07b, ETY$^+$07, Sag07, RGY07, HN07]. Due to the differences in the amount of ambiguity found in training and test data, this strategy probably penalised systems with less robust response to variations in the clustering threshold.

		Macro-averaged Scores			
		F-measures			
rank	team-id	$\alpha = .5$	$\alpha = .2$	Pur	Inv_Pur
1	CU_COMSEM	.78	.83	.72	.88
2	IRST-BP	.75	.77	.75	.80
3	PSNUS	.75	.78	.73	.82
4	UVA	.67	.62	.81	.60
5	SHEF	.66	.73	.60	.82
6	FICO	.64	.76	.53	.90
7	UNN	.62	.67	.60	.73
	ONE-IN-ONE	**.61**	**.52**	*1.00*	*.47*
8	AUG	.60	.73	.50	.88
9	SWAT-IV	.58	.64	.55	.71
10	UA-ZSA	.58	.60	.58	.64
11	TITPI	.57	.71	.45	.89
12	JHU1-13	.53	.65	.45	.82
13	DFKI2	.50	.63	.39	.83
14	WIT	.49	.66	.36	.93
15	UC3M_13	.48	.66	.35	.95
16	UBC-AS	.40	.55	.30	.91
	ALL-IN-ONE	**.40**	**.58**	*.29*	*1.00*

Table 3.5: WePS-1 team ranking

3.6 Conclusions

The WePS-1 task ended with considerable success in terms of participation. In addition to that, all the collected and annotated dataset was released [7] as a benchmark for Web People Search systems.

Ambiguity variability

The variability across test cases has proved to be large and unpredictable. A system that works well with the names in our testbed may not be reliable in practical, open search situations. Partly because of that reason, our testbed happened to be unintentionally challenging for systems, with a large difference between the average ambiguity in training and test datasets. Moreover, it is clear from this experience that building a reliable testbed for the task is not simple, insofar as the same name source does not guarantee a consistent ambiguity across sets of names.

Evaluation measures

In view of the results obtained, we found it necessary to think about specific

[7] http://nlp.uned.es/weps

evaluation measures beyond standard clustering metrics such as Purity and Inverse Purity. These metrics are not tailored to the task and were not designed to handle multiple classification of elements. Furthermore, we found that different values of α in F_α have an effect in the ranking of the baseline systems when combining metrics. The very fact that a baseline can go from the middle of the ranking to the bottom depending on the combining criteria has important implications in the way we measure the contribution of systems to the problem. The two main questions we draw from this experience were: (i) are Purity and Inverse Purity the most appropriate metrics for the WePS task and for clustering problems where overlapping elements are allowed? (ii) how does the combination of evaluation metrics affect the ranking of systems in this task? Chapters 4 and 5 will study these questions in detail.

Chapter 4

Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints

Given the results obtained in the WePS-1 campaign, this chapter deals with the appropriateness of different clustering metrics in the context of WePS, focusing on whether or not they satisfy a number of formal constraints. There has already been some attempts to analyse the properties of clustering evaluation metrics. For instance, in Meila's work [Mei03], a specific metric based on entropy is tested against twelve mathematical constraints. The immediate question is why twelve constraints, or why precisely this set. In this chapter, we also define properties/constraints that any clustering metric should satisfy, but trying to observe a number of rules:

1. Constraints should be intuitive and clarify the limitations of each metric. This should allow the system developer to identify which constraints must be considered for the specific task at hand.

2. It should be possible to demonstrate formally which metrics satisfy which properties (some previously proposed constraints can only be checked empirically).

3. The constraints should discriminate metric families, grouped according to their mathematical foundations, pointing out the limitations of each metric family rather than individual metric variants. This analysis is useful for metric developers, since it ensures that a deeper research in a specific kind of metrics will not solve certain constraints.

We have found four basic formal constraints for clustering evaluation metrics that satisfy the above-mentioned requisites. This set of constraints covers all quality aspects that have been proposed in previous works.

Once the formal conditions have been defined, we have checked all major evaluation metrics, finding that metrics from the same family behave likewise according to these formal constraints. In particular, we found BCubed metrics

(BCubed precision and BCubed recall) to be the only ones that satisfy all our proposed constraints. However, our work opens the possibility of choosing other metrics when, for a particular clustering task, some of the restrictions do not hold and other metric can be found to be best suited according, for instance, to its scale properties.

According to the main characteristics of the WePS task, we also extend the analysis to the problem of overlapping clustering, where items can simultaneously belong to more than one cluster. We show that most metric families cannot capture certain quality properties of overlapping clusters, and no individual metric proposed so far is fully satisfactory. For this reason, we propose an extension of BCubed metrics which satisfies all our formal requirements.

We review the results obtained in WePS-1 and introduce an additional baseline (the so-called "cheat system") that exploits the possibility of overlapping clusters in this task. We found that, unlike Purity and Inverse Purity, the proposed BCubed extension is able to discriminate and penalise an undesirable, "cheat" clustering solution.

The remainder of the chapter is structured as follows: In Section 4.1, we introduce and discuss the set of proposed formal constraints. In Section 4.2, we analyse current metrics according to our proposed constraints. In Section 4.3, we compare our formal constraints with previously proposed constraint sets in the literature. In Section 4.4, we address the evaluation of overlapping clustering and propose the extension to BCubed metrics to handle the problem adequately. Section 4.5 ends with the main conclusions of our study.

4.1 Formal Constraints on Evaluation Metrics for Clustering Tasks

Methodology In order to define formal restrictions on any suitable metric, we will employ the following methodology: each formal restriction consists of a pattern (D_1, D_2) of system output pairs, where D_2 is assumed to be a better clustering option than D_1 according to our intuition. The restriction on any metric Q is then $Q(D_1) < Q(D_2)$. We have identified four basic constraints which we discuss below.

4.1.1 Constraint 1: Cluster Homogeneity

This constraint is an essential quality property that has already been proposed in previous research [RH07]. In terms of WePS, this constraint states that two sets of documents referring to two different people should be separated. We formalise it as follows: Let S be a set of items belonging to categories $L_1 \dots L_n$. Let D_1 be a cluster distribution with one cluster C containing items from two categories L_i, L_j. Let D_2 be a distribution identical to D_1, except for the fact that the cluster C is split into two clusters containing the items with category L_i and the items with category L_j, respectively. Then an evaluation metric Q must satisfy $Q(D_1) < Q(D_2)$.

This constraint is illustrated in Figure 4.1. It constitutes a very basic restriction which states that the clusters must be homogeneous, i.e. they should not mix items belonging to different categories.

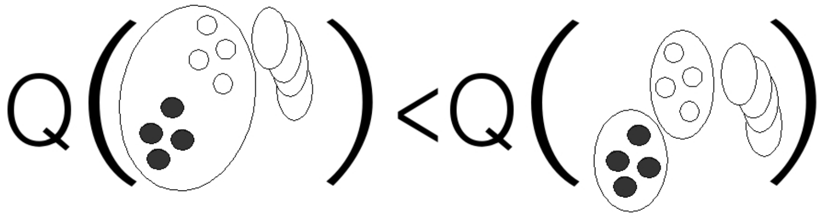


Figure 4.1: Clustering evaluation metric constraint 1: Cluster Homogeneity

4.1.2 Constraint 2: Cluster Completeness

The counterpart to the first constraint is that items belonging to the same category should be grouped in the same cluster[1]. Regarding the WePS task, this means that documents mentioning the same person should be found in the same cluster. In other words, different clusters should contain items from different categories. We can model this notion with the following formal constraint: Let D_1 be a distribution such as that two clusters C_1, C_2 only contain items belonging to the same category L. Let D_2 be an identical distribution, except for the fact that C_1 and C_2 are merged into a single cluster. Then D_2 is a better distribution: $Q(D_1) < Q(D_2)$. This restriction is illustrated in Figure 4.2.

Constraints 1 and 2 are the most basic restrictions that any evaluation metric must hold and refer to the basic goals of a clustering system: keeping items from the same category together, and keeping items from different categories apart. In the next section we will see that, surprisingly, some of the most popular metrics fail to satisfy these constraints.

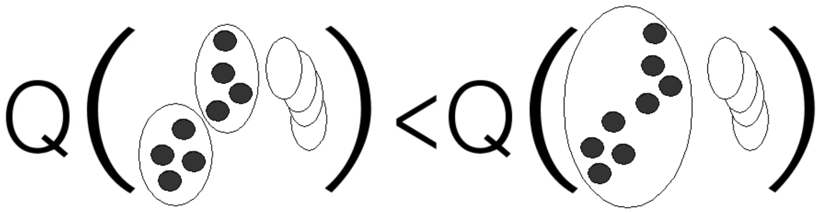


Figure 4.2: Clustering evaluation metric constraint 2: Cluster Completeness

4.1.3 Constraint 3: Rag Bag

An additional intuition on the clustering task is that introducing disorder into a disordered cluster is less harmful than introducing disorder into a clean cluster. In

[1] As in [RH07], we use the term “Completeness” to avoid “Compactness”, which, in the clustering literature, is used as an internal property of clusters which refers to minimising the distance between the items of a cluster.

terms of the WePS task, this means that it is preferable to have one very noisy cluster with documents of infrequent people than to spread this documents in clean clusters that represent more popular individuals. Indeed, for many practical situations, it is useful to have a "rag bag" of items which cannot be grouped with other items (think of "miscellaneous", "other", "unclassified" categories). It is then assumed that such a set contains items of diverse genre. Of course, in any case, a perfect clustering system should identify that these items cannot be grouped, and, therefore, belong to different categories. But when comparing sub-optimal solutions, the intuition is that it is preferable to have clean sets plus a "rag bag" than having sets with a dominant category plus additional noise.

The boundary condition, which constitutes our third restriction, can be stated as follows: Let C_{clean} be a cluster with n items belonging to the same category. Let C_{noisy} be a cluster merging n items from unary categories (there exists just one sample for each category). Let D_1 be a distribution with a new item from a new category merged with the highly clean cluster C_{clean}, and D_2 another distribution with this new item merged with the highly noisy cluster C_{noisy}. Then $Q(D_1) < Q(D_2)$ (see Figure 4.3).

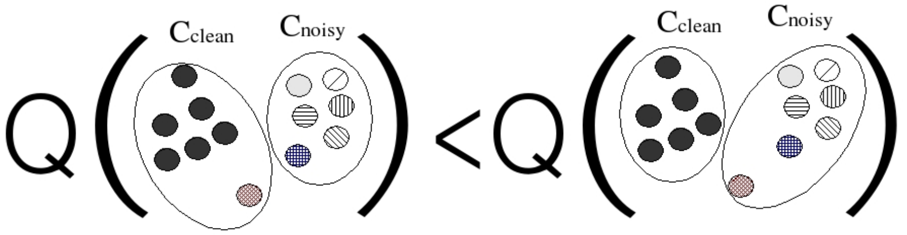


Figure 4.3: Clustering evaluation metric constraint 3: Rag Bag

4.1.4 Constraint 4: Clusters Size vs. Quantity

A small error in a big cluster would be preferable to a large number of small errors in small clusters. This constraint is particularly relevant in WePS. A cluster representing a popular person is likely to have more redundant information on it than a cluster of a person mentioned in few documents. For this reason, the loss of one document is not as critical for celebrities as it is in the case of more infrequent individuals. This property is partially related with the fourth property in [Mei03], called in [RH07] *n-invariance*. We state a boundary condition related to this notion saying that separating one item from its class of $n > 2$ members is preferable to fragmenting n binary categories (see Figure 4.4).

Formally, let us consider a distribution D containing a cluster C_l with $n + 1$ items belonging to the same category L, and n additional clusters $C_1 \ldots C_n$, each of them containing two items from the same category $L_1 \ldots L_n$. If D_1 is a new distribution similar to D, where each C_i is split in two unary clusters, and D_2 is a distribution similar to D, where C_l is split in one cluster of size n and one cluster of size 1, then $Q(D_1) < Q(D_2)$.

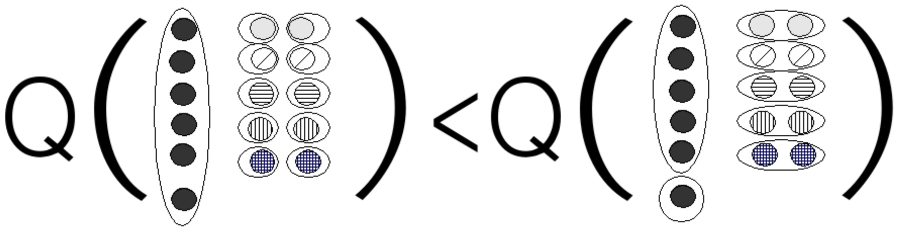


Figure 4.4: Clustering evaluation metric constraint 4: Size vs. Quantity

4.2 Comparison of Evaluation Metrics

Given the large number of metrics proposed for the clustering task, we will group them in four families and try to test properties inherent to the kind of information each family uses.

4.2.1 Evaluation by Set Matching

This metric family was identified as such in [Mei03]. They share the feature of assuming a one to one mapping between clusters and categories, and they rely on the precision and recall concepts inherited from Information Retrieval.

Purity

The most popular measures for cluster evaluation are Purity, Inverse Purity and their harmonic mean (F measure). Purity [ZK01] focuses on the frequency of the most common category into each cluster. Being C the set of clusters to be evaluated, being L the set of categories (reference distribution) and being n the number of clustered items, Purity is computed by taking the weighted average of maximal precision values:

$$\text{Purity} = \sum_i \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j)$$

where the precision of a cluster C_i for a given category L_j is defined as:

$$\text{Precision}(C_i, L_j) = \frac{|C_i \bigcap L_j|}{|C_i|}$$

Inverse Purity

Purity penalises the noise in a cluster, but it does not reward grouping items from the same category together. If we just make one cluster per item, we reach trivially a maximum Purity value. Inverse Purity focuses on the cluster with maximum recall for each category. Inverse Purity is defined as:

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)$$

Inverse Purity rewards grouping items together, but it does not penalise mixing items from different categories; we can reach a maximum value for Inverse Purity by making a single cluster with all items.

F measure

A more robust metric can be obtained by combining the concepts of Purity and

Inverse Purity, matching each category with the cluster that has a highest combined precision and recall, using Van Rijsbergen's F measure [Rij74, LA99, SKK00]:

$$F = \sum_i \frac{|L_i|}{n} \text{max}_j \{F(L_i, C_j)\}$$

where

$$F(L_i, C_j) = \frac{2 \times \text{Recall}(L_i, C_j) \times \text{Precision}(L_i, C_j)}{\text{Recall}(L_i, C_j) + \text{Precision}(L_i, C_j)}$$

$$\text{Recall}(L, C) = \text{Precision}(C, L)$$

Constraint 2

One common issue with these type of metrics is that they cannot satisfy constraint 2 (cluster completeness): as each category is judged only by the cluster which has more items belonging to it, changes in other clusters are not detected. This problem has been previously identified (see [Mei03] or [RH07]). An example can be seen in Figure 4.5: clusters C_1 and C_2 contain items from the same category, so merging them should improve the quality of the distribution (Category completeness constraint). And yet, Purity does not satisfy this constraint in general. Both Inverse Purity and F measure are not sensible to this case, as the cluster with maximal precision and F measure over the category of black circles is C_3.

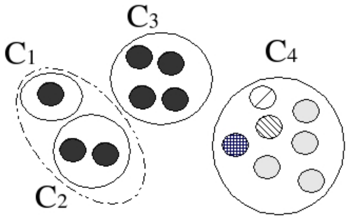
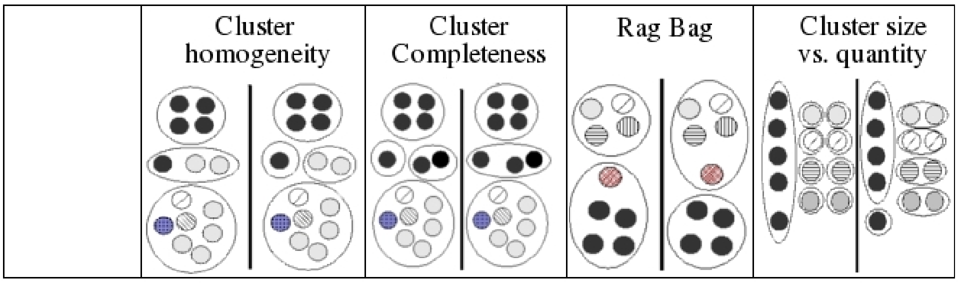


Figure 4.5: Clustering constraint 2 example

Figure 4.6 shows the results of computing several metrics in four test cases instantiating all four constraints. There, we can see counterexamples showing that no metric in this family satisfies constraints 2 and 3, and even constraint 1 is only satisfied by the Purity measure.

4.2.2 Metrics Based on Counting Pairs

Considering statistics over pairs of items is another approach to define evaluation metrics for clustering [HBV01, Mei03]. Let SS be the number of pairs of items belonging to the same cluster and category; SD the number of pairs belonging to the same cluster and different category; DS the number of pairs belonging to different cluster and the same category, and DD the number of pairs belonging to different category and cluster. SS and DD are "good choices", and DS, SD are "bad choices".



Metrics based on set matching												
Purity	**0.71**	**0.78**	✓	0.78	0.78	×	0.55	0.55	×	1	1	×
Inv. Purity	0.78	0.78	×	0.78	0.78	×	1	1	×	**0.69**	**0.92**	✓
F-measure	0.63	0.63	×	0.62	0.62	×	0.61	0.61	×	**0.79**	**0.96**	✓
	OK			**FAIL**			**FAIL**			**OK**		

Metrics based on counting pairs												
Rand	**0.68**	**0.7**	✓	**0.68**	**0.7**	✓	0.72	0.72	×	0.95	0.95	×
Jaccard	**0.31**	**0.32**	✓	**0.31**	**0.35**	✓	0.37	0.37	×	0.71	0.71	×
F&M	**0.47**	**0.49**	✓	**0.47**	**0.52**	✓	0.61	0.61	×	0.84	0.84	×
	OK			**OK**			**FAIL**			**FAIL**		

Metrics based on Entropy												
-Entropy	**-1.03**	**-0.8**	✓	-0.83	-0.83	×	-1.29	-1.29	×	0	0	×
-Class Entr.	-0.65	-0.65	×	**-0.69**	**-0.49**	✓	0	0	×	**-0.61**	**-0.28**	✓
Mutual Inf.	**0.84**	**1.03**	✓	1	1	×	0.99	0.99	×	2.19	2.19	×
	OK			**OK**			**FAIL**			**OK**		

Metrics based on editing distance												
Edit dist.	(steps) 7	7	×	**7**	**6**	✓	6	6	×	**9**	**6**	✓
	FAIL			**OK**			**FAIL**			**OK**		

Bcubed metrics												
Precision	**0.59**	**0.69**	✓	0.62	0.62	×	**0.52**	**0.64**	✓	1	1	×
Recall	0.69	0.69	×	**0.71**	**0.75**	✓	1	1	×	**0.64**	**0.81**	✓
F(BCubed)	**0.63**	**0.69**	✓	**0.66**	**0.67**	✓	**0.68**	**0.78**	✓	**0.78**	**0.89**	✓
	OK			**OK**			**OK**			**OK**		

Figure 4.6: Satisfaction of Formal Constraints: Examples

Rand statistic, Jaccard Coefficient, Folkes and Mallows

Some of the metrics using these figures are:

$$\text{Rand statistic R} = \frac{(SS + DD)}{SS + SD + DS + DD}$$

$$\text{Jaccard Coefficient J} = \frac{SS}{SS + SD + DS}$$

$$\text{Folkes and Mallows FM} = \sqrt{\frac{SS}{SS + SD}\frac{SS}{SS + DS}}$$

Constraints 3 and 4

It is not hard to see that this type of metrics satisfies the first two constraints; but they do not satisfy constraints 3 and 4. Figure 4.6 shows some counterexamples. Take for instance the example for constraint 4: The number of pairs affected by the fragmentation in both distributions is the same. In the first case, one black item is separated from the other four black items. In the second case, n correct binary clusters are fragmented into unary clusters. Therefore, the values for DD, SS, SD and DS are the same in both distributions. The problem here is that the number of item pairs in a cluster has a quadratic dependence with the cluster size, and so changes in bigger clusters have an excessive impact on this type of measures.

4.2.3 Metrics based on entropy

Entropy

The Entropy of a cluster [SKK00, Gho03] reflects how the members of the k categories are distributed within each cluster; the global quality measure is again computed by averaging the entropy of all clusters:

$$\text{Entropy} = -\sum_j \frac{n_j}{n} \sum_i P(i,j) \times \log_2 P(i,j)$$

being $P(i,j)$ the probability of finding an element from the category i in the cluster j, n_j the number of items in cluster j and n the total number of items in the distribution. Other metrics based on entropy have been also defined, for instance, "class entropy" [BHK02], "variation of information" [Mei03], "Mutual Information" [XLG03], Q_o [Dom01] or "V-measure" [RH07].

Constraints 2, 3 and 4

Figure 4.6 shows counterexamples for some of these measures in all constraints: entropy and mutual information fail to satisfy constraints 2, 3, 4, and class entropy, in turn, fails to do it with constraints 1 and 3. In particular, the Rag Bag constraint cannot be satisfied by any metric based on entropy: conceptually, the increase of entropy when an odd item is added is independent from the previous grade of disorder in the cluster. Therefore, it is equivalent to introduce a wrong item in a clean cluster or in a noisy cluster.

Let us formalise our argument: Let C be a cluster with n items. Then the entropy would be computed as

$$E_C = \sum_i P_i \log P_i$$

where P_i is the probability of finding an element of the category i in the cluster. Let C' be the same cluster adding an item that is unique in its own category and was previously isolated. Then

$$E_{C'} = \frac{1}{n+1} \log \frac{1}{n+1} + \sum_i \frac{nP_i}{n+1} \log \frac{nP_i}{n+1}$$

being n the number of items in the cluster. Operating:

$$E_{C'} = \frac{1}{n+1} \log \frac{1}{n+1} + \frac{n}{n+1} \sum_i [P_i * (\log \frac{n}{n+1} + \log P_i)] =$$

$$= \frac{1}{n+1} \log \frac{1}{n+1} + \frac{n}{n+1} [\log \frac{n}{n+1} \sum_i P_i + \sum_i P_i * \log P_i]$$

Since $\sum_i P_i = 1$

$$E_{C'} = \frac{1}{n+1} \log \frac{1}{n+1} + \frac{n}{n+1} [\log \frac{n}{n+1} + E_C]$$

In other words, the increase in entropy depends exclusively from n; the homogeneity or heterogeneity of the cluster does not affect the result.

4.2.4 Evaluation Metrics Based on Edit Distance

Pantel [PL02a] presents an evaluation metric based on transformation rules which opens a new family of metrics. The quality of a clustering distribution is related to the number of transformation rules that must be applied in order to obtain the ideal distribution (one cluster for each category). This set of rules includes merging two clusters and moving an item from one cluster to another. Their metric (which we do not fully reproduce here for lack of space) does not satisfy constraints 1 and 3 (see counterexamples in Figure 4.6). Indeed, metrics based on edit distance cannot satisfy the Rag Bag constraint: regardless of the actual location where we introduce the noisy item, the distance edit is always one movement, hence the quality of both distributions will always be the same.

4.2.5 BCubed: a Mixed Family of Metrics

BCubed precision and recall

We have already seen that none of the previous metric families satisfy all our formal restrictions. The most problematic constraint is *Rag Bag*, which is not satisfied by any of them. However, BCubed precision and recall metrics [BB98b] satisfy all constraints. Unlike Purity or Entropy metrics, which compute the quality of each cluster and category independently, BCubed metrics decompose the evaluation process estimating the precision and recall associated to each item in the distribution[2]. The item precision represents how many items in the same cluster belong to its

[2] Although BCubed has been already presented in Section 2.5, in this section we will provide a more detailed description

category. Symmetrically, the recall associated to one item represents how many items from its category appear in its cluster. Figure 4.7 illustrates how the precision and recall of one item is computed by BCubed metrics.

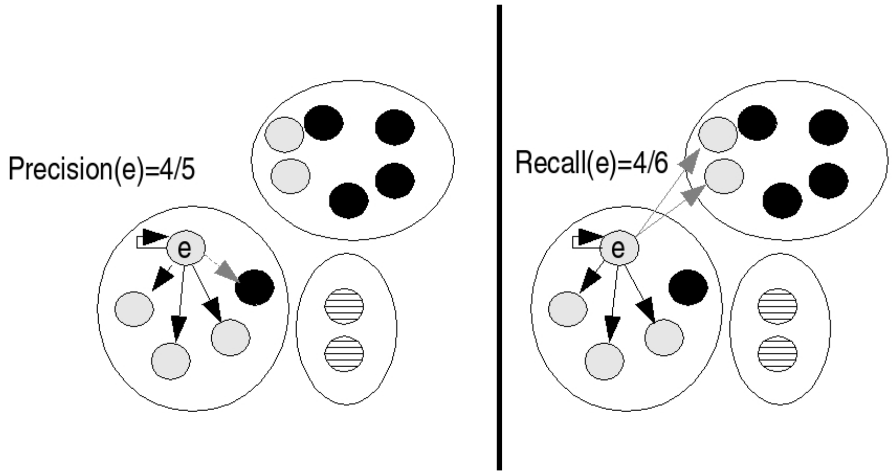


Figure 4.7: Example of computing the BCubed precision and recall for one item

Users point of view

From a user's point of view, BCubed represents the clustering system effectiveness when the user explores the rest of items in the cluster after accessing one reference item. If this item had a high BCubed recall, the user would find most of related items without leaving the cluster. If the reference item had a high precision, the user would not find noisy items in the same cluster. The underlying difference with Purity or Entropy measures is that the adequacy of items depends on the reference item rather than on the predominant category within the cluster.

Correctness

For our study BCubed can be described in terms of a function. Being $L(e)$ and $C(e)$ the category and the cluster of an item e, we can define the correctness of the relation between e and e' in the distribution as:

$$\text{Correctness}(e,e') = \begin{cases} 1 \text{ iff } L(e) = L(e') \longleftrightarrow C(e) = C(e') \\ 0 \text{ otherwise} \end{cases}$$

That is, two items are correctly related when they share a category if and only if they appear in the same cluster. BCubed precision of an item is the proportion of correctly related items in its cluster (including itself). The overall BCubed precision is the averaged precision of all items in the distribution. Since the average is calculated over items, it is not necessary to apply any weighting according to the size of clusters or categories. The BCubed recall is analogous, replacing "cluster" with "category". Formally:

$$\text{Precision BCubed} = \text{Avg}_e[\text{Avg}_{e'.C(e)=C(e')}[\text{Correctness}(e,e')]]$$

$$\text{Recall BCubed} = \text{Avg}_e[\text{Avg}_{e'.L(e)=L(e')}[\text{Correctness}(e,e')]]$$

BCubed vs. other metrics

BCubed combines the best features from other metric families. Just like Purity or Inverse Purity, it is inspired on precision and recall concepts, being easily interpretable. As it occurs with entropy based metrics, it considers the overall disorder of each cluster, not just the predominant category, thus satisfying restrictions 1 and 2 (*homogeneity* and *completeness*). Both BCubed and metrics based on counting pairs consider the relation between pairs of items. However, in BCubed metrics the overall average is computed over single items and so the quadratic effect produced by the cluster size disappears, therefore satisfying restriction 4, *cluster size vs. cluster quantity*. In addition, unlike all other metrics, BCubed also satisfies the *Rag Bag* constraint.

Let us verify the four constraints:

- **Cluster homogeneity constraint:** Splitting a cluster that mixes two categories into two "pure" clusters increases the BCubed precision, and does not affect recall (see Figure 4.1).

- **Cluster completeness constraint:** Unifying two clusters which contain only items from the same category increases the BCubed recall measure, and the precision of joined items remains maximal (see Figure 4.2).

- **Rag Bag constraint:** Let us suppose that we have an item (unique in its category) in an isolated cluster. Introducing the item in a clean cluster of n items (D_1, Figure 4.3) decreases the precision of each item in the clean cluster from 1 to $\frac{n}{n+1}$, and the precision of the item just inserted from 1 to $\frac{1}{n+1}$. So, being N_{tot} the total number of items in the distribution, while the recall is not affected in any way, the overall precision decreasing in the distribution is thus:

$$\text{DEC}_{D_1} = \frac{1 + n * 1}{N_{\text{tot}}} - \frac{\frac{1}{n+1} + n * \frac{n}{n+1}}{N_{\text{tot}}} = \frac{\frac{2n}{n+1}}{N_{\text{tot}}} \simeq \frac{2}{N_{\text{tot}}}$$

On the other hand, introducing the same item in a noisy cluster (D_2, Figure 4.3) decreases the precision of the isolated item from 1 to $\frac{1}{n+1}$, and the items in the noisy cluster from $\frac{1}{n}$ to $\frac{1}{n+1}$. This being so, the overall decrease in the distribution is smaller:

$$\text{DEC}_{D_2} = \frac{1 + n * \frac{1}{n}}{N_{\text{tot}}} - \frac{1 * \frac{1}{n+1} + n * \frac{1}{n+1}}{N_{\text{tot}}} = \frac{1}{N_{\text{tot}}} < \text{DEC}_{D_1}$$

- **Cluster Size vs. Quantity:** In the distribution D_1 from Figure 4.4, $2n$ items decrease their recall in 50%. That represents an overall decrease of:

$$\text{DEC}_{D_1} = \frac{2n}{N_{\text{tot}}} - \frac{2n\frac{1}{2}}{N_{\text{tot}}} = \frac{n}{N_{\text{tot}}}$$

On the other hand, in the distribution D_2, the recall of n items decreases from 1 to $\frac{n}{n+1}$, and the recall of one item decreases from 1 to $\frac{1}{n+1}$. So the overall decrease in the distribution is smaller:

$$\text{DEC}_{D_2} = \frac{n+1}{N_{\text{tot}}} - \frac{n\frac{n}{n+1} + \frac{1}{n+1}}{N_{\text{tot}}} = \frac{\frac{2n}{n+1}}{N_{\text{tot}}} \simeq \frac{2}{N_{\text{tot}}} < \text{DEC}_{D_1}$$

In conclusion, BCubed metrics considered altogether satisfy all our formal constraints. BCubed precision covers restrictions 1 and 3. BCubed recall covers constraints 2 and 4. Table 4.6 contains a sample of clustering distribution pair for each formal constraint. The table shows that BCubed precision and recall metrics cover all of them.

BCubed metrics combination

A remaining issue is how to combine both in a single evaluation metric. According to our formal constraints, any averaging criterion for combining metrics satisfies all formal constraints when these are satisfied by the combined metrics in isolation. This issue is due to the fact that our formal constraints are defined in such a way that each one represents an isolated quality aspect. When a metric does not cover a specific quality aspect, the associated restriction is not affected.

A standard way of combining metrics is Van Rijsbergen's F [Rij74]. It is computed as follows:

$$F(R,P) = \frac{1}{\alpha(\frac{1}{P}) + (1-\alpha)(\frac{1}{R})}$$

being R and P two evaluation metrics and being α and $(1-\alpha)$ the relative weight of each metric ($\alpha = 0.5$ leads to the harmonic average of P,R). The last row in Figure 4.6 shows the results when applying $F_{\alpha=0.5}$ over BCubed Precision and Recall, thus satisfying all formal constraints.

4.3 Related Work: Other Proposed Formal Constraints

But still, are four constraints enough? We do not have a formal argument supporting this, but at least we can compare our set of constraints with previous related proposals.

4.3.1 Dom's Constraints

In [Dom01], Dom proposes five formal constraints. These were extended to seven in [RH07]. The author decomposes the clustering quality into a set of parameters: the number of "noise" and "useful" clusters, the number of "noise" and "useful" categories, and three components of the error mass probability. "Noise" clusters are those that contain items equally from each category. On the opposite side, "useful" clusters have a predominant category. The error mass probability measures to what extent single items are not included in the corresponding "useful" cluster.

The formal constraints consist of testing if specific parameter configurations do lead to a decrease of quality according to the metric over a random set of clustering samples. Basically, these formal constraints capture the idea that a clustering is worse when: (1) the number of useful clusters varies away from the number of categories, (2) the number of noise clusters increases and (3) the error mass parameters increase. Roughly speaking, these ideas are directly correlated with our constraints. For instance, *Cluster Homogeneity* and *Cluster Completeness* implies respectively a decrease and increase of useful clusters regarding the number of categories.

But Dom's restrictions reflect intermediate situations which are not considered explicitly by our formal constraints, since we defined them using boundary conditions. Theoretically speaking, this implies that a metric satisfying our constraints may not satisfy Dom's constraints. However, all metric drawbacks which are detected by Dom's constraints are also detected by our set.

In particular, the results in [RH07] shows that metrics based on Entropy satisfy all these formal constraints, and metrics based on counting pairs fail at least in two properties. In order to explain this result, the authors state that "the number of noise classes or clusters can be increased without reducing any of these metrics" when counting pairs. We believe that our constraint 4 *Cluster size vs. quantity* provides a more in-depth explanation. Increasing the number of noise clusters while fixing the rest of parameters produces smaller clusters (see Figure 4.8). Metrics based on counting pairs give a quadratic relevance to erroneously joined items in bigger clusters, increasing the score when splitting noise clusters. For instance, in Figure 4.8, the right distribution introduces 9 correct item associations at the expense of 27 incorrect pairs. Metrics based on entropy, on the contrary, satisfy the *Cluster size vs. quantity* constraint, thus overcoming this problem.

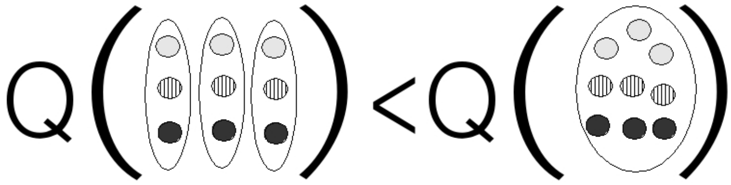


Figure 4.8: More noise clusters implies less quality

Dom's constraints have some drawbacks in relation to our meta-evaluation framework:

1. Dom's constraints detect less limitations than our constraints. For instance, they do not detect drawbacks of entropy-based metrics. Our constraints, however, detect that entropy based metrics do not satisfy the *Rag Bag* constraint.

2. Each Dom's constraint is related to several quality aspects. For instance, the mass error or the number of noise clusters are related simultaneously with the concepts of *homogeneity*, *completeness* and *Cluster Size vs. Quantity*. Therefore, it is not easy to identify the need for satisfying specific constraints in specific clustering applications.

3. It is not easy to prove formally that an evaluation metric satisfies Dom's constraints. As a matter of fact, these restrictions were tested by evaluating

"random" clustering distributions. Our constraints, however, can be formally verified for each family of metrics.

4.3.2 Meila's Constraints

Meila [Mei03] proposes an entropy-based metric (*Variation Information* or VI) and enumerates twelve desirable properties associated with this metric. Properties 1-3, for instance, are positivity, symmetry and triangle inequality, which, taken altogether, imply that VI is a proper *metric* on clusterings. Most of these properties are not directly related to the quality aspects captured by a metric, but rather on other intrinsic features such as scale properties or computational cost. The most relevant properties for our discussion are:

- Property 4 is related with the *cluster size vs. quantity* constraint. It states that the quality of a distribution depends on the relative sizes of clusters but not on the number of points in the data set. Metrics based on counting pairs do not satisfy this property since the number of item pairs increase quadratically regarding the number of items in the distribution.

- Property 7 states that splitting or merging smaller clusters has less impact than splitting or merging larger ones. It states also that the variation in the evaluation measure is independent of anything outside the clusters involved. Although this property is desirable, in practice, all metrics discussed here do satisfy it. Therefore, it does not provide much information about what metrics are more suitable for evaluation purposes.

- Properties 10 and 11 are associated to the idea that splitting all clusters according to item categories improves the results. This corresponds with the formal constraint that we call *Cluster Completeness*.

In short, while Meila's properties are an in-depth characterisation of the VI metric, they do not suggest any additional constraint to our original set. Indeed, the VI metric proposed by Meila does not satisfy our constraint 3 (Rag Bag), being an entropy-based metric (see Section 4.3).

4.4 Evaluation of Overlapping Clustering

Overlapping clusters in WePS

The metrics discussed so far do not (at least explicitly) handle clustering scenarios where the same item can be assigned to more than one cluster/category (overlapping clustering). This is a characteristic that we find in the case of WePS. A web search result for a person name can contain mentions of that name referring to different people. This is often the case in genealogies, search result pages for books, etc. Ideally, a WePS system based on clustering should put these documents in as many clusters as people mentioned with the name.

4.4.1 Extending Standard Metrics for Overlapping Clustering

While each item in standard clustering is assigned to one cluster, in overlapping clustering, each item is assigned to a set of clusters. Let us call "categories" to the set of "perfect" clusters defined in the *gold standard*. Then, any evaluation metric must reflect the fact that, in a perfect clustering, two items sharing n categories should share n clusters.

This apparently trivial condition is not always met. In particular, Purity and entropy-based metrics cannot capture this aspect of the quality of a given clustering solution. This occurs because they focus on the quality of the clusters (Purity) and the quality of the categories (Inverse Purity) independently from each other. Let us consider an example.

Multiplicity constraint

Figure 4.9 represents an overlapping clustering case. The rightmost distribution shows the correct solution: each item (1, 2 and 3) belongs to two categories and therefore appears in two clusters. The leftmost distribution, on the contrary, simply groups all items in just one cluster. This one does not represent the correct clustering. However, the only given cluster is perfectly coherent, since all items share one category (grey). In addition, all the items from the same category share the same cluster (because there is only one). Therefore, cluster/category oriented metrics inevitably think that the leftmost cluster is perfect.

This can serve to propose an additional constraint on evaluation metrics that handles the overlapping clustering problem (which is also illustrated in Figure 4.9).

Multiplicity constraint: The quality of a clustering solution in which the number of clusters is different from the number of categories in the *gold standard*, is always worse than the quality of the *gold standard*.

Multiplicity constraint vs. Dom's constraint

This restriction is related to one of Dom's empirical constraints which says that if the number of clusters diverges from the number of categories the quality must decrease. In the case of non-overlapping clustering it is covered by our constraints 1 and 2 (at least in boundary situations), but in the case of overlapping clusters it has to be explicitly added as an additional constraint.

The problem with Purity and Inverse Purity shows that the extension of quality metrics to overlapping clustering is not trivial. Instead of analysing all metric families in detail, here we will focus on extending *BCubed* metrics, which are the only ones that satisfy all formal constraints proposed in this chapter. In principle, however, metrics which are not cluster/category oriented (such as metrics based on counting pairs or metrics based on step editing distance) should also be extendable to handle the overlapping clustering problem.

4.4.2 Extending BCubed Metrics

BCubed metrics compute the precision and recall associated to each item in the distribution independently. The precision of one item represents the amount of items in the same cluster belonging to its category. Analogously, the recall of one item represents how many items from its category appear in its cluster.

Correctness

As we stated in Section 4.2.5, the correctness of the relation between two items in a non-overlapping clustering is represented by a binary function.

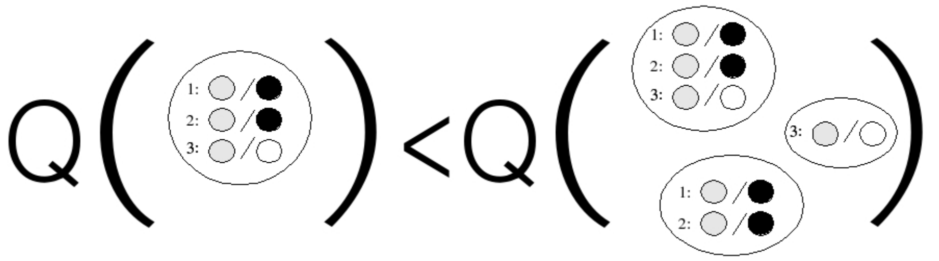


Figure 4.9: Clustering evaluation metric constraint 5: Item Multiplicity

$$\text{Correctness}(e, e') = \begin{cases} 1 \text{ if } L(e) = L(e') \longleftrightarrow C(e) = C(e') \\ 0 \text{ in other case} \end{cases}$$

where $L(e)$ is the cluster assigned to e by the clustering algorithm and $C(e)$ is the cluster assigned to e by the gold standard.

In the case of overlapping clustering the relation between two items can not be represented as a binary function. This is due to the fact that, in overlapping clustering, we must take into account the multiplicity of item occurrences in clusters and categories. For instance, if two items share two categories and share just one cluster, then the clustering does not capture the relation between both items completely (see items 1 and 2 in the second case of Figure 4.10). On the other hand, if two items share three clusters but just two categories, then the clustering introduces more information than necessary. This is the third case in Figure 4.10.

Multiplicity precision and recall

These new aspects can be measured in terms of *precision* and *recall* between two items. Let us define:

$$\text{Multiplicity Precision}(e, e') = \frac{\text{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|C(e) \cap C(e')|}$$

$$\text{Multiplicity Recall}(e, e') = \frac{\text{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|L(e) \cap L(e')|}$$

where e and e' are two items, $L(e)$ is the set of categories and $C(e)$ represents the set of clusters associated to e. Note that Multiplicity Precision is defined only when e, e' share some cluster, and Multiplicity Recall when e, e' share some category. This is enough to define BCubed extensions. Multiplicity Precision is used when two items share one or more clusters, and it is maximal (1) when the number of shared categories is lower or equal than the number of shared clusters, and it is minimal (0) when the two items do not share any category. Conversely, Multiplicity Recall is used when two items share one or more categories: it is maximal when the number of shared clusters is lower or equal than the number of shared categories, and it is minimal when the two items do not share any cluster.

Intuitively, multiplicity precision grows if there is a matching category for each cluster where both items co-occur. Multiplicity recall, on the other hand, grows when we add a shared cluster for each category shared by the two items. If we have less shared clusters than needed, we lose recall; if we have less categories than clusters, we lose precision. Figure 4.10 shows and example on how they are computed.

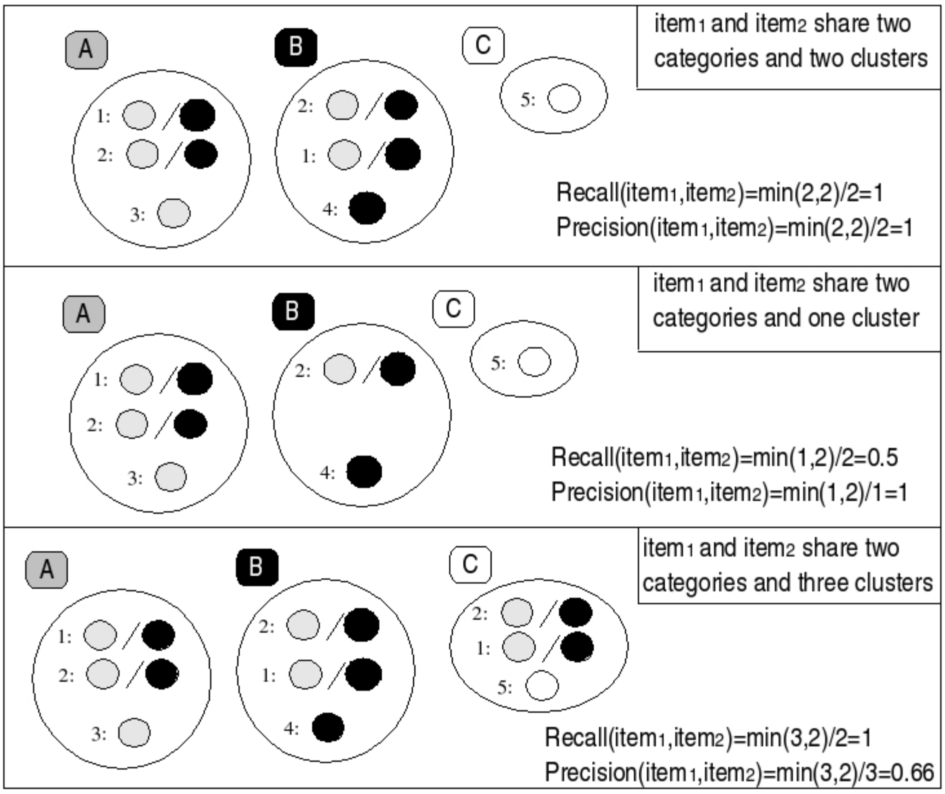

A
B
C
1:
2:
3:
2:
1:
4:
5:
item1 and item2 share two categories and two clusters
Recall(item1,item2)=min(2,2)/2=1
Precision(item1,item2)=min(2,2)/2=1
A
B
C
1:
2:
3:
2:
4:
5:
item1 and item2 share two categories and one cluster
Recall(item1,item2)=min(1,2)/2=0.5
Precision(item1,item2)=min(1,2)/1=1
A
B
C
1:
2:
3:
2:
1:
4:
2:
1:
5:
item1 and item2 share two categories and three clusters
Recall(item1,item2)=min(3,2)/2=1
Precision(item1,item2)=min(3,2)/3=0.66

Figure 4.10: Computing the multiplicity recall and precision between two items for extended BCubed metrics

Overall BCubed metric

The next step is to integrate multiplicity precision and recall into the overall BCubed metric. In order to do so, we will use the original BCubed definitions, but will replace the *Correctness* function with multiplicity precision (for BCubed precision) and multiplicity Recall (for BCubed recall). Then, the extended BCubed precision associated to one item will be its averaged multiplicity precision over other items sharing some of its categories; whereas the overall **extended BCubed precision** will be the averaged precision of all items. The **extended BCubed recall** is obtained using the same procedure. Formally:

$$\text{Precision BCubed} = \text{Avg}_e[\text{Avg}_{e'.C(e)\cap C(e')\neq\emptyset}[\text{Multiplicity precision}(e,e')]]$$

$$\text{Recall BCubed} = \text{Avg}_e[\text{Avg}_{e'.L(e)\cap L(e')\neq\emptyset}[\text{Multiplicity recall}(e,e')]]$$

Note that when clusters do not overlap, this extended version of BCubed metrics behaves exactly as the original BCubed metrics do, satisfying all previous constraints. Figure 4.17 shows that the F combination of the extended BCubed Precision and Recall satisfies all proposed formal constraints.

4.4.3 Extended BCubed: Example of Usage

In this section, we will illustrate how BCubed extended metrics behave using an example (see Figure 4.11). We start from a correct clustering where seven items are distributed along three clusters. Items 1 and 2 belong at the same time to two categories (black and grey). Since both the categories and the clusters are coherent, this distribution has maximum precision and recall.

Duplicate cluster

Now, let us suppose that we duplicate one cluster (black circle in Figure 4.12). In this case, the clustering produces more information than the categories require. Therefore, the recall is still maximum, but at the cost of precision. In addition, the more the clusters are duplicated, the more the precision decreases (see Figure 4.13). On the other hand, if items belonging to two categories are not duplicated, the clustering provides less information than it should, and BCubed recall decreases (Figure 4.14).

Split/merge cluster

If a correct cluster is split, some connections between items are not covered by the clustering distribution and the BCubed recall decreases (Figure 4.15). Contrariwise, if two clusters of the ideal distribution are merged, then some of the new connections will be incorrect, and the multiplicity of some elements will not be covered. Then, both BCubed precision and recall decreases (Figure 4.16).

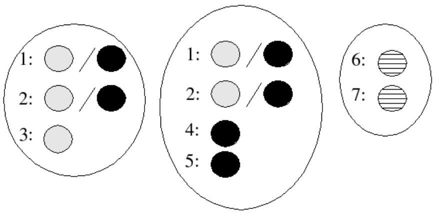


Figure 4.11: BCubed computing example 1 (ideal solution): Precision=1 Recall=1

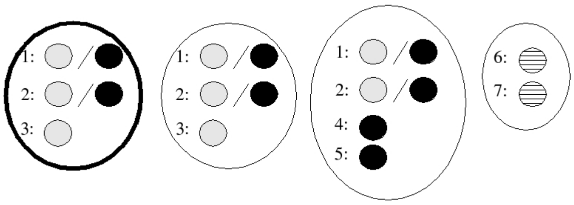


Figure 4.12: BCubed computing example 2 (duplicating clusters): Precision=0.86 Recall=1

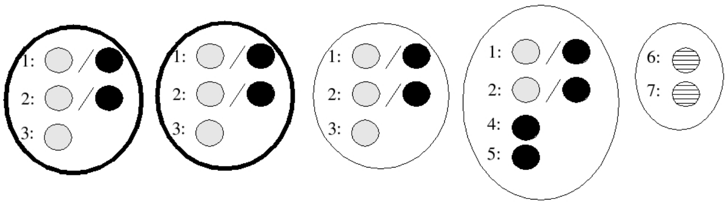


Figure 4.13: BCubed computing example 3 (duplicating clusters): Precision=0.81 Recall=1

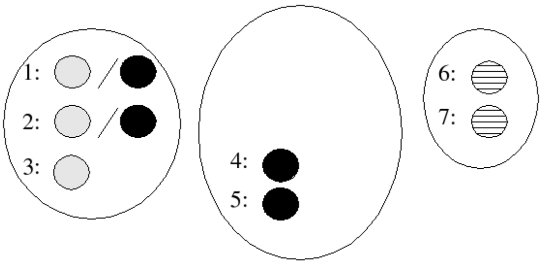


Figure 4.14: BCubed computing example 4 (removing item occurrences): Precision=1 Recall=0.68

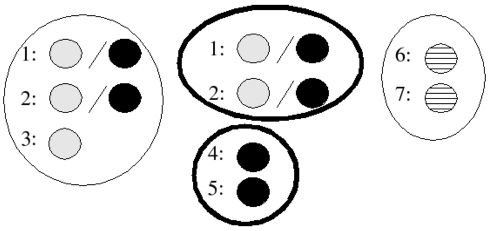


Figure 4.15: BCubed computing example 5 (splitting clusters): Precision=1 Recall=0.74

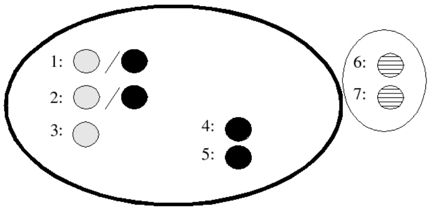


Figure 4.16: BCubed computing example 6 (joining clusters): Precision=0.88 Recall=0.94

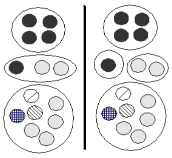
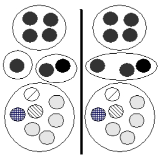
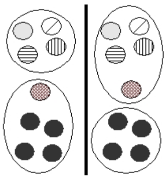
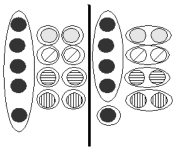
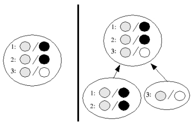
Distributions	BCubed Precision	BCubed Recall	F(Precision,Recall) $\alpha = \frac{1}{2}$
Cluster Homogeneity			
	0.59 0.69 √	0.69 0.69 ×	0.64 0.69 √
Cluster Completeness			
	0.62 0.62 ×	0.71 0.75 √	0.66 0.68 √
Rag Bag			
	0.52 0.64 √	1 1 ×	0.68 0.78 √
Cluster size vs. quantity			
	1 1 ×	0.64 0.81 √	0.78 0.89 √
Overlapping Clustering			
	1 1 ×	0.5 1 √	0.66 1 √

Figure 4.17: F average over BCubed Precision and Recall satisfies all formal constraints: $F(P,R) = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$

4.4.4 Extended BCubed Applied to WePS-1

In this section, we will compare the behaviour of standard metrics Purity and Inverse Purity with the suggested metrics BCubed Precision and Recall, in the context of WePS-1. We will see that the standard metrics used as official results in that campaign cannot discriminate a *cheat* clustering solution from a set of real systems, but the proposed metrics do.

Cheat system

One way of checking the suitability of evaluation metrics consists of introducing undesirable outputs (cheat system) in the evaluation testbed. Our goal then, is to check which set of metrics is required to discriminate these outputs against real systems. Here, we will use the cheat system proposed by Paul Kalmar in the context of the evaluation campaign[3], which consists of putting all items into one large cluster, and then duplicating each item in a new, size-one cluster (see Figure 4.18). Kalmar's cheat system exploits the overlapping clustering characteristic of the WePS task.

Let us suppose that we are clustering a set of documents retrieved by the query "John Smith". In this case, the cheat distribution would imply that every document mentions the same person and, in addition, that every document also talks about another "John Smith" which is only mentioned in that particular document. This solution is very unlikely and, therefore, this cheat system should be ranked at the bottom of the list when compared with real systems. Purity and Inverse Purity, however, are not able to discriminate this cheat distribution.

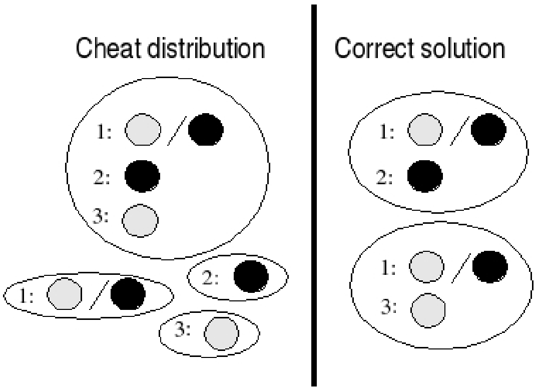


Figure 4.18: Clustering output of a cheat system

System rankings

Table 4.1 shows the system rankings according to Purity, Inverse Purity and the F combination of both ($\alpha = 0.5$). The cheat system obtains a maximum Inverse Purity, because all items are connected to each other in the big cluster. On the other hand, all duplicated items in single clusters contribute to the Purity of the global distribution. As a result, the cheat system ranks fifth according to Purity. Finally, it appears in the second position when both metrics are combined with the F measure.

Cheat system behaviour

The key is overlapping clusters: the WePS clustering task allows systems to

[3] Discussion forum of Web People Search Task 2007 (Mar 23th 2007) http://groups.google.com/group/web-people-search-task—semeval-2007/

Purity		Inverse Purity		F(Purity,Inverse Purity)	
S4	0.81	**Cheat System**	1	S1	0.79
S3	0.75	S14	0.95	**Cheat System**	0.78
S2	0.73	S13	0.93	S3	0.77
S1	0.72	S15	0.91	S2	0.77
Cheat System	0.64	S5	0.9	S4	0.69
S6	0.6	S10	0.89	S5	0.67
S9	0.58	S7	0.88	S6	0.66
S8	0.55	S1	0.88	S7	0.64
S5	0.53	S12	0.83	S8	0.62
S7	0.5	S11	0.82	S9	0.61
S10	0.45	S2	0.82	S10	0.6
S11	0.45	S3	0.8	S11	0.58
S12	0.39	S6	0.73	S12	0.53
S13	0.36	S8	0.71	S13	0.52
S14	0.35	S9	0.64	S14	0.51
S15	0.3	S4	0.6	S15	0.45

Table 4.1: WePS-1 system ranking according to Purity, Inverse Purity and F(Purity, Inverse Purity)

BCubed Precision		BCubed Recall		F(Precision,Recall	
(BP)		(BR)			
S4	0.79	**Cheat System**	0.99	S1	0.71
S3	0.68	S14	0.91	S3	0.68
S2	0.68	S13	0.87	S2	0.67
S1	0.67	S15	0.86	S4	0.58
S6	0.59	S5	0.84	S6	0.57
S9	0.53	S10	0.82	S5	0.53
S8	0.5	S1	0.81	S7	0.51
S5	0.43	S7	0.81	S8	0.5
S7	0.42	S12	0.74	S9	0.48
S11	0.36	S11	0.73	S11	0.42
S10	0.29	S2	0.73	S12	0.38
S12	0.29	S3	0.71	S13	0.38
S13	0.28	S6	0.64	S10	0.38
S14	0.26	S8	0.63	S14	0.36
S15	0.23	S9	0.53	S15	0.3
Cheat System	0.17	S4	0.5	**Cheat System**	0.24

Table 4.2: WePS-1 system ranking according to Extended BCubed Precision, Extended BCubed Recall, and its F combination.

put a document in several clusters if it refers to multiple people. The cheat system exploits this feature, producing a cluster of size one for each document, plus a cluster containing all documents. For example, in Figure 4.18, three elements are shown in the cheat and the correct cluster distributions[4]. Inverse Purity is, by definition, perfect in the cheat system output, since all documents can be found in one cluster. This would usually correspond to a low Purity for this noisy cluster. But at this point, by duplicating each document and generating a new singleton cluster, we add many clusters with maximal Purity, and hence the average Purity is substantially increased. In summary, the cheat system produces a clustering which is useless, but gets a high score.

Let us see the results when using BCubed metrics. BCubed Recall behaves similarly to Inverse Purity, ranking the cheat system in first position. BCubed Precision, however, does not behave as Purity. In this case, the cheat system goes down to the end of the ranking. The reason here is that BCubed computes the precision of items rather than the precision of clusters. In the cheat system output, all items are duplicated and inserted into a single cluster, increasing the number of clusters. Therefore, the clustering solution provides more information than required, and the overall BCubed precision of the distribution is dramatically reduced (see Section 4.4.2). On the other hand, the BCubed recall slightly decreases (0.99) because the multiplicity of a few items belonging to more than two categories is not covered by the cheat system.

We just compare our proposed BCubed extension to Purity/Inverse Purity because, besides being the official measures used at WePS-1, they can be directly applied to overlapped clusters without modification. This is not possible with other metric families, such as metrics based on counting pairs (what is now a right/wrong pair?) entropy-based metrics or step editing metrics (which need new steps for duplicating items and removing duplicates). All that metrics need to be redefined before being applied to overlapping clustering problems.

4.5 Conclusions

Constraints

In this chapter, we have analysed extrinsic clustering evaluation metrics from a formal perspective, proposing a set of formal constraints that a good evaluation metric should satisfy in a generic clustering problem. Four constraints have been proposed. These correspond with basic intuitions about the quality features of a clustering solution. We have also compared our constraints with other related works, in order to check that they cover the basic features proposed in previous related research.

BCubed precision and recall

A practical conclusion of our work is that the combination of BCubed precision and recall metrics is the only one that is able to satisfy all constraints (for non-overlapping clustering). We take this result as a recommendation to use BCubed

[4] The clustering is represented as follows: (i) elements in the clustering are identified by the numbers, (ii) the coloured circles indicate to which class each element belongs in the *gold standard* (iii) more than one coloured circle next to an element means that the element belongs to multiple classes in the *gold standard*

metrics for generic clustering problems. Tt must be noted, however, that in general terms there is a wide range of clustering applications. For certain specific applications, some of the constraints may not apply, and new constraints may appear, which could make other metrics more suitable in those cases. Some recommendations apply:

- If the system quality is determined by the most representative cluster for each category, metrics based on matching between clusters and categories can be appropriate (e.g. Purity and Inverse Purity). However, we have to take into account that these metrics do not always detect small improvements in the clustering distribution, and that they might have negative implications in the system evaluation/refinement cycles.
- If the system quality is not determined by the most representative cluster for each category, other metric families based on entropy, editing distances, counting pairs, etc. would be more appropriate.
- If the system developer wants to avoid the quadratic effect over cluster sizes (related to our fourth formal constraint), we recommend to avoid using metrics based on counting pairs. Instead of this, the developer may use entropy-based metrics, editing distances or BCubed metrics.
- In addition to that, if the developer does not want to penalise merging unrelated items in a "rag bag" ("other" or "miscellaneous" cluster), then the only recommendable choice is BCubed metrics.

Overlapping clustering

We have also examined the case of overlapping clustering in WePS, where an item can belong to more than one category at once. Most evaluation metrics are not prepared to deal with cluster overlaps and its definition must be extended to handle them. An exception is Purity and Inverse Purity, which can be applied directly, but which fail to satisfy a very simple constraint that we introduce specifically to handle overlapping clustering. We have then focused on BCubed metrics, proposing an intuitive extension of BCubed precision and recall that handles overlaps, and that becomes the original BCubed metrics in the absence of overlapping.

We have tested the metrics using the clustering testbed from the WePS-1 competitive evaluation task, in which Purity and Inverse Purity (combined *via* Van Rijsbergen's F) were used for the official system scores. A cheating solution, which receives an unreasonably high F score (rank 2 in the testbed), is detected by the extended BCubed metrics, which sends the cheating solution to the last position in the ranking.

Chapter 5

Combining Evaluation Metrics via the Unanimous Improvement Ratio

In the previous chapter we have studied different extrinsic clustering quality metrics. We also presented the problems of evaluating overlapping clustering and ultimately proposed an extension of the BCubed that tackles this problem. The second question that was raised at the end of the WePS-1 campaign was how the combination of metrics affects the ranking of systems in this task. Specifically, we observed that a baseline system (ONE-IN-ONE) is ranked above many participants, but this baseline can be ranked much lower just by modifying the metrics combination weight. In this chapter we study this problem and propose a complementary evaluation method.

There exists a wide set of extrinsic clustering quality metrics, but all of them are grounded on two dimensions: (i) to what extent items in the same cluster also belong to the same group in the *gold standard*; and (ii) to what extent items in different clusters also belong to different groups in the *gold standard*. We have reviewed a wide set of extrinsic metrics in the previous chapter: Entropy and class entropy [SKK00, Gho03], Purity and Inverse Purity [ZK01], BCubed Precision and Recall metrics [BB98b], metrics based on counting pairs [HBV01, Mei03], etc.[1] In order to evaluate systems, metrics are usually combined according to Van Rijsbergen's F function [Rij74]:

$$F_\alpha(A,B) = A * B/(\alpha * R + (1-\alpha) * P)$$

where the α parameter allows to assign a relative weight to each metric. After stating the α value, the system improvements according to F are checked by means of statistical significant tests over test cases.

Our research goals are:

1. To verify to what extent the clustering evaluation results can be biased by the assigned metric weighting (i.e. the value for α), even when detected

[1] See [AGAV08] for a detailed overview.

differences are statistically significant.

2. To introduce a measure to quantify improvements without dependencies from metric weighting (the *Unanimous Improvement Ratio*).

In Section 5.1, we discuss how metric weights can bias the results of an evaluation. In Section 5.2, we introduce our proposal, and, in Section 5.3, we test it with some empirical studies over the WePS-1 data. Finally, in Section 5.4, we present our conclusions.

5.1 The Effects of Metric Weighting in Clustering Tasks

F measure

Although there is an implicit consensus among researchers that the *F measure* [Rij74] is the best way of combining evaluation metric pairs, F requires assigning relative weights to the individual metrics involved. For some tasks this requirement is not a problem, given that (i) metrics are often correlated and (ii) the mathematical properties of F ensure a certain robustness across different parametrisations. Therefore, sometimes the metric weights have only a minor impact of the system ranking produced by F.

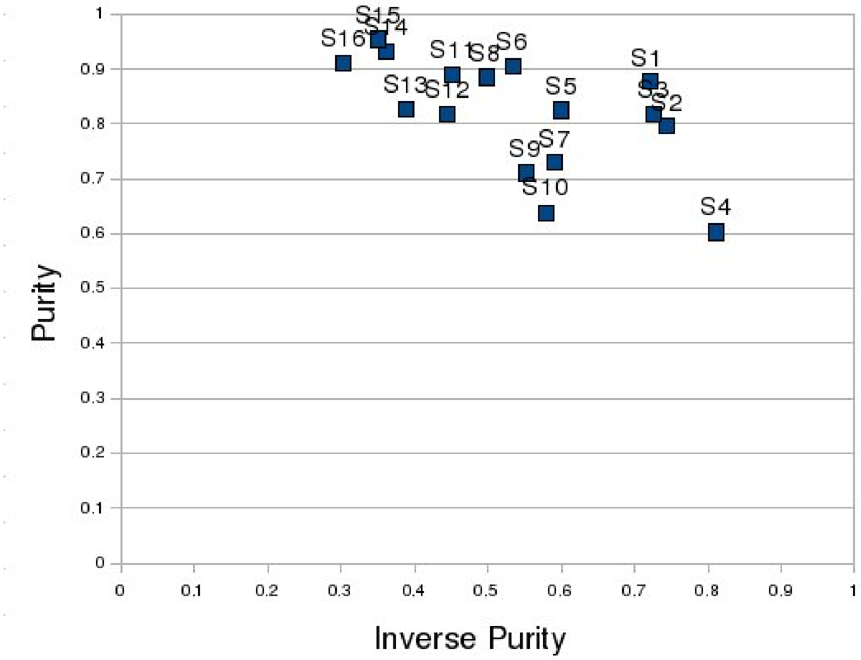


Figure 5.1: Evaluation results for clustering systems in WePS-1

Clustering behaviour

Clustering, however, is very sensitive to the parametrisation in metric combining functions. For instance, Figure 5.1 shows the Purity and Inverse Purity values obtained for each system the WePS-1 campaign. The figure shows that there is an important trade-off between Purity and Inverse Purity. As a result of this, depending

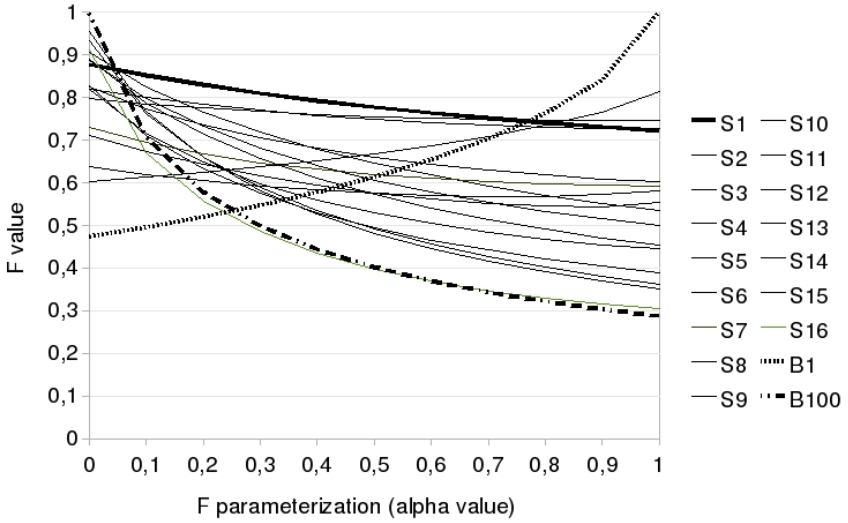


Figure 5.2: WePS-1 system evaluation results for several F parametrisations

on the metric weighting in the F combining function, the systems are ranked in a different way. Figure 5.2 shows the F values obtained by each system across different α values in F. This graph includes two baseline systems introduced in Chapter 3. The first is the ONE-IN-ONE baseline which consists of producing one cluster for each document (we will refer to it as B_1). The second baseline is ALL-IN-ONE and groups all documents in just one cluster (here named B_{100}).

α interpretation

Note that B_1 is better than most systems according to α values bigger than 0.5. We could conclude that most systems do not behave better than assigning one cluster to each document (baseline B_1). But the nature of the task suggests a different α value for the evaluation. Let us then consider two alternative clustering distributions. In the first one, all documents referring to a tennis player are included in the same cluster which contains also some documents that mention other people with the same name the tennis player has. In the other clustering distribution, there exists a cluster containing just documents about the tennis player, but not all of them. The first distribution will obtain the maximum Inverse Purity, while the second distribution will obtain the maximum Purity. According to $F_{0.5}$ both distributions are equivalent in terms of quality. However, from our point of view, the first distribution is better, given that it is easier to discard some documents about other people in cluster than exploring all clusters looking for the other documents that mention the tennis player. In conclusion, the parameter α should be fixed, for instance, at 0.2, thus giving more weight to Inverse Purity than to Purity.

As we saw in Chapter 3, according to $F_{0.2}$, a different baselines ranking is obtained (see Table 5.1). In this case, the baseline B_1 does not improve any system. That is, according to $F_{0.2}$, most systems represent a contribution with respect to the baseline approaches. Hence we must conclude that the task interpretation is crucial and it can affect the results substantially. In this case $F_{0.2}$ seems to be more reasonable for this particular task. And yet, why using $\alpha = 0.2$ and not, for instance, 0.3?

$F_{0.5}$		$F_{0.2}$	
Ranked systems	F result	Ranked systems	F result
S_1	0.78	S_1	0.83
S_2	0.75	S_3	0.78
S_3	0.75	S_2	0.77
S_4	0.67	S_6	0.76
S_5	0.66	S_5	0.73
S_6	0.65	S_8	0.73
S_7	0.62	S_{11}	0.71
$\mathbf{B}_1$	0.61	S_7	0.67
S_8	0.61	S_{14}	0.66
S_9	0.58	S_{15}	0.66
S_{10}	0.58	S_{12}	0.65
S_{11}	0.57	S_9	0.64
S_{12}	0.53	S_{13}	0.63
S_{13}	0.49	S_4	0.62
S_{14}	0.49	S_{10}	0.6
S_{15}	0.48	$\mathbf{B}_{100}$	0.58
$\mathbf{B}_{100}$	0.4	S_{16}	0.56
S_{16}	0.4	$\mathbf{B}_1$	0.49

Table 5.1: WePS-1 rankings for $F_{0.5}$ and $F_{0.2}$ using Purity and Inverse Purity

	B_1	S_{14}	Statistical significance
$F_{0.5}$	0.61	0.49	0.022
$F_{0.2}$	0.52	0.66	0.015

Table 5.2: Statistical significance of improvements: $F_{0.5}$ vs. $F0.2$

Statistical significance

A standard statistical significance test (such as the T-test or Wilcoxon) does not address this issue, because it is only applied to the outcome variable F and does not consider Purity and Inverse Purity values. For instance, B_1 improves S_{14} with statistical significance (see Table 5.2) according to the Wilcoxon test on $F_{0.5}$ ($\alpha < 0.05$) but it is improved by S_{14} when using $F_{0.2}$. In addition, we have identified 105 system pairs where one system improves the other with statistical significance according to the Wilcoxon test. In in 89 cases (%84) out of this set there exists a statistically significant quality decrease according to one of the metrics.

α parametrisation

We might think that it is enough to use the same α parameterisation that is used in the competition for which our system is designed. However, the meaning of the α value can change across competitions depending on the data distribution. For instance, according to $F_{0.5}$, the ONE-IN-ONE baseline approach improved the ALL-IN-ONE baseline for WePS-1. However, the situation reverses in WePS-2: the

ALL-IN-ONE baseline seems substantially better.

In summary, we need a metric combining function which does not depend on any arbitrary weighting criterion. This measure should ensure that a system improvement is robust across metric combining criteria and it should also reflect the range of the improvement in order to select the best one.

5.2 Proposal

5.2.1 Unanimous Improvements

Properties of a combining function

The problem of combining evaluation metrics is closely related with the theory of conjoint measurement. In [AGAV08] the role of conjoint measurement theory in our problem is described in detail. Rijsbergen [Rij74] argued that it is impossible to determine empirically which metric combining function (over Precision and Recall) is the most adequate in the context of Information Retrieval evaluation. However, starting from the measurement theory principles, Rijsbergen described the set of properties that a metric combining function should satisfy. This set includes the *Independence* axiom (also called *Single Cancellation*), from which the *Monotonicity* property derives. The Monotonicity axiom implies that the quality of a system that surpasses or equals another one according to all partial metrics is necessarily equal or better than the second. In other words, it represents an improvement with no dependence on how the metrics were weighted.

Unanimous Improvement

We will refer to this quality relation as an *Unanimous Improvement*. Formally, being $Q_X(a)$ the quality of a according to a set of metrics in X:

$$Q_X(a) \geq_\forall Q_X(b) \text{ if and only if } x(a) \geq x(b) \forall x \in X$$

Equality

This relation has no dependence on how metrics are scaled, weighted or on their degree of correlation in the metric set. In other words, it implies an "empirical" improvement, but without information about the improvement range. From its definition and the antisymmetry property, the equality ($=_\forall$) and the strict relationship $>_\forall$ are derived. The unanimous equality implies that both systems obtain the same score for all metrics:

$$Q_X(a) =_\forall Q_X(b) \equiv (Q_X(a) \geq_\forall Q_X(b)) \wedge (Q_X(b) \geq_\forall Q_X(a))$$

Strict unanimous improvement

The strict unanimous improvement implies that one system improves the other at least for one of the metrics and that it is not improved by any of the other metrics.

$$Q_X(a) >_\forall Q_X(b) \equiv (Q_X(a) \geq_x Q_X(b)) \wedge \neg(Q_X(a) =_x Q_X(b)) \equiv$$

$$(Q_X(a) \geq_\forall Q_X(b)) \wedge \neg(Q_X(a) \geq_\forall Q_X(b))$$

Metric biased improvements

The non comparability $||$ is also derived. It means that some metrics reward one system and some metrics reward the other. We refer to this cases as *metric biased improvements*.

$$Q_X(a) \|_\forall Q_X(b) \equiv \neg(Q_X(a) \geq_\forall Q_X(b)) \wedge \neg(Q_X(b) \geq_\forall Q_X(a))$$

The need for UIR

The theoretical properties of the Unanimous Improvement are described in depth in [AGAV08]. The most important is that the Unanimous Improvement is the only relational structure that, while satisfying the Independence (Monotonicity) axiom, does not depend on metric weightings. In other words, we can claim that: *A system improvement according to a metric combining function does not depend in any way on metric weightings only if there is no quality decrease according to any individual metric.*

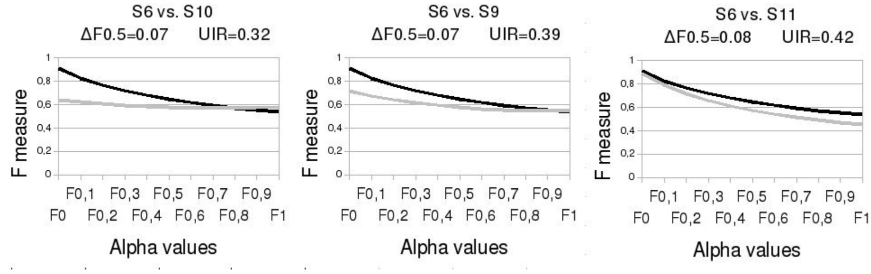


Figure 5.3: F measure vs. UIR: rewarding robustness

5.2.2 Unanimous Improvement Ratio

Given that the Unanimous Improvement is the only metric combining function that does not depend on metric weighting, our unique observable over each test case is a three-valued function (unanimous improvement, equality or biased improvement). However, we need a quantitative function in order to validate system improvements.

Formal constraints

Having two systems, a and b, and the Unanimous Improvement relationship over test cases, we have samples for which a improves b ($Q_X(a) \geq_\forall Q_X(b)$), b improves a ($Q_X(b) \geq_\forall Q_X(a)$) and also biased improvements ($Q_X(a)\|_\forall Q_X(b)$). We will refer to these sets as $T_{a\geq_\forall b}$, $T_{b\geq_\forall a}$ and $T_{a\|_\forall b}$ respectively. We will refer to the total amount of samples as T. We will define the quantitative measure Unanimous Improvement Ratio (UIR) according to three formal restrictions:

1. An increment of $T_{a\|_\forall b}$ samples implies a decrement in MIR. In the extreme case, if all samples are metric weighting biased ($T_{a\|_\forall b} = T$), then UIR=0.

2. If $T_{a\geq_\forall b} = T_{b\geq_\forall a}$ then UIR$= 0$.

3. Given a fixed $T_{a\|_\forall b}$, UIR is proportional to $T_{a\geq_\forall b}$ and inversely proportional to $T_{b\geq_\forall a}$.

UIR

Given these restrictions, we propose the following UIR definition:

$$\mathrm{UIR}_{X,T}(a,b) = \frac{|T_{a\geq_\forall b}| - |T_{b\geq_\forall a}|}{|T|} =$$

$$\frac{|t \in T/Q_X(a) \geq_\forall Q_X(b)| - |t \in T/Q_X(b) \geq_\forall Q_X(a)|}{|T|}$$

UIR limitations

UIR has two main limitations. First, it is not *transitive*, as it is the case with the Unanimous Improvement [AGAV08]. Therefore, it is not possible to define a linear system ranking based on UIR. In addition, there is some information loss when comparing systems given that the ranges in evaluation results are not considered.

On the other hand, the main advantage of UIR is that no metric weighting is necessary. In addition, given that the Unanimous Improvement does not consider metric ranges, the scale properties or normalisation issues of individual metrics do not affect the results.

5.3 Experiments

This section provides experiments in the WePS-1 data in order to confirm that:

1. UIR rewards improvements that are robust across metric weighting schemes.
2. Given a set of equally robust improvements, the measure ideally rewards the system that produces the largest improvement.
3. There exists a threshold for UIR values, such as that obtaining a UIR above the threshold guarantees that an improvement is robust, and this threshold is not too strong, so that we can still identify differences between systems.

5.3.1 Rewarding Robustness Across α Values

F_α curves

Figure 5.3 shows three examples of system comparisons. Each curve represents the F_α value obtained for the system for different α values. System S6 (black curves) is compared with S10, S9 and S11 (grey curves) in each of the graphs. In all cases there is a similar quality increase according to $F_{0.5}$. However, UIR points out some differences: Depending on to what extent the improvement is robust across α values in F, UIR assigns different values to the improvement. S6 vs. S11 (rightmost graph) gives the largest UIR, because those systems do not swap their F values for any α. S6 vs. S10, on the other hand, has the smallest UIR value because the performances of S6 and S10 swap around $\alpha = 0.8$.

Another way of testing whether UIR rewards robustness is to consider two kinds of system comparisons separately: (i) system pairs for which F_α increases for all α values, and (ii) system pairs for which F increases for some α values and decreases for other α values. Table 5.3 shows the average increments for UIR and $F_{0.5}$ in each case. Note that UIR rewards the absence of contradiction between α values substantially (0.53 vs. 0.14). Notably, the absolute increase of $F_{0.5}$ is similar for both cases. In other words, although $F_{0.5}$ assigns the same relevance to Purity

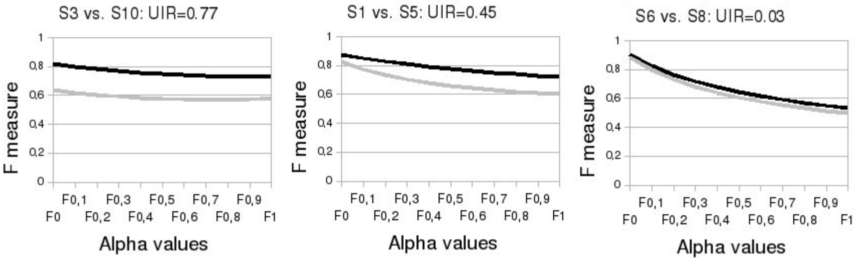


Figure 5.4: F vs. UIR: reflecting improvement ranges

	Improvements for all α 28 system pairs	Other cases 125 system pairs
$\mid \triangle F_{0.5}\mid$	0.12	0.13
|UIR|	0.53	0.14

Table 5.3: UIR and $F_{0.5}$ increase when F increases for all α values

and Inverse Purity, a certain $F_{0.5}$ improvement range does not say anything about whether we can improve both Purity and Inverse Purity at the same time.

Statistical significance

We can also confirm this conclusion by considering both metrics (Purity and Inverse Purity) independently. According to the statistical significance of the improvements for independent metrics, we can distinguish three cases:

1. *Contradictory improvements*: One metric increases and the other decreases, with statistical significance in both cases.

2. *Robust improvements*: Both metrics improve significantly, or at least one improves significantly and the other does not decrease significantly.

3. *No improvement*: There is no statistically significant differences for any metric.

For this purpose we will use the Wilcoxon test with $p < 0,05$. Surprisingly, Table 5.4 shows that the $F_{0.5}$ increase is even bigger whenever improvements are contradictory than whenever they are robust. Apparently, $F_{0.5}$ rewards individual

	Robust improvements 53 pairs	Contradictory improvements 89 pairs	No imp. 11 pairs
$\mid \triangle F_{0.5}\mid$	0.11	0.15	0.05
|UIR|	0.42	0.08	0.027

Table 5.4: UIR and $F_{0.5}$ increases vs. statistical significance tests

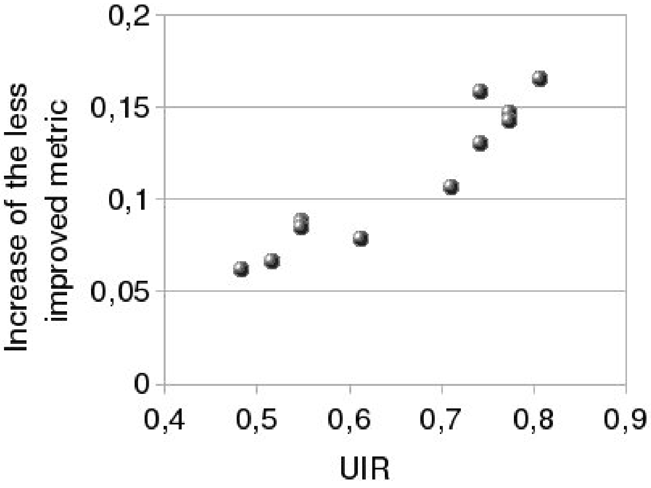


Figure 5.5: UIR vs. the improvement according to the less improved metric.

metric improvements obtained at the cost of (smaller) decreases in the other metric. UIR has a sharply different behaviour, rewarding robust improvements.

5.3.2 Reflecting Improvement Ranges

We have verified empirically that UIR measures to what extent an improvement is robust across alternative α values. However, given a set of equally robust improvements, the measure should also reward the system that produces the largest improvement.

α curves

Let us consider an example taken from the WePS-1 testbed. Figure 5.4 represents the $F_{\alpha\in[0,1]}$ values for three system pairs. In all cases, one system improves the other for all α values, yet depending on the improvement range, UIR assigns higher values to larger improvements.

Correlation with individual metrics

As a matter of fact, whenever both metrics are improved, the metric that has the weakest improvement determines the behaviour of UIR: Figure 5.5 illustrates this relationship for the ten system pairs with a largest improvement for both criteria; the Pearson correlation in this graph is 0.94.

5.3.3 UIR Threshold

What UIR value is appropriate to state that a system improvement is robust enough? We could set a very restrictive threshold and say, for instance, that an improvement is significantly robust when UIR≥ 0.75. But such restriction would hardly be satisfied, and then, the UIR test would not be informative: many robust system improvements would remain undetected by this test.

So, our question now is whether there exists a threshold for UIR values such as that obtaining a UIR above the threshold guarantees that an improvement is robust, and, at the same time, the threshold is not too strong, so that we can still identify

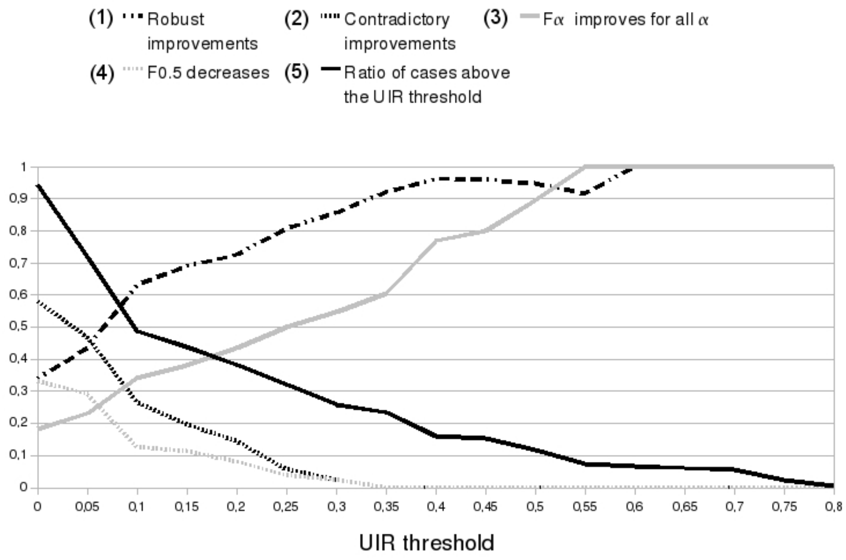


Figure 5.6: Improvement detected across UIR thresholds

actual differences between systems.

Figure 5.6 shows the ratio of system pairs (a, b) (black curve) such as that UIR(a, b) is bigger than a given threshold (horizontal axis). We have added a few more curves that represent key features of the system pairs:

1. The proportion of robust system improvements, i.e. cases where both metrics improve significantly, or, at least, cases in which one improves significantly and the other does not decrease significantly

2. The proportion of contradictory system improvements (see definition above).

3. The ratio of system pairs for which $F_{0.5}$ increases for all α values ($F_\alpha(a) > F_\alpha(b) \forall \alpha$).

4. The ratio of system pairs for which $F_{0.5}$ decreases although UIR is positive ($F_{0.5}(a) < F_{0.5}(b)$).

As the figure shows, an UIR threshold of 0.25 accepts around 25% of all system pairs. From this set, the number of contradictory improvements and the number of cases where $F_{0.5}$ decreases are low (4% and 6% respectively). Also, in 50% of the cases F_α increases for all α values, and in 80% of the cases improvements are robust. It seems, therefore, that UIR≥ 0.25 constitutes a reasonable threshold.

5.4 Conclusions

The analysis we have described in this chapter shows that the comparison of systems in clustering tasks is highly sensitive to the way of combining evaluation metrics. The UIR measure presented in this chapter allows us to combine evaluation metrics without assigning a relative weight to each metrics, and the empirical analysis has showed that UIR rewards robust improvements with respect to different metric weights.

UIR can be exploited in two ways. First, according to the UIR≥ 0.25 threshold that was inferred from our empirical study, UIR is able to test the robustness of system improvements in shared tasks (such as the WePS clustering task). Second, given that UIR provides quantitative values, it is an alternative way of selecting the best approach during system training processes.

In the next chapter, we will test the UIR method on the results of the second WePS campaign.

Chapter 6

The WePS-2 Campaign

While designing the second WePS evaluation campaign we took into account both the experience acquired in WePS-1 and the studies on evaluation metrics described in the previous chapters. During the campaign we introduced the following improvements: the use of a robust clustering evaluation metric (extended BCubed); the application of a complementary results analysis (Unanimous Improvement Ratio); a more cost-effective method for the creation of the testbed; and a new attribute extraction task [SA09].

In WePS-2, 19 research teams from over the world took part in either or both of the following tasks: (i) clustering web pages to solve the ambiguity of search results, and (ii) extracting 18 kinds of attribute values for target individuals whose names appear on a set of web pages.

In this chapter, we present an overview of the WePS 2 clustering task, including: a description of the datasets (Section 6.1), the methodology to produce our *gold standard*, the evaluation methodology (Section 6.2) and the campaign design. We also provide an overview of the participating systems in Section 6.3. The results of the evaluation are presented and discussed in Section 6.4, and we finish with some concluding remarks in Section 6.5.

6.1 Testbed

The data distributed to participants was divided in development and test data.

6.1.1 Development Data

This data was handed to participants so that they could develop and test their systems before processing the evaluation test set. The development data consists of the corpora and clustering *gold standard* previously used for the WePS-1 campaign (Chapter 3), and it was built basically with the same methodology used in WePS-2. The WePS-1 data includes 47 ambiguous names and up to 100 manually clustered search results for each name. The number of clusters per name has a large variability (from 1 up to 91 different people sharing the same name) even for the 10 names extracted from Wikipedia biographies. Once again, we assumed that names with a

Wikipedia entry would be less ambiguous (because a celebrity tends to monopolise search results), but our Wikipedia names had an average of 23 clusters per name in the WePS-1 training data and twice this amount in the WePS-1 test data. Note that average ambiguity is not very informative for this task, because it does not seem to follow a binomial distribution: a name corresponding with just two people seems as likely to occur as a name with 30 or a name with 90.

Additionally, this data includes the Web03 corpus [Man06] which features a more diverse number of documents for each name and a lower average ambiguity.

6.1.2 Test Data

Name sources

The WePS-2 test data consists of 30 datasets (Table 6.1), each one corresponding to a certain ambiguous name. As in WePS-1, three different sources were used to obtain the names:

Wikipedia. Ten names were randomly sampled from the list of biographies in the English version of Wikipedia. In this occasion, unlike the WePS-1 dataset, our hypothesis of lower ambiguity for names in the Wikipedia has a correspondence with the data: as we can see from the results of the manual annotation (Table 6.1), six out of these ten datasets, contain less than ten different people, and three are dominated by only one person.

ACL'08. Another ten names where randomly extracted from the list of Programme Committee members for the annual meeting of the Association for Computational Linguistics (ACL'08). These cases present a different type of ambiguity scenario, in which we know in advance that at least someone of the people mentioned should be a Computer Science scholar.

US Census. Using the lists of first and last names in the 1990 US Census[1] data, we composed 10 random names. In order to avoid extremely rare or inexistent name combinations, we weighted the probability of choosing a name according to its frequency in the Census. The result is a set of fairly ambiguous names, with an average of 30 different people mentioned in each dataset.

Search results

We obtained the top 150 search results for each name from an Internet search engine[2] (using the name as a quoted query and searching only for pages written in English). All the information obtained from the search results (snippets, position in the ranking, document title, original URL) was stored and distributed to the participants as part of the datasets.

Documents filtering

Some web pages from the search results were not included in the final corpus. In some cases, pages could not be downloaded or were not available at the time the corpus was created. We also filtered out documents that did not contain at least one occurrence of the person name. Finally, in order to simplify the preprocessing task for participants, only HTML pages were included in the datasets.

[1] http://www.census.gov/genealogy/names/

[2] We used the Web Search Service API provided by Yahoo! (http://developer.yahoo.com/search/)

Name	entities	documents	discarded
Wikipedia names			
Bertram Brooker	1	128	30
David Tua	1	134	36
Franz Masereel	3	126	26
Herb Ritts	2	127	31
James Patterson	4	133	33
Jason Hart	22	130	38
Louis Lowe	24	100	25
Mike Robertson	39	123	35
Nicholas Maw	1	135	36
Tom Linton	10	135	41
Average	10.70	127.10	33.10
ACL'08 names			
Benjamin Snyder	28	95	40
Cheng Niu	7	100	7
David Weir	26	128	33
Emily Bender	19	120	31
Gideon Mann	2	95	6
Hao Zhang	24	100	13
Hoi Fang	21	90	28
Ivan Titov	5	101	28
Mirella Lapata	2	91	1
Tamer Elsayed	8	101	18
Average	14.20	102.10	20.50
Census names			
Amanda Lentz	20	121	46
Helen Thomas	3	127	27
Janelle Lee	34	93	37
Jonathan Shaw	26	126	46
Judith Schwartz	30	124	39
Otis Lee	26	118	40
Rita Fisher	24	109	13
Sharon Cummings	30	113	29
Susan Jones	56	110	30
Theodore Smith	54	111	43
Average	30.30	115.20	35.00
Global average	18.64	114.42	29.42

Table 6.1: WePS-2 test data

6.1.3 Manual Annotation

Each dataset was annotated independently by two different assessors. Each of these assessors was requested to manually group the search results so that each group contained all and only those documents referring to one of the individuals sharing the ambiguous name. Once the first 100 document were grouped, the annotation stopped. A web application (fig. 6.1) was developed to make easier the annotation process, allowing the annotator to browse and edit the clusters quickly as well as leaving comments regarding specific clusters or documents.

Overlapping clusters

Assigning the same document to more than one cluster was allowed whenever necessary; for instance, a web page with results from Amazon might be a list of books written by two different authors sharing the same name; in this case, the annotator was supposed to assign the page to two different clusters.

Discarded documents

In some cases, it was impossible to decide which individual was mentioned in a certain page, and, therefore, page was discarded from the dataset. These are the most frequent reasons to discard a page:

- Unclear pages: in general pages that do not offer enough information. This is often the case for Facebook, Linked-in and other public profiles from social networks. For instance, a Facebook public profile may just contain the name and a statement such as “I like movies and chocolate”. This information is compatible with virtually any cluster, and is therefore not useful to resolve ambiguity.

- Genealogy pages: These documents are simply too complex to treat: they are too long and contain large genealogy trees which are hard to compare with other web pages.

- Pages in languages other than English: these are documents which were incorrectly tagged by the Yahoo language identifier filter, and are mostly in Chinese (e.g. in *Hao Zhang*, *Feng Hui* datasets), Arabic (*Tamer Elsayed*), Norwegian or Finnish (*Ivan Titov*).

Inter-annotator agreement

Once both assessors annotated the full dataset, they met, discussed their annotations, and produced a single manual tagging of the data, which is used as WePS-2 *gold standard*. The cause for most disagreements was a different interpretation of which facts constitute enough evidence to merge a specific page with a given cluster. There are frequent borderline cases where many interpretations are possible, thus making it difficult to establish a general annotation policy.

Difficult cases

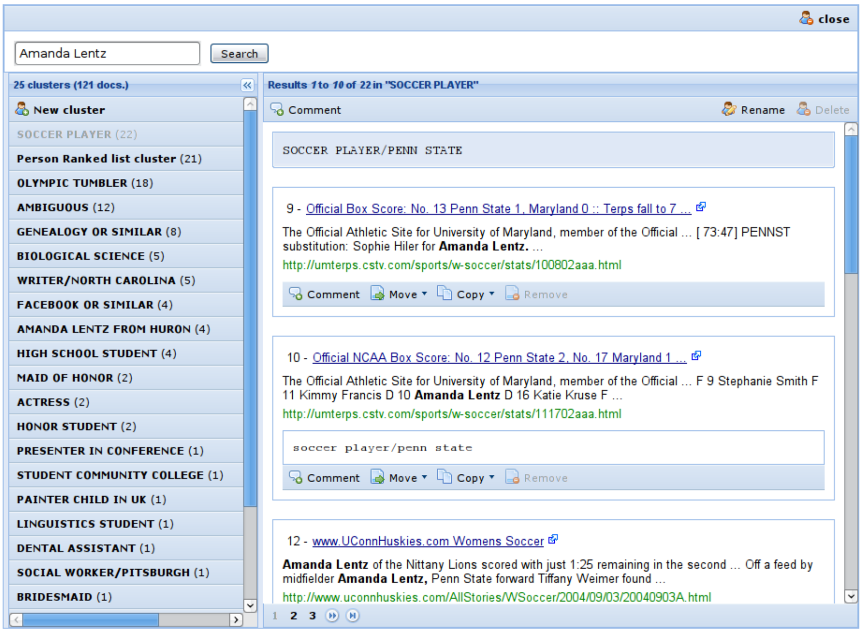
Occupations are usually a good hint for the task. For example, in the *Benjamin Snyder* dataset we found three documents that mention people living in Boston. One of them is an MIT student in engineering, while another one is a lawyer. It is therefore reasonable to assume that they are different people. The third document, however, describes a wine aficionado. But, as it happens, this is more of a hobby than an occupation, and it could be compatible with the previous ones. One of the annotators decided that there was not enough information to either create a new cluster for the document or to add it to one of the existing cluster, hence discarding the document. The second annotator, however, decided that it was appropriate to consider it a separate person since there is no information linking the wine aficionado with the other two profiles. After the discussion, the second option prevailed.

Other conflictive cases simply derive from the task complexity: sometimes, a page provides a large amount of text and the key information is not visible at a first glance. Datasets in which there is a high ambiguity, the annotator has to keep all the details of the people he finds, and compare them to the new evidence offered by each new page. Eventually this mechanism leads to human errors. The strategy of having two independent annotations and then a discussion helped us to detect this kind of errors.

WePS-1 vs. WePS-2

In comparison with WePS-1, the new dataset has a much lower ambiguity: in average, there are 18.64 different people per name, but the predominant person for a given name owns half of the documents. Again, note that averages are not particularly informative, because there are many extreme cases. For instance, there are 3 (out of 30) names referring to just one person, and 6 cases with 30 or more different people sharing a single name.

There is also a higher number of discarded documents compared to WePS-1. This is due to the more conservative tagging guidelines that prevented many documents with insufficient information from being grouped.



25 clusters (121 docs.)

SOCCER PLAYER (22)
Person Ranked list cluster (21)
OLYMPIC TUMBLER (18)
AMBIGUOUS (12)
GENEALOGY OR SIMILAR (8)
BIOLOGICAL SCIENCE (5)
WRITER/NORTH CAROLINA (5)
FACEBOOK OR SIMILAR (4)
AMANDA LENTZ FROM HURON (4)
HIGH SCHOOL STUDENT (4)
MAID OF HONOR (2)
ACTRESS (2)
HONOR STUDENT (2)
PRESENTER IN CONFERENCE (1)
STUDENT COMMUNITY COLLEGE (1)
PAINTER CHILD IN UK (1)
LINGUISTICS STUDENT (1)
DENTAL ASSISTANT (1)
SOCIAL WORKER/PITSBURGH (1)
BRIDESMAID (1)

Results *1* to *10* of 22 in "SOCCER PLAYER"

SOCCER PLAYER/PENN STATE

9 - Official Box Score: No. 13 Penn State 1, Maryland 0 :: Terps fall to 7 ...
The Official Athletic Site for University of Maryland, member of the Official ... [73:47] PENNST substitution: Sophie Hiler for **Amanda Lentz.** ...
http://umterps.cstv.com/sports/w-soccer/stats/100802aaa.html

10 - Official NCAA Box Score: No. 12 Penn State 2, No. 17 Maryland 1 ...
The Official Athletic Site for University of Maryland, member of the Official ... F 9 Stephanie Smith F 11 Kimmy Francis D 10 **Amanda Lentz** D 16 Katie Kruse F ...
http://umterps.cstv.com/sports/w-soccer/stats/111702aaa.html

soccer player/penn state

12 - www.UConnHuskies.com Womens Soccer
Amanda Lentz of the Nittany Lions scored with just 1:25 remaining in the second ... Off a feed by midfielder **Amanda Lentz,** Penn State forward Tiffany Weimer found ...
http://www.uconnhuskies.com/AllStories/WSoccer/2004/09/03/20040903A.html

Figure 6.1: WePS-2 clustering annotation GUI

6.2 Evaluation Methodology

Extended BCubed

In this campaign, we used the extended BCubed clustering metrics for evaluating the systems. As we thoroughly described in Chapter 4, B-Cubed [BB98b] metrics are the only ones that satisfy four intuitive formal constraints on evaluation metrics for the clustering problem. We then extended the original B-Cubed definition to handle overlapping clustering.

Metrics combination

As we did in WePS-1 (Section 3.3), the ranking of systems was obtained using F-measure $F_{\alpha=0.5}$ in order to combine the metrics (in this case BCubed precision and BCubed recall). We also included the results for $F_{\alpha=0.2}$. This additional measure rewards a better recall while still considering the precision aspect. This parametrisation of the F measure captures the intuition that filtering out a few noisy documents from the relevant cluster (i.e. having a problem of precision) is more acceptable to the user than having to inspect all other clusters in search for missing information (i.e. having a problem of recall in the relevant cluster). An additional results analysis was performed using the UIR (*Unanimous Improvement Ratio*) measure described in Chapter 5.

Baselines

As in WePS-1, two simple baseline approaches were applied to the test data: *ALL-IN-ONE* and *ONE-IN-ONE*. A third baseline consists of a simple clustering system: a HAC algorithm with single linkage. Similarity is calculated using cosine; documents are represented by a bag of words and weighted with tf*idf. We evaluated two variations of this baseline, one that uses a BoW of tokens in the document (*HAC-TOKENS*) and other that uses bigrams (*HAC-BIGRAMS*). English stopwords are removed in each case. A fixed similarity threshold is obtained from the training data and then applied to the test data.

Upper bound systems

The selection of an appropriate threshold (or other clustering stop criteria) is a challenging issue [PK06] in the WePS task. In order to provide an upper bound of the results that can be achieved with the previous baseline system and a perfect threshold selection, we have evaluated the results obtained whenever the best threshold is selected for each topic (i.e. each person name) – *BEST-HAC-TOKENS* and *BEST-HAC-BIGRAMS–*. This is not a baseline, but it gives us an insight on the relevance of the clustering threshold, as well as the degree of improvement simple baseline systems like *HAC-TOKENS* and *HAC-BIGRAMS* are capable of.

WePS-1 cheat system

Finally, we have also included the WePS-1 cheat system in the results, in order to verify that the new metrics detect and penalise this non-informative baseline.

6.2.1 Campaign Design

The schedule for the evaluation campaign was set as follows: (i) release of the task description, development data sets and scoring program; (ii) release of the test data sets; (iii) participants send their answers to the task organisers[3]; (iv) the task organisers evaluate the answers and send the results.

Task description

The task description was released before the start of the official evaluation. The training data and evaluation scripts were already available since they were produced and published during the WePS-1 campaign. The participants had three months to develop their systems.

Evaluation period

The official evaluation period started with the release of the test datasets. These datasets included the search results metadata and HTML documents for each dataset. Participants had one week to run their systems and submit up to five different sets of answers to the organisers.

Results submission

Once the evaluation period finished, the answers were evaluated by the organisers and the detailed results were submitted all of the teams. The ranking with all submitted runs was then made public, as well as the *gold standard* for the test data.

6.3 Participants

In this section we summarise some of the common traits found in the systems developed by the participants.

Preprocessing

All systems have some preprocessing stage where HTML documents are con-

[3] Participants were allowed to submit up to five runs.

verted into plain text (Java HTML Parser[4] and Beautiful Soup[5] were among the most popular tools used for this purpose). The next steps generally involve tokenisation, stop-word removal and, in some cases, sentence detection [CLH09, IOS^{+}09, RBGST09, HZ09]. Porter stemming was performed by some teams [BHH^{+}09, GMV^{+}09, LZL^{+}09, MRA09, Ven09], although it is not clear to which extent it affects the clustering results (some of the top systems use it and some do not).

Features

The most commonly used feature is the Bag of Words (BoW). Nevertheless, in some cases it was restricted to sentences where the ambiguous name occurs within a window of words [IOS^{+}09, RBGST09, KF09]. Named entity recognition (NER) is usually presented as a critical feature in order to obtain accurate clustering results. Indeed, it is the second most commonly used feature [IOS^{+}09, NTCKM09, KF09, HZ09, PTV^{+}09, GO09][6]. Surprisingly, three out of the top four systems in the ranking did not use NER [CLH09, BHH^{+}09, IOS^{+}09, RBGST09]. It seems NER is not necessary to build a competitive system, although it may still be a valuable source of information. In most cases, features were weighted using simple tf*idf functions. Other measures used were the gain ratio [LFHDC09], Kullback-Leibler divergence [MRA09] and self information [RBGST09].

Bigrams were used by [CLH09, RBGST09] and seemed to provide a good trade-off between precision and recall. Among the most sparse features we find systems that used hyperlinks [IOS^{+}09, GO09, CLH09, LFHDC09], email addresses, phone numbers, dates [LFHDC09], variations of the ambiguous name [GO09], etc.

Clustering algorithms

Hierarchical Agglomerative Clustering (HAC) is the most popular clustering algorithm, although the choice of linkage varies (e.g. single link in [CLH09, NTCKM09], group average in [IOS^{+}09]). In cases where a hierarchical algorithm like HAC was used, the number of clusters in the output was usually determined by a fixed similarity threshold. This threshold determines how close two elements (documents or clusters) must be in order to be grouped together. Two teams [LFHDC09, Ven09] used a relatively novel clustering algorithm: Fuzzy ants clustering [SDCCK07]. This algorithm determines by itself the number of clusters without the necessity of a similarity threshold. The similarity measure between documents was commonly handled using the cosine of feature vectors.

Overlapping clusters

Only four teams considered overlapping clusters, but this feature did not play an important role on the WePS-2 data.

6.4 Results of the Evaluation

We were contacted by 32 teams expressing their interest in the clustering task. 17 teams out of them submitted a total of 75 different runs.

[4] http://htmlparser.sourceforge.net/

[5] http://www.crummy.com/software/BeautifulSoup/

[6] The most used tool was Stanford's NER software (http://nlp.stanford.edu/ner/).

6.4.1 Comparison of Systems

Table 6.2 presents the results of the 17 participants and the 3 baseline systems. F values are macro-averaged, i.e., F is computed for every test case, and then averaged over all test cases. When a team submitted multiple runs, we chose the run with best score as the team representative in the ranking.

		Macro-averaged Scores			
		F-measures		B-Cubed	
rank	run	$\alpha = .5$	$\alpha = .2$	Pre.	Rec.
	BEST-HAC-TOKENS	.85	.84	.89	.83
	BEST-HAC-BIGRAMS	.85	.83	.91	.81
1	PolyUHK	.82	.80	.87	.79
2	UVA_1	.81	.80	.85	.80
3	ITC-UT_1	.81	.76	.93	.73
4	XMEDIA_3	.72	.68	.82	.66
5	UCI_2	.71	.77	.66	.84
6	LANZHOU_1	.70	.67	.80	.66
7	FICO_3	.70	.64	.85	.62
8	UMD_4	.70	.63	.94	.60
	HAC-BIGRAMS	.67	.59	.95	.55
9	UGUELPH_1	.63	.75	.54	.93
10	CASIANED_4	.63	.68	.65	.75
	HAC-TOKENS	.59	.52	.95	.48
11	AUG_4	.57	.56	.73	.58
12	UPM-SINT_4	.56	.59	.60	.66
	ALL_IN_ONE	.53	.66	.43	1.00
	CHEAT_SYS	.52	.65	.43	1.00
13	UNN_2	.52	.48	.76	.47
14	ECNU_1	.41	.44	.50	.55
15	UNED_3	.40	.38	.66	.39
16	PRIYAVEN	.39	.37	.61	.38
	ONE_IN_ONE	.34	.27	1.00	.24
17	BUAP_1	.33	.27	.89	.25

Table 6.2: WePS-2 official team ranking using B-Cubed measures

The results according to B-Cubed metrics are shown in Table 6.2. It is worth noticing that:

Purity/Inverse purity

- There are subtle differences, but no major ranking swaps between BCubed and Purity-Inverse Purity rankings (see Table 6.3). The exception is the cheat system baseline, which is no longer one of the best systems according to the new metrics.

Best systems

- The first three teams have similar performance in terms of $F_{0.5}$. Out of these teams, UVA_1 has the most balanced result (0.85 precision, 0.80 recall), and ITC-UT_1 is more precision-oriented (0.93 precision, 0.73 recall), therefore it gets penalised in the $F_{0.2}$ measure. The same team (PolyUHK, which was CU-COMSEM in WePS-1) obtained the best score both in WePS-1 and WePS-2.

- Five teams fall below the ALL_IN_ONE baseline. In spite of all of them reaching higher precision values than the baseline, their combined F measure cannot compensate the perfect recall of that baseline. Note that the ALL_IN_ONE baseline has a BCubed precision higher than expected (0.43), because in average half of the documents in every test set belong to one single person. In the WePS-1 dataset documents were more evenly distributed among the clusters, and therefore the ONE_IN_ONE baseline gave better results than the ALL_IN_ONE baseline strategy. Baselines performance

- The *HAC-TOKENS* and *HAC-BIGRAMS* baselines obtain high precision but poor recall, which places them in the middle of the ranking. Bigrams improve the recall while maintaining a high precision, and so it achieves a better combined score. Note that these baseline systems might be oriented towards precision because the WePS-1 dataset had, in average, very small clusters. HAC systems

- The upper bound systems *BEST-HAC-TOKENS* and *BEST-HAC-BIGRAMS* obtain excellent results, a bit better than the three best teams. This seems to be an indication that the best scoring systems are doing a good job, because they nearly match the behaviour of oracle systems which know which is the best clustering threshold for each test instance. On the other hand, this result also suggests that improving the selection of the clustering threshold may lead to competitive results even with a naive clustering approach. Upper bound systems

6.4.2 Robustness of F Results with Different α Values

UIR

The grouping thresholds chosen by the clustering systems imply a trade-off choice between BCubed Recall and Precision; systems tend to achieve better results according to one metric at the cost of the other. Therefore, the system ranking can suffer drastic changes depending on the α parameter chosen for the F combining function, given that it determines the relative weight assigned to Precision and Recall. This phenomenon was discussed in detail in Chapter 5. In that chapter, the UIR (*Unanimous Improvement Ratio*) measure was proposed in order to check to what extent the detection of a system improvement is biased by the metric weighting scheme (i.e. the α parameter in our case).

Table 6.4 shows the results of applying UIR to the WePS-2 systems. The third column represents the set of systems that are improved by the corresponding system with a UIR>0.25. The fourth column represents the *reference system*, which is defined as follows: given a system a, the system that improves a with maximum UIR. It represents the system with which a should be replaced in order to improve results without losing any partial evaluation metric. Finally, the last column represents the UIR between the system and its reference.

PolyUHK

UIR adds new insights into the evaluation process. First of all, note that, despite

		Macro-averaged Scores			
		F-measures			
rank	run	$\alpha = .5$	$\alpha = .2$	Pur	Inv_Pur
	BEST-HAC-TOKENS	.90	.89	.93	.88
	BEST-HAC-BIGRAMS	.90	.87	.94	.86
1	PolyUHK	.88	.87	.91	.86
2	UVA_1	.87	.87	.89	.87
3	ITC-UT_1	.87	.83	.95	.81
	CHEAT_SYS	.87	.94	.78	1.00
4	UMD_4	.81	.76	.95	.72
5	XMEDIA_3	.80	.76	.91	.73
6	UCI_2	.80	.84	.75	.89
7	LANZHOU_1	.80	.78	.85	.77
8	FICO_3	.80	.76	.90	.73
	HAC-BIGRAMS	.78	.64	.96	.67
9	UGUELPH_1	.74	.84	.64	.95
10	CASIANED_4	.73	.77	.72	.83
	HAC-TOKENS	.71	.64	.96	.60
11	AUG_4	.69	.68	.79	.68
12	UPM-SINT_4	.67	.70	.69	.74
	ALL_IN_ONE	.67	.79	.56	1.00
13	UNN_2	.64	.59	.80	.57
14	ECNU_1	.53	.56	.60	.63
15	PRIYAVEN	.53	.49	.71	.48
16	UNED_3	.51	.48	.71	.48
17	BUAP_1	.37	.30	.89	.27
	ONE_IN_ONE	.34	.27	1.00	.24

Table 6.3: WePS-2 team ranking using Purity/Inverse Purity metrics

the three top-scoring systems having a similar performance in terms of F (0.82, 0.81 and 0.81), PolyUHK is consistently the best according to UIR (it is the reference for 10 systems). In the most extreme case, UIR(PolyUHK,PRIYAVEN)=1, which means that PolyUHK improves both precision and recall of PRIYAVEN for all test cases in the dataset. Therefore, UIR clearly points out a best system, where F alone could only discern a set of three top scoring systems.

Baseline approaches

Although the ALL_IN_ONE baseline is better than five systems according to F, it is not better than any of them according to UIR. In fact, only the ONE_IN_ONE baseline is able to improve some system (BUAP_2). Therefore, UIR also adds the capability of detecting baseline approaches: if a system is adopting a baseline behaviour (for instance, using a very low clustering threshold that ends up setting up one big cluster), F will not clearly signal this problem (the F value obtained is better than five systems), but UIR will certainly signal a problem, because this baseline strategy cannot robustly improve any system.

System	$F_{0.5}$	Improved systems (UIR > 0.25)	Reference system	UIR for the reference system
PolyUHK (S1)	0.82	S2 S4 S6 S7 S8 S11..S17 B_1	-	-
ITC-UT_1 (S2)	0.81	S4 S6 S7 S8 S11..S17 B_1	S1	0.26
UVA_1 (S3)	0.81	S2 S4 S7 S8 S11..S17 B_1	-	-
XMEDIA_3 (S4)	0.72	S11 S13..S17	S1	0.58
UCI_2 (S5)	0.71	S12..S16	-	-
UMD_4 (S6)	0.71	S4 S7 S11 S13..S17 B_1	S1	0.35
FICO_3 (S7)	0.70	S11 S13..S17	S2	0.65
LANZHOU_1 (S8)	0.70	S11..S17	S1	0.74
UGUELPH_1 (S9)	0.63	S4 S12 S14 S16	-	-
CASIANED_5 (S10)	0.63	S12..S16	-	-
AUG_4 (S11)	0.57	S14..S17	S3	0.68
UPM-SINT_1 (S12)	0.53	S14 S16	S1	0.71
ALL_IN_ONE_BASELINE (B_{100})	0.53	B_{COMB}	-	-
UNN_2 (S13)	0.52	S15 S16	S1	0.9
COMBINED_BASELINE (B_{COMB})	0.52	-	B_{100}	0.65
ECNU_1 (S14)	0.42	-	S1	0.9
UNED_3 (S15)	0.41	S16	S1	0.97
PRIYAVEN (S16)	0.39	-	S1	1.00
ONE_IN_ONE_BASELINE (B_1)	0.34	S17	S1	0.29
BUAP_2 (S17)	0.33	-	S6	0.84

Table 6.4: WePS-2 results with BCubed Precision and Recall, F and UIR measures.

6.5 Conclusions

The WePS-2 campaign maintained the high level of participation achieved in its first edition. This campaign has also featured more robust clustering evaluation measures and a more efficient annotation process, based on the experience acquired in WePS-1. The results of the evaluation have been also analysed using a novel approach (Unanimous Improvement Ratio) that tackles the bias introduced by metric weighting schemes.

In parallel with the clustering task, WePS-2 also included a new person Attribute Extraction task [SA09] which is not reported in this thesis. The extraction of biographical attributes can be a valuable source of information for accurate document clustering, but it also represents an important aid for real users to browse the clustering results.

During a discussion panel with WePS-2 we received feedback from participants. A particularly interesting suggestion was that, in order to focus on the comparison of clustering and feature selection methods, participants could share feature vectors provided by the organisation, along with the test collections. Moreover, in a forthcoming edition of the WePS campaign we intend to address the relationships between both tasks in an integrated way. This should provide new insights into the challenges faced by WePS systems.

Part III

Empirical Studies

Chapter 7

The Scope of Query Refinement in the WePS Task

In the context of the WePS task, one question that naturally arises is whether search results clustering can effectively help users for this task. Eventually, a query refinement made by the user – for instance, adding an affiliation or a location – might have the desired disambiguation effect without compromising recall. The hypothesis underlying most research on Web People Search is that query refinement is risky, because it can enhance precision but it will usually harm recall. Adding the current affiliation of a person, for instance, might make information about previous jobs disappear from search results.

The goal of this chapter is to empirically confirm this hypothesis. We want to evaluate the actual impact of using query refinements in the Web People Search (WePS) clustering task (as defined in the framework of the WePS evaluation). In order to do so, we have studied to what extent a query refinement can successfully filter relevant results and which type of refinements are the most successful. In our experiments, we have considered the search results associated to one individual as a set of relevant documents. We have tested the ability of different query refinement strategies to retrieve those documents. Our results are conclusive: in most occasions there is a "near-perfect" refinement that filters out most relevant information about a given person, but this refinement is very hard to predict from a user's perspective.

In Section 7.1, we describe the datasets that were used for our experiments. The experimental methodology and results are presented in Section 7.2. Finally, we present our conclusions in Section 7.4.

7.1 Dataset

We have used the WePS-2 testbed for our experiments. In order to generate query refinement candidates, we extracted several types of features from each document. First, we applied a simple preprocessing to the HTML documents in the corpus, converting them to plain text and tokenising. Then, we extracted tokens and word n-grams for each document (up to four words length). A list of English stopwords

field	F	prec.	recall	cover.
ae_affiliation	0.99	0.98	1.00	0.46
ae_award	1.00	1.00	1.00	0.04
ae_birthplace	1.00	1.00	1.00	0.09
ae_degree	0.85	0.80	1.00	0.10
ae_email	1.00	1.00	1.00	0.11
ae_fax	1.00	1.00	1.00	0.06
ae_location	0.99	0.99	1.00	0.27
ae_major	1.00	1.00	1.00	0.07
ae_mentor	1.00	1.00	1.00	0.03
ae_nationality	1.00	1.00	1.00	0.01
ae_occupation	0.95	0.93	1.00	0.48
ae_phone	0.99	0.99	1.00	0.13
ae_relatives	0.99	0.98	1.00	0.15
ae_school	0.99	0.99	1.00	0.15
ae_work	0.96	0.95	1.00	0.07
stf_location	0.96	0.95	1.00	0.93
stf_organisation	1.00	1.00	1.00	0.98
stf_person	0.98	0.97	1.00	0.82
tokens	1.00	1.00	1.00	1.00
bigrams	1.00	1.00	1.00	0.98
trigrams	1.00	1.00	1.00	1.00
fourgrams	1.00	1.00	1.00	0.98
fivegrams	1.00	1.00	1.00	0.98

Table 7.1: Query refinement results for clusters of size 1

was used to remove tokens and n-grams composed of stopwords. Using the Stanford Named Entity Recognition Tool[1] we obtained the lists of persons, locations and organisations mentioned in each document.

Additionally, we used attributes manually annotated for the WePS-2 Attribute Extraction Task [SA09]. These are person attributes (affiliation, occupation, variations of name, date of birth, etc.) for each individual sharing the name searched. These attributes emulate the kind of query refinements that a user might try in a typical people search scenario.

7.2 Experiments

In our experiments, we consider each set of documents (cluster) related to one individual in the WePS corpus as a set of relevant documents for a person search. For instance, the James Patterson dataset in the WePS corpus contains a total of 100 documents, and 10 of them belong to a British politician named James Patterson. The WePS-2 corpus contains a total of 552 clusters that were used to evaluate the different types of QRs.

Best query refinements

For each person cluster, our goal is to find the best query refinements (QRs);

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

field	F	prec.	recall	cover.
ae_affiliation	0.76	0.99	0.65	0.40
ae_award	0.67	1.00	0.50	0.02
ae_birthplace	0.67	1.00	0.50	0.10
ae_degree	0.63	0.87	0.54	0.15
ae_email	0.74	1.00	0.60	0.16
ae_fax	0.67	1.00	0.50	0.09
ae_location	0.77	1.00	0.66	0.32
ae_major	0.71	1.00	0.56	0.09
ae_mentor	0.75	1.00	0.63	0.04
ae_nationality	0.67	1.00	0.50	0.01
ae_occupation	0.76	0.98	0.65	0.52
ae_phone	0.75	1.00	0.63	0.13
ae_relatives	0.78	0.96	0.68	0.15
ae_school	0.68	0.96	0.56	0.17
ae_work	0.81	1.00	0.72	0.17
stf_location	0.83	0.97	0.77	0.98
stf_organisation	0.89	1.00	0.83	1.00
stf_person	0.83	0.99	0.74	0.98
tokens	0.96	0.99	0.94	1.00
bigrams	0.95	1.00	0.92	1.00
trigrams	0.94	1.00	0.92	1.00
fourgrams	0.91	1.00	0.86	0.99
fivegrams	0.89	1.00	0.84	0.99

Table 7.2: Query refinement results for clusters of size 2

in an ideal case, an expression that is present in all documents in the cluster but not present in documents outside the cluster. For each QR type (affiliation, e-mail, n-grams of various sizes, etc.) we consider all candidates found in at least one document from the cluster, and we pick up the one that leads to the best harmonic mean ($F_{\alpha=.5}$) of precision and recall on the cluster documents (there might be more than one).

For instance, when we evaluate a set of *token* QR candidates for the politician in the James Patterson dataset, we find that among all tokens that appear in the documents of its cluster, ”republican” gives us a perfect score, while “politician“ obtains a low precision (we retrieve documents from other politicians named James Patterson).

Coverage

In some cases a cluster might not have any candidate for a particular type of QR. For instance, manual person attributes like phone number are sparse and will not be available for every individual, whereas tokens and n-grams are always present. We exclude those cases when computing F, and instead we report a *coverage* measure which represents the number of clusters which have at least one candidate of this type of QR. In this way, we know how often we can use an attribute (coverage) and how useful it is whenever it is available (F measure).

These figures represent a ceiling for each type of query refinement: they represent the efficiency of the query when the user selects the best possible refinement

field	F	prec.	recall	cover.
ae_affiliation	0.51	0.96	0.39	0.81
ae_award	0.26	1.00	0.16	0.20
ae_birthplace	0.33	0.99	0.24	0.28
ae_degree	0.37	0.90	0.26	0.36
ae_email	0.35	0.96	0.23	0.33
ae_fax	0.30	1.00	0.19	0.15
ae_location	0.34	0.96	0.23	0.64
ae_major	0.30	0.97	0.20	0.22
ae_mentor	0.23	0.95	0.15	0.22
ae_nationality	0.36	0.88	0.26	0.16
ae_occupation	0.52	0.93	0.40	0.80
ae_phone	0.34	0.96	0.23	0.33
ae_relatives	0.32	0.95	0.22	0.16
ae_school	0.40	0.95	0.29	0.43
ae_work	0.45	0.94	0.34	0.38
stf_location	0.62	0.87	0.53	1.00
stf_organisation	0.67	0.96	0.56	1.00
stf_person	0.59	0.95	0.47	1.00
tokens	0.87	0.90	0.86	1.00
bigrams	0.79	0.95	0.70	1.00
trigrams	0.75	0.96	0.65	1.00
fourgrams	0.67	0.97	0.55	1.00
fivegrams	0.62	0.96	0.50	1.00

Table 7.3: Query refinement results for clusters of size >=3

field	F	prec.	recall	cover.
best-ae	1.00	0.99	1.00	**0.74**
best-all	1.00	1.00	1.00	1.00
best-ner	1.00	1.00	1.00	0.99
best-nl	1.00	1.00	1.00	1.00

Table 7.4: Query refinement results for clusters of size 1

field	F	prec.	recall	cover.
best-ae	0.77	1.00	0.65	**0.79**
best-all	0.95	1.00	0.93	1.00
best-ner	0.92	0.99	0.88	1.00
best-nl	0.96	1.00	0.94	1.00

Table 7.5: Query refinement results for clusters of size 2

for a given QR type.

Size of the cluster

We have split the results in three groups depending on the size of the target cluster: (i) infrequent people, mentioned in only one document (335 clusters of size 1); (ii) people that appear in two documents (92 clusters of size 2), these documents

field	F	prec.	recall	cover.
best-ae	0.60	0.97	0.47	**0.92**
best-all	0.89	0.96	0.85	1.00
best-ner	0.74	0.95	0.63	1.00
best-nl	0.89	0.95	0.85	1.00

Table 7.6: Query refinement results for clusters of size >=3

field	1	2	>=3
ae_affiliation	20.96	17.88	29.41
ae_occupation	20.25	21.79	24.60
ae_work	3.23	8.38	8.56
ae_location	12.66	12.29	8.02
ae_school	7.03	6.70	6.42
ae_degree	3.23	3.91	5.35
ae_email	5.34	6.15	4.28
ae_phone	6.19	5.03	3.21
ae_nationality	0.28	0.00	3.21
ae_relatives	7.03	5.03	2.67
ae_birthplace	4.22	5.03	1.60
ae_fax	2.95	1.68	1.60
ae_major	3.52	3.91	1.07
ae_mentor	1.41	2.23	0.00
ae_award	1.69	0.00	0.00

Table 7.7: Distribution of the person attributes used for the ”best-ae“ query refinement strategy

often belong to the same domain, or are very similar; and (iii) all other cases (125 clusters of size >=3).

QR types

We also report on the aggregated results for certain subsets of QR types. For instance, if we want to know what results will get a user that picks the best person attribute, we consider all types of attributes (e-mail, affiliation, etc.) for every cluster, and pick up the ones that lead to the best results.

We consider four groups: (i) *best-all* selects the best QR among all the available QR types (ii) *best-ae* considers all manually annotated attributes (iii) *best-ner* considers automatically annotated NEs; and (iv) *best-ng* uses only tokens and n-grams.

7.3 Results

Results of the evaluation for each cluster size (one, two, more than two) are presented in Tables 7.1, 7.2 and 7.3. These tables display results for each QR type. Then Tables 7.4, 7.5 and 7.6 show the results for aggregated QR types.

Best refinements

Two main results can be highlighted: (i) The best overall refinement is, in average, very good ($F = .89$ for clusters of size ≥ 3). In other words, there is usually at least one QR that leads to (approximately) the desired set of results; (ii) this best refinement, however, is not necessarily an intuitive choice for the user. One would expect users to refine the query with a person's attribute, such as his affiliation or location. But the results for the best (manually extracted) attribute are significantly worse ($F = .60$ for clusters of size ≥ 3), and they cannot always be used (coverage is .74, .79 and .92 for clusters of size 1, 2 and ≥ 3).

Manual attributes

The manually tagged attributes from WePS-2 are very precise, although their individual coverage over the different person clusters is generally low. Affiliation and occupation, which are the most frequent attributes, obtain the largest coverage (0.81 and 0.80 for sizes ≥ 3). Moreover, the recall of this type of QRs is low in clusters of two, three or more documents. When evaluating the "best-ae" strategy, we found that there is at least one manual attribute that can be used as QR with high precision in many clusters. This is mostly the case for clusters of three or more documents (0.92 coverage) and it decreases with smaller clusters, probably because there is less information about the person and, consequently less biographical attributes are to be found.

In Table 7.7 we show the distribution of the actual QR types selected by the "best-ae" strategy. The best type is affiliation, which is selected in 29% of the cases. Affiliation and occupation together cover around half of the cases (54%), and the rest is a long tail in which each attribute makes a small contribution to the total. Again, this is a strong indication that the best refinement is probably very difficult to predict *a priori* for the user.

Named entities

Automatically recognised named entities in the documents obtain better results in general than manually tagged attributes. This is probably due to the fact that they can capture all kinds of related entities, or just entities that happen to coocur with the person name. For instance, the pages of a university professor that is usually mentioned together with his PhD students could be refined with any of their names. This circumstance shows us that a good QR can be any information related to the person, but also that we might need to know the person very well in advance in order to choose this QR.

Tokens and n-grams

Tokens and n-grams give us a kind of "upper boundary" of what is possible to achieve using QRs. Furthermore, they include almost anything that is found in the manual attributes and the named entities. They also frequently include QRs that are not realistic for a human refinement. For instance, in clusters of just two documents it is not uncommon that both pages belong to the same domain or that they are near duplicates. In those cases, tokens and ngram QR will probably include non informative strings. In some cases, the QRs found are neither directly biographical or related NEs, but topical information (e.g. the term "soccer" in the pages of a football player or the ngram "alignment via structured multilabel" that is the title of a paper written by a Computer Science researcher). These cases widen, even more, the range of effective QRs. The overall results of using tokens and n-grams are almost perfect for all clusters, but at the cost of considering every possible bit of information about the person or even unrelated text.

7.4 Conclusions

In this chapter we have studied the potential effects of using query refinements to perform the Web People Search task. We have shown that, although in general there are query refinements that perform well to retrieve the documents of most individuals, the nature of these ideal refinements varies widely in the studied dataset, and there is no single intuitive strategy leading to robust results. Even if the attributes of the person are well known beforehand (which is hardly realistic, given the fact that, in most cases, this is precisely the information needed by the user), there is no way of anticipating which expression will lead to good results for a particular person. This leads us to confirm that search results clustering might indeed be of practical help in real world situations.

As to future work, we are keen to set up interactive experiments in order to study the way in which search engine users perform name query refinements. We also intend to compare the upper boundary of query refinements to the actual performance of users.

Chapter 8

The Role of Named Entities in WePS

Our goal in this chapter is to study which document features can contribute to the WePS task. More specifically, we try to find out the role that can be played by named entities (NEs).

NEs have been extensively used in name disambiguation, as shown in Section 2.2. Moreover, among the 16 teams that submitted results for the first WePS campaign, 10 of them[1] used NEs in their document representation. This makes NEs the second most common type of feature – only the BoW feature was more popular. Other features used by the systems include noun phrases [CM07b], word n-grams [PM07], emails and URLs [dVAdPSVD07], etc. In 2009, the WePS-2 campaign showed similar trends regarding the use of NE features.

Due to the complexity of systems, the results of the WePS evaluation do not provide a direct answer regarding the advantages of using NEs over other computationally lighter features such as BoW or word n-grams. But the WePS campaigns did provide a useful, standardised resource to perform a type of studies that were not possible in the past. In the next section, we describe this dataset as well as how it has been adapted for our purposes.

In this part of our thesis we intend to answer the following questions: (i) How reliable is NEs overlap between documents as a source of evidence to cluster pages? (ii) How much recall does it provide? (iii) How unique is this signal? (i.e. is it redundant with other sources of information such as n-gram overlap?); and (iv) How sensitive is this signal to the peculiarities of a given NE recognition system, such as the granularity of its NE classification and the quality of its results?

Our aim here is to reach conclusions which are are not tied to a particular choice of Clustering or Machine Learning algorithms. We have made two decisions in this direction: first, we have focused on the problem of deciding whether two web pages refer to the same individual or not (page coreference task). This is the kind of relatedness measure that most clustering algorithms use, but, in this way, we can

[1] By team ID: CU-COMSEM, IRST-BP, PSNUS, SHEF, FICO, UNN, AUG, JHU1, DFKI2, UC3M13

factor out the algorithm and its parameter settings alike. Second, we have developed a measure, *Maximal Pairwise Accuracy* (PWA) which, given an information source for the problem, estimates an upper bound for the performance of any Machine Learning algorithm using this information. We have used PWA as the basic metric to study the role of different document features in solving the coreference problem, and then we have checked the predictive power of PWA with a Decision Tree algorithm.

The remainder of the chapter is organized as follows. First, we describe our experimental settings (datasets and features we have used) in Section 8.1 and our empirical study in both Sections 8.2 and 8.3. In Section 8.4, we present the results of applying the learned feature combinations to the clustering task. The chapter ends with some conclusions in Section 8.5.

8.1 Experimental Settings

We have used the testbeds from WePS-1 (Chapter 3)[2] and WePS-2 (Chapter 6) evaluation campaigns.

8.1.1 Features

Token-based

The evaluated features can be grouped in four main groups: token-based, word n-grams, phrases and NEs. Whenever it is possible, we have generated *local* versions of these features that only consider the sentences of the text mentioning the ambiguous person name[3]. Token-based features considered include document full text tokens, lemmas (using the OAK analyser, see below), title, snippet (returned in the list of search results) and URL (tokenised using non alphanumeric characters as boundaries) tokens. English stopwords were removed, including web specific stopwords, as file and domain extensions, etc.

Word n-grams

We generated word n-grams of length 2 to 5, using the sentences found in the document text. Punctuation tokens (commas, dots, etc) were generalised as the same token. N-grams were discarded whenever they consisted only of stopwords or whenever they did not contain at least one token formed by alphanumeric characters (e.g. n-grams like “at the” or “# @” were filtered out). Noun phrases (using OAK analyser) were detected in the document and filtered in a similar way.

Named entities

Named entities were extracted using two different tools: the Stanford NE Recogniser and the OAK System[4].

Stanford NE Recogniser[5] is a high-performance Named Entity Recognition (NER) system based on Machine Learning. It provides a general implementation of linear chain Conditional Random Field sequence models and includes a model

[2] The WePS-1 corpus includes data from the Web03 testbed [Man06] which follows similar annotation guidelines, although the number of document per ambiguous name is more variable.

[3] A very sparse feature might never occur in a sentence with the person name. In that cases there is no *local* version of the feature.

[4] From the output of both systems we have discarded person NEs made of only one token (these are often first names that significantly deteriorate the quality of the comparison between documents).

[5] http://nlp.stanford.edu/software/CRF-NER.shtml

trained on data from CoNLL, MUC6, MUC7, and ACE newswire. Three types of entities were extracted: person, location and organisation.

OAK[6] is a rule based English analyser that includes many functionalities (POS tagger, stemmer, chunker, Named Entity (NE) tagger, dependency analyser, parser, etc). It provides a fine grained NE recognition covering 100 different NE types [Sek08]. Given the sparseness of most of these fine-grained NE types, we have merged them in coarser groups: event, facility, location, person, organisation, product, periodx, timex and numex.

Baseline NE

We have also used the results of a baseline NE recognition for comparison purposes. This method detects sequences of two or more uppercased tokens in the text, and discards those that are found lowercased in the same document or that are composed solely from stopwords.

Other features

Other features to be taken into account are: emails, outgoing links found in the web pages and two boolean flags that indicate whether a pair of documents is linked or belongs to the same domain. Because of their low impact in the results, these features have not received an individual analysis, but they are included in the "all features" combination in Figure 8.7.

8.1.2 Reformulating WePS as a Classification Task

Given the fact that our goal is to study the impact of different features (information sources) in the task, a direct evaluation in terms of clustering has serious disadvantages. Given the output of a clustering system, it is not straightforward to assess why a document has been assigned to a particular cluster. There are at least three different factors involved in this process: the document similarity function, the clustering algorithm and its parameter settings. Features are part of the document similarity function, but its performance in the clustering task depends on the other factors as well. This makes it difficult to perform error analysis in terms of the features used to represent the documents.

Therefore, we have decided to transform the clustering problem into a classification problem: deciding whether two documents refer to the same person. A similar classification approach can be found in previous works on name disambiguation (see Section 2.3). Each pair of documents in a name dataset is considered a classification instance. Instances are labelled as coreferent (if they share the same cluster in the gold standard) or non coreferent (if they do not share the same cluster). Hence, we can evaluate the performance of each feature separately by measuring its ability to rank coreferent pairs higher and non coreferent pairs lower. In the case of feature combinations we can study them by training a classifier or using the maximal pairwise accuracy methods (explained in Section 8.3).

Each instance (pair of documents) is represented by the similarity scores obtained by using different features and similarity metrics. For each feature we have calculated three similarity metrics: Dice's coefficient, cosine (using standard tf.idf weighting) and a measure that simply counts the size of the intersection set for a

[6] http://nlp.cs.nyu.edu/oak . OAK was also used to detect noun phrases and extract lemmas from the text.

given feature between both documents. After testing these metrics, we found that Dice provides the best results across different feature types. Differences between Dice and cosine were consistent, although they were not especially large. A possible explanation is that Dice does not take into account the redundancy of an n-gram or NE in the document, whereas the cosine distance does. This can be a crucial factor involved, for instance, in the document retrieval by topic, but it does not seem to be the case when dealing with name ambiguity.

The resulting classification testbed consists of 293,914 instances with the distribution shown in Table 8.1, where each instance is represented by 69 features.

	true	false	total
WePS1	61,290	122,437	183,727
WePS2	54,641	55,546	110,187
WePS1+WePS2	115,931	177,983	293,914

Table 8.1: Distribution of instances in the WePS classification testbed

8.2 Analysis of Individual Features

There are two main aspects related to the usefulness of a feature for WePS task. The first one is its performance, that is to say, to what extent the similarity between two documents according to a feature implies that both mention the same person. The second aspect is to what extent a feature is orthogonal or redundant with respect to the standard token based similarity.

8.2.1 Feature Performance

According to the transformation of WePS clustering problem into a classification task (described in Section 8.1.2), we follow the next steps to study the performance of individual features. First, we compute the Dice coefficient similarity over each feature for all document pairs. Then we rank the document pair instances according to these similarities. A good feature should rank positive instances on top. If the number of coreferent pairs in the top n pairs is t_n and the total number of coreferent pairs is t, then $P = \frac{t_n}{n}$ and $R = \frac{t_n}{t}$. We plot the obtained precision/recall curves in Figures 8.1, 8.2, 8.3 and 8.4.

Non-NE

From these figures, we can draw the following conclusions: First, considering subsets of tokens or lemmatised tokens does not outperform the basic token distance (Figure 8.1 compares token-based features). We see that only local and snippet tokens perform slightly better at low recall values, but do not go beyond recall 0.3. Second, shallow parsing or n-grams longer than 2 do not seem to be effective, but using bi-grams improves the results in comparison with tokens. Figure 8.2 compares n-grams of different sizes with noun phrases and tokens. All in all, noun phrases have a poor performance, and bi-grams give the best results up to recall 0.7.

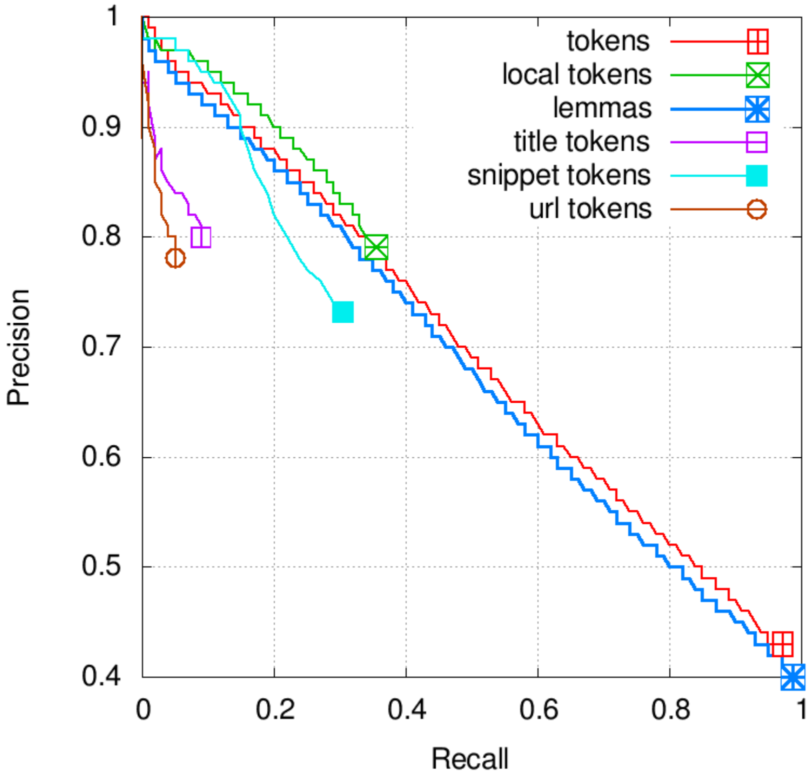


Figure 8.1: Precision/Recall curve of token-based features

Four-grams give slightly better precision but only reach 0.3 recall, and three-grams do not give better precision than bi-grams.

NE vs. tokens

Third, individual types of NEs do not improve over tokens. Figure 8.3 and Figure 8.4 display the results obtained by the Stanford and OAK NER tools respectively. In the best case, Stanford person and organisation named entities obtain results that match the tokens feature, but only at lower levels of recall.

NER systems

Finally, using different NER systems clearly leads to different results. Surprisingly, the baseline NE system yields better results in a one to one comparison, although it must be noted that this baseline agglomerates different types of entities that are separated in the case of Stanford and OAK, and this has a direct impact on its recall. The OAK results are below the tokens and NE baseline, possibly due to the sparseness of its very fine grained features. In NE types, cases such as person and organisation results are still lower than obtained with Stanford.

8.2.2 Redundancy

In addition to performance, named entities (as well as other features) are potentially useful for the task only if they provide information that complements (i.e. that does

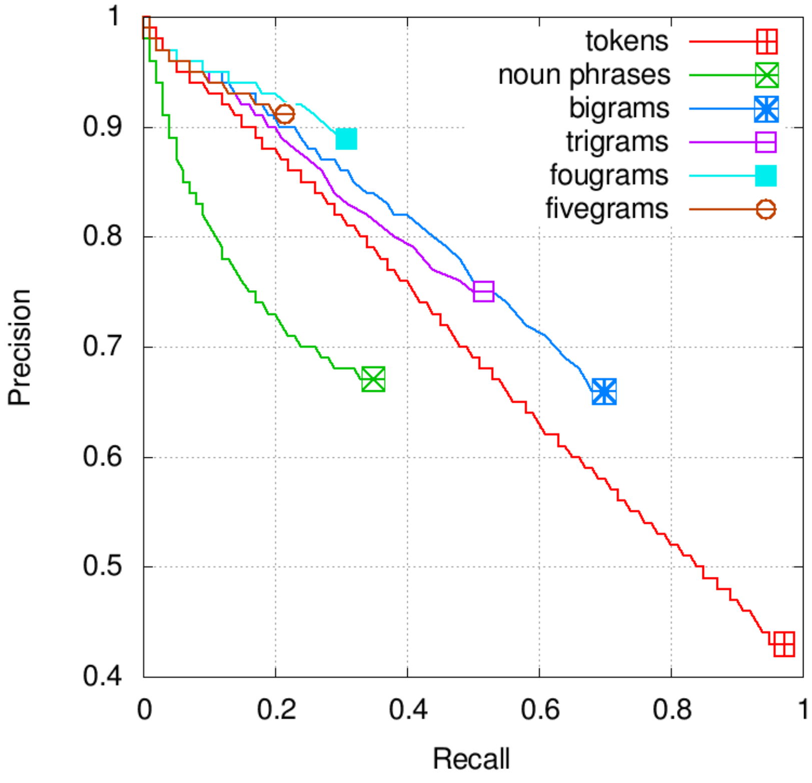


Figure 8.2: Precision/Recall curve of word n-grams

not substantially overlap) the basic token based metric. To estimate this redundancy, let us consider all document tuples of size four $< a, b, c, d >$. In 99% of the cases, token similarity is different for $< a, b >$ than for $< c, d >$. We take combinations such that $< a, b >$ are more similar to each other than $< c, d >$ according to tokens. That is:

$$\text{sim}_{\text{token}}(a, b) > \text{sim}_{\text{token}}(c, d)$$

Then, for any other feature similarity $sim_x(a, b)$, we will talk about *redundant* samples when $sim_x(a, b) > sim_x(c, d)$, *non redundant* samples when $sim_x(a, b) < sim_x(c, d)$, and *non informative* samples when $sim_x(a, b) = sim_x(c, d)$. If all samples are redundant or non informative, then sim_x does not provide additional information for the classification task.

NE redundancy

Figure 8.5 shows the proportion of redundant, non redundant and non informative samples for several similarity criteria, as compared to token-based similarity. In most cases, NE based similarities give little additional information: the baseline NE recogniser, which has the largest independent contribution, gives additional information in less than 20% of cases.

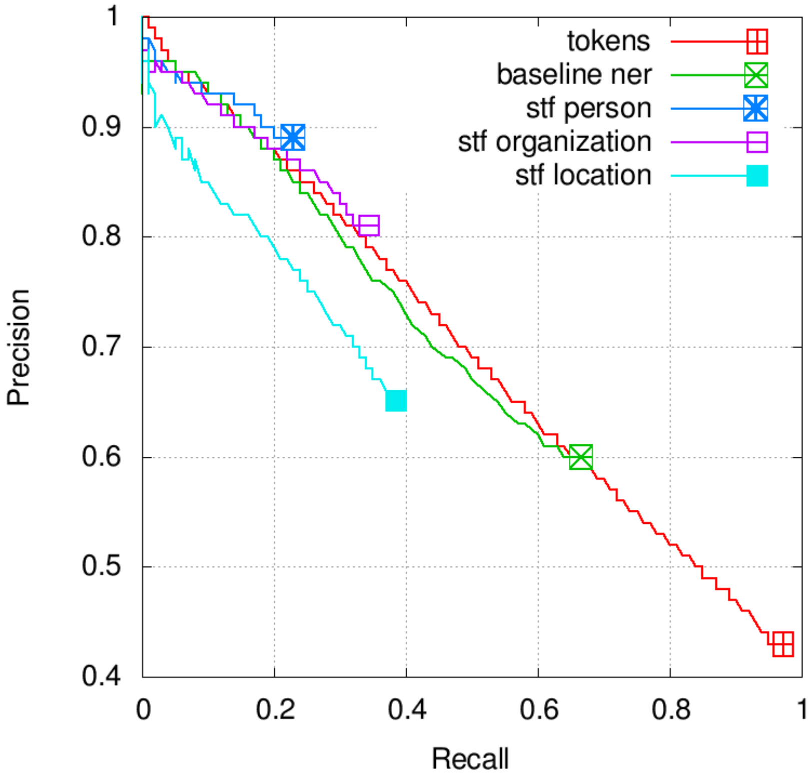


Figure 8.3: Precision/Recall curve of NEs obtained with the Stanford NER tool

In summary, analysing individual features, the NEs do not outperform BoW in terms of the classification task. In addition, NEs tend to be redundant regarding BoW. However, if we are able to combine the contributions of the different features optimally, the BoW approach could be improved. We address this issue in the next section.

8.3 Analysis of Feature Combinations

Machine Learning algorithm

Up to now, we have analysed the usefulness of individual features for the WePS Task. Nevertheless, this analysis begs to ask to what extent the NE features can contribute to the task whenever they are combined together and with token and n-gram based features. First, we use each feature combinations as the input for a Machine Learning algorithm. More specifically, we use a Decision Tree algorithm and WePS-1 data for training and WePS-2 data for testing.

Maximal Pairwise Accuracy

Results obtained with this setup, however, can be dependent on the choice of the ML approach. To overcome this problem, in addition to the results of a Decision Tree Machine Learning algorithm, we introduce a *Maximal Pairwise Accuracy*

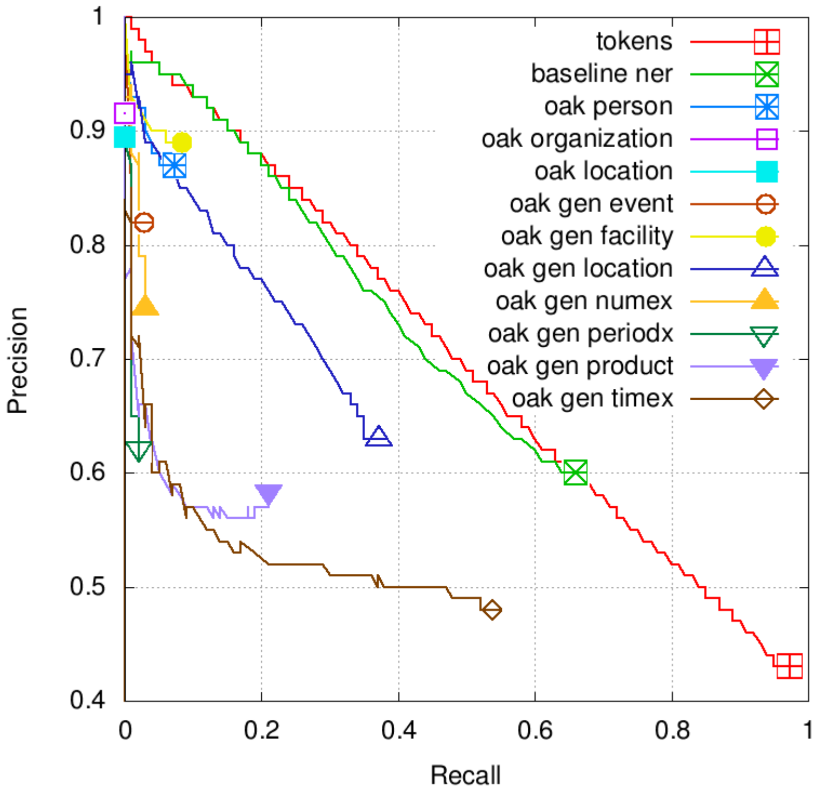


Figure 8.4: Precision/Recall curve of NEs obtained with the OAK NER tool

(MPA) measure that provides an upper bound for any machine learning algorithm using a feature combination.

We can estimate the performance of an individual similarity feature x by computing the probability of the similarity $x(a, a')$ between two pages referring to the same person being higher than the similarity $x(b, c)$ between two pages referring to different people. Let us call this estimation *Pairwise Accuracy*:

$$PWA = \text{Prob}(x(a, a') > x(c, d))$$

When combining a set of features $X = \{x_1 \dots x_n\}$, a perfect Machine Learning algorithm would learn to always "listen" to the features providing correct information and ignore the features giving erroneous information. In other words, if at least one feature gives correct information, then the perfect algorithm would produce a correct output. This is what we call the *Maximal Pairwise Accuracy* estimation of an upper bound for any ML system using the set of features X:

$$\text{MaxPWA}(X) =$$

$$\text{Prob}(\exists x \in X . x(a, a') > x(c, d))$$

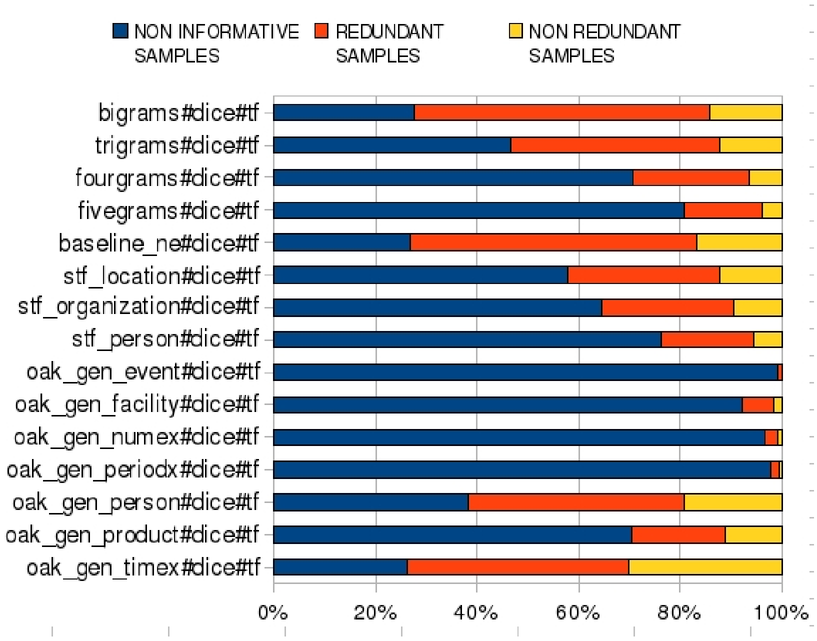


Figure 8.5: Independence of similarity criteria with respect to the token based feature

The upper bound (MaxPWA) of feature combinations happens to be highly correlated with the PWA obtained by the Decision Tree algorithm (using its confidence values as a similarity metric). Figure 8.6 shows this correlation for several feature combinations. This is an indication that the Decision Tree is effectively using the information in the feature set.

Decision Trees vs. Maximal Pairwise Accuracy

Figure 8.7 shows the PWA upper bound estimation and the actual PWA performance of a Decision Tree ML algorithm for three combinations: (i) all features; (ii) non linguistic features, i.e., features which can be extracted without natural language processing machinery: tokens, url, title, snippet, local tokens, n-grams and local n-grams; and (iii) just tokens. The results show that, according to both the Decision Tree results and the upperbound (MaxPWA), adding new features to tokens improves the classification. However, using non-linguistic features obtains similar results than using all features. Our conclusion then is that NE features are useful for the task, but do not seem to offer a competitive advantage when compared with non-linguistic features, and are more computationally expensive. Note that we are using NE features in a direct way: our results do not exclude the possibility of effectively exploiting NEs in more sophisticated ways, such as, for instance, exploiting the underlying social network relationships between NEs in the texts.

8.4 Results on the Clustering Task

Clustering algorithm

In order to validate our results, we have tested whether the classifiers learned with our feature sets lead to competitive systems for the clustering task. In order to do so,

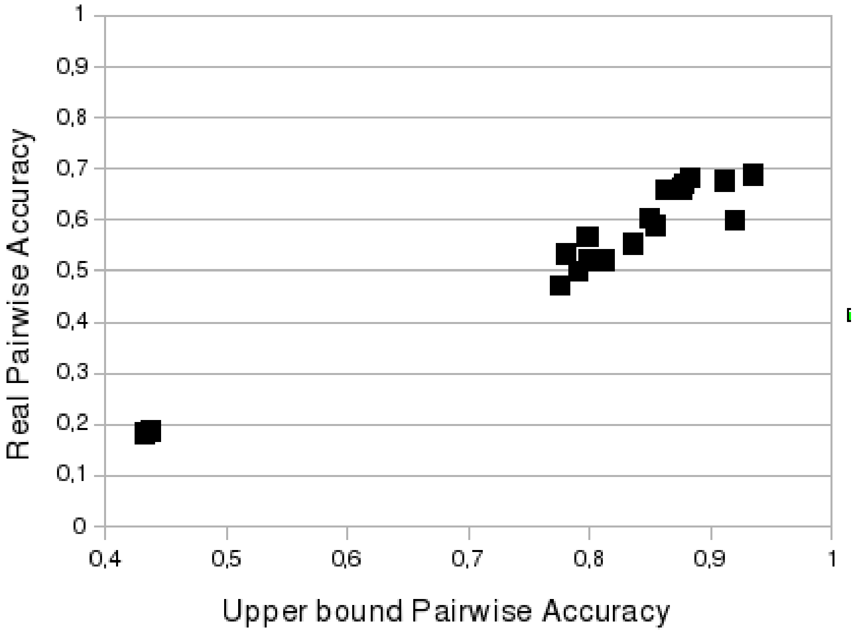


Figure 8.6: Estimated PWA upper bound versus the real PWA of decision trees trained with feature combinations

we use the output of the classifiers as similarity metrics for a particular clustering algorithm, using WePS-1 to train the classifiers and WePS-2 for testing.

Distance threshold

We have employed a Hierarchical Agglomerative Clustering algorithm (HAC) with single linkage, using the classifier's confidence value in the negative answer for each instance as a distance metric[7] between document pairs. HAC is the algorithm used by some of the best performing systems in the WePS-2 evaluation (see Section 6.3). The distance threshold was trained using the WePS-1 data. We report results with the official WePS-2 evaluation metrics: extended B-Cubed Precision and Recall [AGAV08].

Features

Two Decision Tree models were evaluated: (i) *ML-ALL* is a model trained using all the available features (which obtains 0.76 accuracy in the classification task) (ii) *ML-NON_LING* was trained with all the features except for OAK and Stanford NEs, noun phrases, lemmas and gazetteer features (which obtains 0.75 accuracy in the classification task). These are the same classifiers considered in Figure 8.7.

System comparison

Table 8.2 shows the results obtained in the clustering task by the two DT models, together with the four top scoring WePS-2 systems and the average values for all WePS-2 systems. We found that a ML based clustering using only non linguistic information slightly outperforms the best participant in WePS-2. Surprisingly, adding linguistic information (NEs, noun phrases, etc.) has a small negative impact on the results (0.81 versus 0.83), although the classifier with linguistic information was a bit better than the non-linguistic one. This seems to be another indication that

[7] The DT classifier output consists of two confidence values, one for the positive and one for the negative answer, that add up to 1.0 .

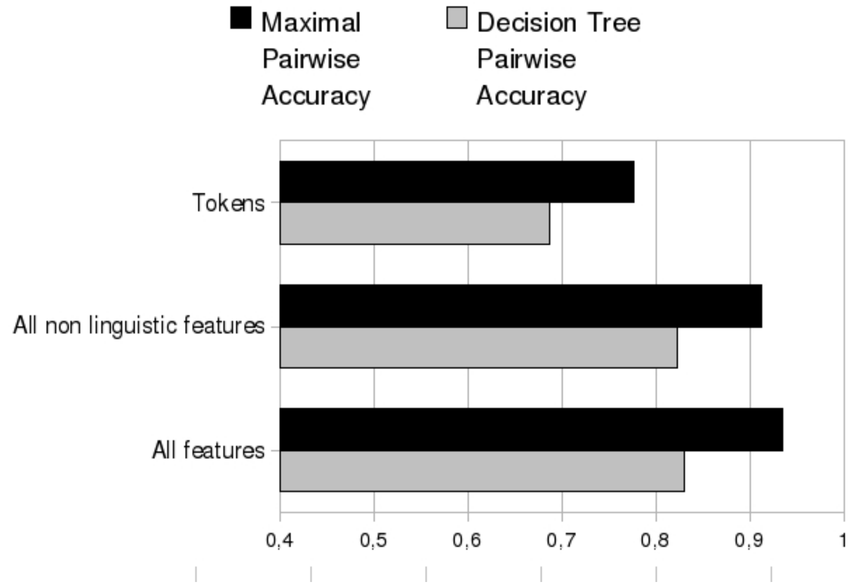


Figure 8.7: Maximal Pairwise Accuracy vs. results of a Decision Tree

		B-Cubed	
run	F-$\alpha =_{0.5}$	Pre.	Rec.
ML-NON_LING	.83	.91	.77
S-1	.82	.87	.79
ML- ALL	.81	.89	.76
S-2	.81	.85	.80
S-3	.81	.93	.73
S-4	.72	.82	.66
WePS-2 systems aver.	.61	.74	.63

Table 8.2: Evaluation of Decision Tree models on the WePS-2 clustering task

the advantages of using noun phrases and other linguistic features to improve the task are non-obvious to say the least.

8.5 Conclusions

We have presented an empirical study that tries to determine the potential role of several sources of information to solve the Web People Search clustering problem, with a particular focus on studying the role of named entities in the task.

To abstract the study from the particular choice of a clustering algorithm and a parameter setting, we have reformulated the problem as a co-reference classification task: deciding whether two pages refer to the same person or not. We have also proposed the *Maximal Pairwise Accuracy* estimation that establish an upper bound for the results obtained by any Machine Learning algorithm using a particular set of features.

Our results indicate that (i) named entities do not provide a substantial competitive advantage in the clustering process when compared to a rich combination of simpler features that do not require linguistic processing (local, global and snippet tokens, n-grams, etc.); (ii) results are sensitive to the NER system used: when using all NE features for training, the richer number of features provided by OAK seems to have an advantage over the simpler set of NE classes in Stanford NER and a baseline NE system.

This is not exactly a prescription against the use of NEs for Web People Search, because linguistic knowledge can be useful for other aspects of the problem, such as visualisation of results and description of the persons/clusters obtained: for example, from a user point of view a network of the connections of a person with other persons and organisations (which can only be done with NER) can be part of a person's profile and may help to provide a summary of the cluster contents. And yet, from the perspective of the clustering problem *per se*, a direct use of NEs and other linguistic features does not seem to pay off.

Chapter 9

Conclusions

We now summarise the results obtained in this work, highlight our contributions to the research topic and discuss our plans for future work.

9.1 Contributions

In this thesis, we have made a number of contributions to the problem of name ambiguity while searching for people in the World Wide Web. These contributions focused on five main topics: (i) a study of the actual necessity to solve this problem automatically, (ii) the development of reference test collections, (iii) the development of improved evaluation metrics, (iv) a thorough study of the relevance of predicting the number of clusters and its implication in the evaluation process and (v) an in-depth study of the role named entities and other linguistic features play in this task. Our work has led to a number of scientific conclusions and also to the release of a number of software packages for the research community; both types of contributions are summarised below.

9.1.1 The Need for Name Disambiguation Systems

Our experimentation on the scope of query refinement strategies (see Chapter 7) has provided us with empirical evidence of the actual need for automatic approaches to the task. We have found that in most cases there is an optimal query refinement (i.e. a number of words added to the person name) which leads to near optimal precision and recall. However, this high performance query refinement is unlikely to be hypothesized by an actual user in a practical search scenario. Therefore, it is indeed necessary to develop automatic approaches in order to help users to find all relevant information about the person they are looking up.

This result supports the interest raised by the problem both in the scientific community and in the Web search business (see Chapter 1). As it happens, given the fact that there usually is a near-perfect refinement, this trend can be used to develop new approaches to the problem (search for optimal query refinements).

9.1.2 Development of Reference Test collections

We carried out a dedicated evaluation campaign, WePS, held in 2007 (as a SemEval 2007 task) and in 2009 (as a WWW 2009 workshop). These campaigns laid the foundations of a proper scientific study of the Web People Search problem. These were our main accomplishments:

- Standardisation of the problem: at present, a majority of researchers focuses on the problem as a search results mining (clustering and information extraction) task, as it has been defined in WePS.
- Creation of standard benchmarks for the task: since the launch of the first WePS campaign in 2007, the number of publications related to Web People Search has grown substantially. Most of them use the WePS test suites as a *de facto* standard benchmark. As of summer 2009 there were already more than 70 research papers citing WePS overviews; this not only suggests that WePS has indeed become a standard reference for the task, but also that it has contributed to raise the interest in this kind of research problems.

WePS-1 and WePS-2 datasets were built following a consistent methodology, and altogether provide a reasonable benchmark with around 8,000 manually annotated documents (including an exhaustive annotation of person attributes in the WePS-2 dataset). The decision to create the testbeds manually was supported by careful study of previous research. Other testbeds using automatic annotation methods (pseudo-ambiguity) do not provide a realistic distribution of individuals associated to a name (see Section 2.1.2). In this thesis, we have shown this is a key aspect of the task, since the number of individuals sharing a person name is difficult to predict. Even names obtained from the same sources (in our case, Wikipedia entries and a list of prominent scholars in a specific Computer Science field) present a wide dispersion of their ambiguity. Furthermore, the overall degree of ambiguity does not seem to follow a normal distribution.

Nevertheless, manual annotation constitutes an expensive, somewhat tricky process (see Sections 3.2.3 and 6.1.3). Before starting the WePS campaign, we expected annotation to be much simpler than for a Word Sense Disambiguation testbed. After all, we just had to distinguish homographs, and being able to distinguish whether two documents refer to two different (and often unrelated) people seemed quite an easy task to accomplish. It appears we had too optimistic expectancies: web documents are often very short, and they tend to provide partial descriptions of people. When comparing two documents, we usually have two partial descriptions which are more or less likely to be compatible. For instance, if two people share the same name, the same job and they live in the same city, they are probably (but not certainly!) the same, the probability depending on the actual size of the city, the specificity of the job and the originality of the name. For this reason, if we want to create a consistent testbed, it is utterly necessary to train assessors to reach a common implicit evidence threshold for clustering two documents.

Note that we had to remove certain (otherwise valid) types of documents, such as genealogies and public (that is, Facebook-like) profiles from social networks, because either they introduce a great deal of uncertainty in the clustering process (Facebook public profiles are usually very thin) or they require too much annotation effort (as is usually case with genealogies). Even after removing these pages, there still was a considerable number of documents humans could not annotate with absolute certainty. In any case we found that, although the annotation task is difficult, rankings are stable across different annotators (similarly to what happens in traditional IR test collections): when evaluating systems using *gold standards* produced independently by two annotators there was no significant changes in the system ranking (see Section 3.2.3).

9.1.3 Development of Improved Clustering Evaluation Metrics

The first step we need to take in order to assess the appropriateness of an evaluation metric is to define the set of constraints that must be satisfied. We have seen that the constraints found in the state of the art can be summarised in just four boundary conditions, setting aside scale aspects (see Section 4). When we examined the variety of clustering metrics that had been proposed, we grouped them in families and observed that metrics in each family satisfied the same basic conditions. Surprisingly all evaluation metrics (with the notable exception of BCubed precision and recall), did not satisfy all four conditions.

Another aspect we have analysed is the evaluation of overlapping clustering. In spite of overlapping not being a very frequent phenomenon in this task, it is necessary to take it into account during the evaluation process. After extending the BCubed metrics for overlapping clusters, our experiments showed that they overcame the limitations of other standard clustering metrics.

Finally, we also proposed the *Unanimous Improvement Ratio* as a measure that complements Precision and Recall weighting functions (most prominently Van Rijsbergen's F measure) indicating the robustness of differences in F to changes in the α parameter that assigns the relative weights of Precision and Recall. Our Unanimous Improvement Ratio can be used in any evaluation methodology that involves combining the contribution from different quality metrics, but it is of particular value in the context of text clustering tasks. One of the reasons for this circumstance is the crucial role played by the clustering stopping criterion which determines the number of clusters in the output. In terms of the WePS task, this criterion should match with the number of individuals associated with a name, that is, the degree of ambiguity.

9.1.4 The Relevance of the Clustering Stopping Criterion

Both in the state of the art collections and in our own testbeds, we have seen that the degree of ambiguity is quite variable and it is not as predictable as expected, even when comparing names from the same source (e.g. encyclopedia). This aspect of the problem presents a challenge for systems. According to our experiments, a simple baseline system can achieve higher scores than the best team in WePS competitions by using the best possible threshold for each topic (see Section 8.4). Therefore, the accurate estimation of the number of clusters represents a key aspect of the disambiguation process. Nevertheless, a good cluster stopping criterion has proved to be insufficient for achieving a high performance. Our experiments indicate that training the stopping criteria over a baseline approach achieves poor results (see Section 8.4). However, whenever the relative weight of similarity features is trained as well, a robust and competitive result is obtained (see Section 6.4).

In terms of evaluation, the stopping criterion determines the trade-off between precision and recall metrics (e.g. Purity and Inverse Purity). We have shown that this trade-off is more prominent across clustering systems than in other tasks (see Section 5.1). This notion leads to a high variability in the rankings depending on the relative weight of individual metrics. That is why the Unanimous Improvement Ratio provides crucial information in the context of the task.

9.1.5 The Role of Named Entities

Most recent work on name ambiguity uses person names, organisations and locations coocurring in documents as a key feature for the task. Consequently, named entity recognition has been used extensively in name disambiguation systems. One of our goals here was to check whether linguistic information and named entities in particular – which are computationally costly to extract, as compared to the usual Information Retrieval features – can indeed provide a competitive advantage to solve name ambiguity in Web People Search.

Our experiments have shown that named entities are not necessarily more useful than other features such as word n-grams (see Section 8). Given the fact that this notion represents a negative result, we have been particularly exhaustive in our experimentation. We have compared the performance of different document representations in terms of detecting coreferent document pairs and also in terms of clustering. In addition, we have considered feature types both individually and in combination. During the study, we found that named entities did not achieve better results than word n-grams in this particular case. The combination of both types of features using machine learning algorithms does not seem able to provide better results than n-grams. In broad terms, our experiments indicate that linguistic information (named entities, noun phrases, etc.) cannot be directly exploited by a machine learning process to provide better results than computationally cheap features such as word n-grams.

But the latter result does not necessarily represent a prescription against using named entities in order to solve the problem. We have just used explicit named entity occurrences in the documents as candidate features for Machine Learning algorithms, but there are more sophisticated ways of using this type of information. For instance, named entities are interrelated, and the social network of entities can perhaps be used to add implicit information into the clustering process (or even more so, to lead the whole process!). In addition, extracting named entities is probably useful to present the information to the user and simplify the task of choosing the intended person/cluster.

9.1.6 Testbeds and Software Released

During the research phase of this thesis, three products have been both generated and made available to the research community:

- The large testbed for the Web People Search task [1]. This testbed consists of three manually annotated collections developed for the WePS-1 and WePS-2 evaluation campaigns [AGV05, AGS07, AGS09]. The WePS datasets have been used in research works other than the WePS campaigns (see for instance [KNTM08, CMP08]). These specific collections have been made available pre-processed, semantically annotated with NLP tools and indexed with the Lucene search software. Both are freely downloadable from the Web.

- A graphical user interface has been designed to assist the manual grouping of document collections. The interface has been used for the generation of the WePS-2 clustering test collection and can be employed in the annotation process of other document clustering tasks.

- An evaluation package has been released, including standard clustering evaluation metrics, the extended BCubed metrics and the Unanimous Improvement Ratio .

9.2 Further Directions

There are at least four research questions which remain open for future work:

The first one consists on the exploration of new approaches to the representation of documents. Some of the aspects that could be studied in the future include feature weighting techniques, the definition of new similarity metrics between documents and the usage of external resources such as Wikipedia or the Google N-gram corpus. These studies should set the ground for stronger baseline systems in future evaluation campaigns.

Regarding the evaluation, the methods proposed for WePS task can be applied to new domains and tasks. In addition, we are currently working on the definition of clustering quality metrics that take into account the estimated cognitive effort of the user whenever exploring search results. Although some models have been proposed [HP96, Leu01, CPGV05], they do not consider aspects such as the probability that a cluster would be explored by the user. Generally speaking, we have to study how best to assist users when searching people in the Web by considering the interactive aspects of the task and bringing actual users into the evaluation loop.

Finally, there seem to be many possibilities for future WePS campaigns. The Web People Search problem is related also to the problem of searching organisations in the Web. It seems we could easily extend the task in this direction. It is also interesting to consider the multilingual aspects of the task. Indeed, current Web People Search engines (Spock, Zoominfo, etc.) are not yet able to identify that two

[1] Available in http://nlp.uned.es/weps/weps-1-data/ and http://nlp.uned.es/weps/weps-2-data/

document clusters actually refer to the same person whenever they are expressed in different languages. Another interesting aspect is to address the relations between clustering and attribute extraction in an integrated task. This should provide new insights into the challenges faced by WePS systems. Finally, WePS participants have also suggested to have a track where participants share the same document representations (feature vectors), and therefore focus exclusively on how to use that representation to provide an efficient clustering.

Let us conclude by observing that searching people in the World Wide Web is one of the many tasks which were not adequately supported by standard search engines when we started this research, back in 2004. Since then, things have quickly evolved in the Web: people enter personal information in different ways (for instance, having a Facebook profile is now much more common than having a web home page or a blog), and search engines integrate a higher amount of information (not just web pages) in a much more sophisticated manner. We see our research effort as a modest contribution in a massive effort to turn Web search facilities into powerful text mining engines. These mining engines will be able to filter, classify and synthesise information in different ways according to different information needs. We hope that our research represents a step – even if small – into one of the biggest challenges faced by this type of systems: facing name ambiguity as an essential step towards effective information fusion.

Bibliography

[AAG09] Javier Artiles, Enrique Amigó, and Julio Gonzalo. The role of named entities in Web People Search. In *EMNLP '09: Proceedings of the ACL-09 conference on Empirical methods in Natural Language Processing*. Association for Computational Linguistics, 2009.

[AE06] Eneko Agirre and Philip Edmonds, editors. *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[AGA09a] Enrique Amigó, Julio Gonzalo, and Javier Artiles. Combining Evaluation Metrics via the Unanimous Improvement Ratio and its Application in WePS Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[AGA09b] Javier Artiles, Julio Gonzalo, and Enrique Amigó. The impact of query refinement in the Web People Search Task. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009.

[AGAV08] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Journal of Information Retrieval*, 2008.

[AGS07] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Association for Computational Linguistics, 2007.

[AGS09] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[AGV05] Javier Artiles, Julio Gonzalo, and Felisa Verdejo. A testbed for people searching strategies in the WWW. In *SIGIR*, 2005.

[AKE04] Reema Al-Kamha and David Embley. Grouping search-engine returned citations for person-name queries. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*. ACM Press, 2004.

[ASG08] Javier Artiles, Satoshi Sekine, and Julio Gonzalo. Web People Search: results of the first evaluation and the plan for the second. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1071–1072, New York, NY, USA, 2008. ACM.

[BAdR07] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. UVA: Language Modeling Techniques for Web People Search. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. ACL, 2007.

[Bal08] Krisztian Balog. *People Search in the Enterprise*. PhD thesis, University of Amsterdam, June 2008.

[BB98a] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, 1998.

[BB98b] Amit Bagga and Breck Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 79–85, 1998.

[BF08] Alex Baron and Marjorie Freedman. Who is Who and What is What: Experiments in Cross-Document Co-Reference. In *EMNLP*, pages 274–283. Association for Computational Linguistics, 2008.

[BHH$^+$09] Krisztian Balog, Jiyin He, Katja Hofmann, Valentin Jijkoun, Christof Monz, Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. The University of Amsterdam at WePS2. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[BHK02] J. Bakus, M. F. Hussin, and M. Kamel. A SOM-Based Document Clustering Using Phrases. In *Proceedings of the 9th International Conference on Neural Information Procesing (ICONIP'02)*, 2002.

[Blu05] Matthias Blume. Automatic Entity Disambiguation: Benefits to NER, Relation Extraction, Link Analysis, and Inference. In *International Conference on Intelligence Analysis*, 2005.

[BM05] Ron Bekkerman and Andrew McCallum. Disambiguating Web Appearances of People in a Social Network. In *Proceedings of the 14th international conference on World Wide Web*, 2005.

[BMI06] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Extracting Key Phrases to Disambiguate Personal Name Queries in Web Search. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval? at ACL'06*, 2006.

[CdVS05] Nick Craswell, Arjen de Vries, and Ian Soboroff. Overview of the TREC-2005 Enterprise Track. In *TREC 2005 Conference Notebook*, pages 199–205, 2005.

[CLH09] Ying Chen, Sophia Yat Mei Lee, and Chu-Ren Huang. PolyUHK: A Robust Information Extraction System for Web Personal Names. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[CM07a] Ying Chen and James Martin. Towards Robust Unsupervised Personal Disambiguation. In *EMNLP-CoNLL-2007*, 2007.

[CM07b] Ying Chen and James H. Martin. CU-COMSEM: Exploring Rich Features for Unsupervised Web Personal Name Disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. ACL, 2007.

[CMP08] Ying Chen, James Martin, and Martha Palmer. Robust Disambiguation of Web-Based Personal Names. In *ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 276–283, Washington, DC, USA, 2008. IEEE Computer Society.

[CPGV05] Juan M. Cigarrán, Anselmo Peñas, Julio Gonzalo, and Felisa Verdejo. Evaluating Hierarchical Clustering of Search Results. In *SPIRE*, pages 49–54, 2005.

[Cuc07] Silviu Cucerzan. Large Scale Named Entity Disambiguation Based on Wikipedia Data. In *The EMNLP-CoNLL-2007*, 2007.

[Dom01] Byron Dom. An Information-Theoretic External Cluster-Validity Measure. IBM Research Report, 2001.

[dVAdPSVD07] David del Valle-Agudo, César de Pablo-Sánchez, and María Teresa Vicente-Díez. UC3M-13: Disambiguation of Person Names Based on the Composition of Simple Bags of Typed Terms. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. ACL, 2007.

[EE07] Jeremy Ellman and Gary Emery. UNN-WePS: Web Person Search using co-Present Names and Lexical Chains. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[ETY$^+$07] Ergin Elmacioglu, Yee Fan Tan, Su Yan, Min-Yen Kan, and Dongwon Lee. PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[FH04] Michael Ben Fleischman and Eduard Hovy. Multi-Document Person Name Resolution. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004.

[GA04] Chung Heong Gooi and James Allan. Cross-Document Coreference on a Large Scale Corpus. In *HLT-NAACL*, 2004.

[GCY92] William A. Gale, Kenneth W. Church, and David Yarowsky. Work on statistical methods for Word Sense Disambiguation. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60, 1992.

[Gho03] J. Ghosh. Scalable Clustering Methods for Data Mining. In Nong Ye, editor, *Handbook of Data Mining*. Lawrence Erlbaum, 2003.

[GMV$^+$09] José Carlos González, Pablo Maté, Laura Vadillo, Rocío Sotomayor, and Álvaro Carrera. Learning by doing: A baseline approach to the clustering of web people search results. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[GO09] Jun Gong and Douglas Oard. Determine the Entity Number in Hierarchical Clustering for Web Personal Name Disambiguation. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[Gri94] Ralph Grishman. Whither written language evaluation? In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 120–125, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[Guh04] Guha, R. and Garg, A. Disambiguating people in search. Technical report, Stanford University, 2004.

[HBV01] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[HN07] Andrea Heyl and Günter Neumann. DFKI2: An Information Extraction Based Approach to People Disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[HP96] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84. ACM Press, 1996.

[HZ09] Xianpei Han and Jun Zhao. CASIANED: Web Personal Name Disambiguation Based on Professional Categorization. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[HZG05] Hui Han, Hongyuan Zha, and C. Lee Giles. Name disambiguation in author citations using a K-way spectral clustering method. pages 334–343, June 2005.

[IOS$^+$09] Masaki Ikeda, Shingo Ono, Issei Sato, Minoru Yoshida, and Hiroshi Nakagawa. Person Name Disambiguation on the Web by TwoStage Clustering. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[IXZ07] José Iria, Lei Xia, and Ziqi Zhang. WIT: Web People Search Disambiguation using Random Walks. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[KB07] Paul Kalmar and Matthias Blume. FICO: Web Person Disambiguation Via Weighted Similarity of Entity Contexts. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[KCN$^+$07] Dmitri V. Kalashnikov, Stella Chen, Rabia Nuray, Sharad Mehrotra, and Naveen Ashish. Disambiguation Algorithm for People Search on the Web. In *Proc. of IEEE International Conference on Data Engineering (IEEE ICDE)*, 2007.

[KF09] Paul Kalmar and Dayne Freitag. Features for Web Person Disambiguation. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[KNL$^+$09] I.S. Kang, S.H. Na, S. Lee, H. Jung, P. Kim, W.K. Sung, and J.H. Lee. On co-authorship for author disambiguation. *Information Processing and Management*, 45(1):84–97, 2009.

[KNTM08] Dmitri V. Kalashnikov, Rabia Nuray-Turan, and Sharad Mehrotra. Towards Breaking the Quality Curse. A Web-Querying Approach to Web People Search. In *Proc. of Annual International ACM SIGIR Conference*, 2008.

[LA99] B. Larsen and C. Aone. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[Leu01] Anton Leuski. Evaluating document clustering for Interactive Information Retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 33–40, New York, NY, USA, 2001. ACM.

[LFHDC09] Els Lefever, Timur Fayruzov, Véronique Hoste, and Martine De Cock. Fuzzy Ants Clustering for Web People Search. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[LHF07] Els Lefever, Véronique Hoste, and Timur Fayruzov. AUG: A combined classification and clustering approach for web people disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[LMR05] X. Li, P. Morie, and D. Roth. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*, 26(1):45–58, 2005.

[LZL$^+$09] Man Lan, Yu Zhe Zhang, Yue Lu, Jian Su, and Chew Lim Tan. Which who are they? people attribute extraction and disambiguation in web search results. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[Mal05] Bradley Malin. Unsupervised name disambiguation via social network similarity. In *Workshop on Link Analysis, Counterterrorism, and Security*, 2005.

[Man06] Gideon Mann. *Multi-Document Statistical Fact Extraction and Fusion*. PhD thesis, Johns Hopkins University, 2006.

[Mei03] Marina Meila. Comparing clusterings. In *Proceedings of COLT 03*, 2003.

[MRA09] Juan Martínez-Romo and Lourdes Araujo. Web People Search Disambiguation using Language Model Techniques. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[MS99] Christopher D. Manning and Hinrich Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999.

[MY03] Gideon S. Mann and David Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL) at HLT-NAACL*. Association for Computational Linguistics, 2003.

[NC08] Hien T. Nguyen and Tru H. Cao. *Named Entity Disambiguation: A Hybrid Statistical and Rule-Based Incremental Approach*, pages 420–433. Springer, 2008.

[Nei02] D. B. Neill. Fully Automatic Word Sense Induction by Semantic Clustering. Master's thesis, Cambridge University, 2002.

[NLS04] Cheng Niu, Wei Li, and Rohini K. Srihari. Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004.

[NTCKM09] Rabia Nuray-Turan, Zhaoqi Chen, Dmitri Kalashnikov, and Sharad Mehrotra. Exploiting Web querying for Web People Search in WePS2. In *Rabia Nuray-Turan Zhaoqi Chen Dmitri V. Kalashnikov Sharad Mehrotra*, 2009.

[Ped06] Ted Pedersen. Improving Name Discrimination : A Language Salad Approach. In *Proceedings of the EACL 2006 Workshop on Cross-Language Knowledge Induction*, 2006.

[PK06] Ted Pedersen and Anagha Kulkarni. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 276–279, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[PL02a] P. Pantel and D. Lin. Efficiently Clustering Documents with Committees. In *Proceedings of the PRICAI 2002 7th Pacific Rim International Conference on Artificial Intelligence*, pages 18–22, Tokyo, Japan, August 2002.

[PL02b] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619, New York, NY, USA, 2002. ACM.

[PM07] Octavian Popescu and Bernardo Magnini. IRST-BP: Web People Search Using Name Entities. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[PNH06] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Personal Name Resolution Crossover Documents by a Semantics-Based Approach. *IEICE TRANS. INF. & SYST.*, E89-D, 2006.

[PPK05] Ted Pedersen, Amruta Purandare, and Anagha Kulkarni. Name Discrimination by Clustering Similar Contexts. In *CICLing*, 2005.

[PTV$^+$09] David Pinto, Mireya Tovar, Darnes Vilariño, Héctor Díaz, and Héctor Jiménez-Salazar. An Unsupervised Approach based on Fingerprinting to the Web People Search task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[Rap03] Reinhard Rapp. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322, 2003.

[RBGST09] Lorenza Romano, Krisztian Buza, Claudio Giuliano, and Lars Schmidt-Thieme. XMedia: Web People Search by Clustering with Machinely Learned Similarity Measures. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[RGY07] Delip Rao, Nikesh Garera, and David Yarowsky. JHU1 : An Unsupervised Approach to Person Name Disambiguation using Web Snippets. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[RH07] Andrew Rosenberg and Julia Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[Rij74] Van C. Rijsbergen. Foundation of evaluation. *Journal of Documentation*, 30(4):365–373, 1974.

[RK99] Y. Ravin and Z. Kazi. Is Hillary Rodham Clinton the President? Disambiguating names across documents. In *Proceedings of the ACL '99 Workshop on Coreference and its Applications Association for Computational Linguistics*, 1999.

[SA09] Satoshi Sekine and Javier Artiles. WePS2 Attribute Extraction Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[Sag07] Horacio Saggion. SHEF: Semantic Tagging and Summarization Techniques Applied to Cross-document Coreference. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[Sag08] Horacio Saggion. Experiments on Semantic-based Clustering for Cross-document Coreference. In *International Joint Conference on Natural language Processing*, 2008.

[Sch92] Hinrich Schutze. Context space. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60, 1992.

[SDCCK07] Steven Schockaert, Martine De Cock, Chris Cornelis, and Etienne E. Kerre. Clustering web search results using fuzzy ants: Research Articles. *Int. J. Intell. Syst.*, 22(5):455–474, 2007.

[Sek08] Satoshi Sekine. Extended Named Entity Ontology with Attribute Information. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.

[SJP04] Amanda Spink, Bernard Jansen, and Jan Pedersen. Searching for people on Web search engines. *Journal of Documentation*, 60:266 – 278, 2004.

[SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[SO07] Kazunari Sugiyama and Manabu Okumura. TITPI: Web People Search Task Using Semi-Supervised Clustering Approach. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*. ACL, 2007.

[Tiw05] Charu Tiwari. Entity Identification On The Web. Master's thesis, School Of Information Technology, Indian Institute of Technology, Bombay, 2005.

[TKL06] Y.F. Tan, M.Y. Kan, and D. Lee. Search engine driven author disambiguation. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 314–315. ACM New York, NY, USA, 2006.

[TWH01] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society (Series B)*, page 411–423, 2001.

[TZZ+07] Jie Tang, Jing Zhang, Duo Zhang, Limin Yao, Chunlin Zhu, and Juan-Zi Li. ArnetMiner: An Expertise Oriented Search System for Web Community. In *Semantic Web Challenge*, 2007.

[VBA+95] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[Ven09] Priya Venkateshan. Clustering Web People Search Results Using Fuzzy Ant- Based Clustering. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[WGLD05] Xiaojun Wan, Jianfeng Gao, Mu Li, and Binggong Ding. Person resolution in person search results: WebHawk. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM Press, 2005.

[WL02] D. Winchester and M. Lee. Using Proper Names to Cluster Documents. Technical report, Acquiring (and Using) Linguistic (and World) Knowledge for Information Access: Papers from the spring Symposium (Technical Report SS-02-09), 2002.

[WRC97] Nina Wacholder, Yael Ravin, and Misook Choi. Disambiguation of Proper Names in Text. In *ANLP*, pages 202–208, 1997.

[XLG03] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM Press, 2003.

[ZK01] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report TR 01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001.

[ZK02] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, New York, NY, USA, 2002. ACM.

Appendix A

Publications

The publications that have resulted from this thesis and their related chapters are listed below:

Chapter 3

- Javier Artiles, Julio Gonzalo, and Felisa Verdejo. A testbed for people searching strategies in the WWW. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2005.
- Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval 2007 WePS evaluation: Establishing a benchmark for the Web People Search Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. ACL, 2007.
- Javier Artiles, Satoshi Sekine, and Julio Gonzalo. Web People Search: results of the first evaluation and the plan for the second. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1071–1072, New York, NY, USA, 2008.

Chapter 4

- Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 2008.

Chapter 5

- Enrique Amigó Julio Gonzalo, and Javier Artiles. Combining Evaluation Metrics via the Unanimous Improvement Ratio and its Application in WePS Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

Chapter 6

- Javier Artiles, Julio Gonzalo, and Satoshi Sekine. WePS 2 Evaluation Campaign: Overview of the Web People Search clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

- Satoshi Sekine and Javier Artiles. WePS2 Attribute Extraction Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

- Javier Artiles, Julio Gonzalo, and Satoshi Sekine (Eds). WePS-2: Proceedings of the Workshop of the Second Web People Search Evaluation Campaign. Madrid, Spain. 2009.

Chapter 7

- Javier Artiles, Julio Gonzalo, and Enrique Amigó. The impact of query refinement in the Web People Search Task. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*, 2009.

Chapter 8

- Javier Artiles, Enrique Amigó, and Julio Gonzalo. The role of named entities in Web People Search. In *EMNLP '09: Proceedings of the ACL-09 conference on Empirical methods in Natural Language Processing*. Association for Computational Linguistics, 2009.