Semantic Network Analysis for
Unsupervised Topic Linking and Labelling

Ioana Hulpuş

SUPervisor:
Dr. Conor Hayes

INTERNAL EXAMINER:
Dr. Paul Buitelaar

EXTERNAL EXAMINER:
Dr. Harith Alani

Dissertation submitted in pursuance of the degree of Doctor of Philosophy

April 2014
The studies presented in this thesis were performed at the National University of Ireland, Galway (NUI Galway). The research was financially supported by the Science Foundation Ireland (SFI) under grant No. SFI/08/CE/11380 (Lion-2).

©2014 Ioana Hulpuş. All rights reserved.
Abstract

We rely more and more on machines to organise, analyse and summarise the vast amount of textual digital information that is being produced at a rate never seen before. At the same time, we notice an increase in availability of structured knowledge that is understandable by both humans and machines. The integration between unstructured text and structured knowledge is crucial for availing of the knowledge contained in text. The research questions that we tackle in this thesis are essential for understanding how applications can effectively link text elements to external background knowledge, and how this background knowledge can assist humans in the interpretation of vast text collections.

Towards this goal, this thesis deals primarily with two core problems: word-sense disambiguation and topic labelling. Word-sense disambiguation is a fundamental problem that needs to be dealt with by most systems that need to integrate text and background knowledge. In this thesis, we investigate two scenarios for word-sense disambiguation. The first scenario focuses on disambiguation with multiple sense inventories simultaneously, and has not been addressed before. We tackle this problem by proposing a versatile disambiguation approach that only requires a short textual definition of word senses. The second scenario addresses word-sense disambiguation with a pre-given semantic graph, DBpedia. We propose a new disambiguation algorithm that solely relies on graph proximity for solving this problem. The novelty lies in that no previous work took a semantic graph approach to disambiguation with DBpedia.

The second core problem this thesis tackles is topic labelling. Topic labelling is necessary for displaying text mining results in a human interpretable way. Broadly, its goal is to find a phrase that captures the essence (gist) behind a group of related words (topic). Our approach exploits the structure of the semantic graph of DBpedia in order to solve this problem.

The unifying high-level hypothesis behind our research is that structural properties of concepts reveal their semantic properties. All our findings show a substantial correspondence between distributional semantics and semantics captured in the structure of semantic networks. This opens new opportunities for integrating the knowledge extracted from text through text mining and background knowledge, as well as for leveraging the benefits of this integration. Throughout this thesis we evaluate our proposed methods through user studies, compare their performances to related work and discuss our findings.
Declaration

I declare that this thesis is composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Ioana Hulpuș,

12 May 2014
Acknowledgements

First of all, I would like to thank Dr. Conor Hayes who made this thesis possible through his supervision, trust and unfailing support. He has been a great model for me, and I feel extremely fortunate to have had the chance of learning from him, and to have benefited from his invaluable guidance in my research.

Great thanks to Marcel, who played multiple roles in my life during my studies: advisor and co-author, friend and above all life partner. His constant support, trust, and never-ending patience and understanding, have been invaluable and life-saving at times.

My gratitude extends to Dr. Derek Greene for all the insights he has shared at various times during my work. I also want to show my appreciation to Dr. Manuel Fradinho, who introduced me to research, trusted me despite my lack of experience, and encouraged me to pursue this PhD.

Special thanks and utmost gratitude to my parents, the absolute perfect parents, who always trusted me, and had faith that I can achieve anything. Their immense support despite the physical distance throughout these years has been a constant help, a constant source of love, energy and inspiration. Tons of thanks to my sister Adela, who always trusted in me, always encouraged me and has always inspired me. She perfectly understood every single phase I went through, she has been my great friend through all the struggle in both professional and personal life.

Massive gratitude to Geta my dear friend for all the volumes of tea, chats and NLP insights; also to all my other friends and colleagues in former Deri and current Insight, especially Benjamin, Vasek and Samantha, who have walked the same path with me; then, all my friends from my life outside PhD. A special category here are my Capoeira comrades and instructors, who have provided me with the best possible after work time, the so much needed world where professional achievement does not matter.

I want to also thank the Insight (former Deri) administrative team for all the help and professionalism. And of course, I can only feel huge gratitude towards Ireland and the Irish taxpayer for adopting me and treating me so well all this time. Their help came through SFI who created the right environment for this thesis, funding me and my research through the LION-2 project.
Familiei mele

(To my family)
Contents

List of figures xiii
List of tables xvii

1. Introduction 3
   1.1. Motivation ........................................ 4
   1.2. Research Questions ................................ 7
   1.3. Contributions ...................................... 9
      1.3.1. Word Sense Disambiguation with Multiple Sense Inventories . 9
      1.3.2. Word Sense Disambiguation with a Graph-Structured Knowledge Base 10
      1.3.3. Topic Labelling ................................ 11
   1.4. Thesis Outline .................................... 12

I. Background 15

2. Foundations 17
   2.1. Basic Notions ...................................... 17
      2.1.1. Notions of Word-Sense Disambiguation .................. 17
      2.1.2. Basic Notions of Text Document Representation .............. 19
      2.1.3. Basic Notions of Graphs ................................ 21
   2.2. Semantic Networks .................................. 22
   2.3. Probabilistic Topic Models ............................ 24
   2.4. Semantic Web and Linked Data ........................ 26
      2.4.1. Semantic-Web Notions ............................... 27
      2.4.2. Linked Data ....................................... 30
      2.4.3. Semantic Web and Topic Models ....................... 31
   2.5. Knowledge Bases .................................... 32
      2.5.1. DBpedia .......................................... 32
2.5.2.  WordNet ......................................................... 33

3.  Related Work .................................................. 35

3.1.  Related Work on Unsupervised Word Sense Disambiguation ................. 35
  3.1.1.  Word Sense Disambiguation - The Big Picture ............................ 36
  3.1.2.  Gloss-based WSD ........................................... 43
  3.1.3.  WSD with Semantic Networks ....................................... 45
  3.1.4.  Word-Sense Disambiguation with DBpedia ............................... 48

3.2.  Related Work on Topic Labelling .................................... 50
  3.2.1.  Label Extraction from Text ....................................... 50
  3.2.2.  Keyphrase Extraction from Text with Support from External Knowledge Bases .................................................. 52
  3.2.3.  Label Finding in External Knowledge Bases ............................. 53

II.  Core .............................................................. 55

  4.1.  Introduction ..................................................... 57
    4.1.1.  The Joint WSD Problem ...................................... 59
    4.1.2.  Approach Overview ......................................... 60
  4.2.  Eigenvalue-Based Word Sense Disambiguation ................................ 61
    4.2.1.  The HITS Algorithm ........................................... 61
    4.2.2.  Eigenvalue-based WSD Solution Coherence ......................... 63
    4.2.3.  Theoretical Comparison to Sum of Overlaps ....................... 63
  4.3.  Adaptive Sense Modelling for WSD .................................. 65
    4.3.1.  Adaptive Sense Vectors (ASV) ................................ 65
    4.3.2.  Using ASV with Heuristics .................................... 68
  4.4.  Evaluation ...................................................... 72
    4.4.1.  Datasets ...................................................... 73
    4.4.2.  User Study .................................................... 74
    4.4.3.  Compared Methods ........................................... 75
    4.4.4.  Tests and Performance Measures .................................. 77
    4.4.5.  Results and Discussion ....................................... 80
  4.5.  Conclusions ................................................... 85
5. Proximity Measures on the DBpedia Graph for WSD

5.1. Introduction .................................................. 87
   5.1.1. Contributions ............................................. 90
5.2. Approach Overview ......................................... 90
   5.2.1. Toy Example ............................................. 91
   5.2.2. The Blueprint of Sen-Dis ............................. 92
   5.2.3. Sen-Dis Process Overview ............................ 94
5.3. Sense Ego-Networks ....................................... 94
   5.3.1. Preliminary Check on Sense Ego-Network Overlap .... 96
5.4. Word Clusters as Disambiguation Context ................. 97
   5.4.1. Employed Clustering Methods ......................... 98
5.5. Relatedness measures ..................................... 99
   5.5.1. Inverse Shortest Path [ISP] .......................... 100
   5.5.2. Multiple paths based measures ....................... 100
   5.5.3. Personalised PageRank Cosine Similarity [PPR] .... 103
   5.5.4. Evaluation of the Relatedness Measures .............. 103
5.6. Global Optimisation for Joint Word Sense Disambiguation 114
   5.6.1. Frozen-past wrapper function ......................... 114
   5.6.2. B&B for Joint WSD ..................................... 116
   5.6.3. Selecting a Preliminary Solution with Simulated Annealing 120
5.7. Sen-Dis: The Semantic Network based WSD Process ........ 120
5.8. Evaluation and Results ................................... 121
   5.8.1. E-WSD for Documents ................................. 122
   5.8.2. Evaluated Methods and Settings ..................... 122
   5.8.3. Ground Truth Datasets ............................... 123
   5.8.4. Performance Measures ............................... 124
   5.8.5. Results ............................................... 126
5.9. Conclusion .................................................. 131

6. Using the DBpedia Graph for Unsupervised Topic Labelling

6.1. Introduction .................................................. 135
6.2. Overview ................................................... 137
   6.2.1. Motivating Example ................................... 137
   6.2.2. Problem Statement .................................... 138
   6.2.3. Approach Overview .................................... 139
6.3. Topic Graph Extraction .................................... 139
6.4. Topic Labelling with Graph Centrality Measures . . . . . . . . . . . . . . . . 141
  6.4.1. Focused Closeness Centrality . . . . . . . . . . . . . . . . . . . . . 143
  6.4.2. Focused Information Centrality . . . . . . . . . . . . . . . . . . . . 144
  6.4.3. Focused Betweenness Centrality . . . . . . . . . . . . . . . . . . . . 145
  6.4.4. Focused Random Walk Betweenness Centrality . . . . . . . . . . . . . 146
6.5. Kanopy: Topic and Document Labelling on top of WSD . . . . . . . . . . . . 149
  6.5.1. From Topic Labelling to Document Labelling . . . . . . . . . . . . . 149
6.6. Labelling Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
  6.6.1. Experiments and Evaluation of Topic Labelling . . . . . . . . . . . . 151
  6.6.2. Experiments and Evaluation of Document Labelling . . . . . . . . . 161
6.7. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 178

III. Conclusion 181

7. Conclusion 183
  7.1. Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
      7.1.1. ASV and E-WSD . . . . . . . . . . . . . . . . . . . . . . . . . . . 185
      7.1.2. Sen-Dis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
      7.1.3. Topic Labelling with Focused Centralities . . . . . . . . . . . . . 187
      7.1.4. Topics in Semantic Networks . . . . . . . . . . . . . . . . . . . . 187
  7.2. Directions for Future Research . . . . . . . . . . . . . . . . . . . . . . . . 188

A. Branch and Bound for Word Sense Disambiguation 191
  A.1. Formalisation of the objective function . . . . . . . . . . . . . . . . . . . 191
  A.2. The Bounding Function . . . . . . . . . . . . . . . . . . . . . . . . . . 192
  A.3. Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193

B. DBpedia Concept Index 197

C. DBpedia Candidate Sense pre-filter 199

D. List of DBpedia and Yago classes containing named entities 201
  D.1. DBpedia . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 201
  D.2. YAGO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 203

E. Demonstration 205

Bibliography 211
# List of figures

1.1. The Knowledge Discovery in Databases (KDD) process [38] .................................................. 5

1.2. Kanopy document processing pipeline ................................................................. 12

1.3. Proposed document processing pipeline and thesis contributions outline ........ 13

2.1. Semantic networks ...................................................................................................... 23

2.2. Example of four topics from [126] ........................................................................ 25

2.3. The Semantic Web architecture [20] ................................................................. 28

3.1. Conceptual framework of word-sense disambiguation ........................................ 37

4.1. Bipartite graph relating concepts to their common gloss-words ......................... 61

4.2. Examples of bipartite graphs with their corresponding adjacency and hubs matrices, as well as the sum of pairwise scalar products and the principal eigenvalue, $\lambda_1$ .................................................................................................................. 64

4.3. Top two feasible solutions of the WSD problem in Table 4.1 ......................... 66

4.4. Bipartite graphs generated by each combination of meanings in the toy disambiguation example in Table 4.1, with the corresponding eigenvalue scores ($\lambda$). ................................................................. 68

4.5. A set of four examples illustrating how sense weights influence the weights of graph edges. ................................................................................................................. 70

4.6. The user interface of the web application used to collect human sense annotations 75

4.7. Results on the computationally ambiguous words; ............................................. 81
4.8. Box-plot representation of the distributions of senses over gloss lengths 83
4.9. The proportion of WordNet senses out of all retrieved senses 84

5.1. Example of semantic networks containing three target concepts, and the resulted cliques; 92
5.2. Three concepts $A_1$, $B_1$ and $C_1$ with their corresponding ego-graphs. When used in the same feasible solution, the three ego-networks merge and form the feasible solutions’s ego-network. 96
5.3. The weights of the paths as function of the path length; Katz@0.5 represents the weight of the path as scored by $\text{rel}^{\text{Katz}}_K$ with $\alpha = 0.5$. 107
5.4. Merging of resource nodes to category nodes that correspond to the same concept. C:X stands for Category:X; Nodes $D$ and $G$ are example seed nodes. Nodes $A$ and $G$ in the graph on the right are the nodes obtained after merging $A$ and $C : A$, and $G$ and $C : G$ from the graph on the left; 109
5.5. The frozen past wrapper routine for disambiguation 116
5.6. Disambiguation process with Sen-Dis 121
5.7. E-WSD for documents, with DBpedia 123
5.8. Precision vs Recall on the NYT10 dataset 127
5.9. Number of noun phrases per document extracted by Sen-Dis and annotated by humans, on the NYT10 dataset 129

6.1. The topic labelling process 139
6.2. Example of sense ego-graphs, topic graph and core topic graph; 141
6.3. Node centralities in the ego-network of topic with DBpedia concepts: [Government, Economy, Policy, Money, Unemployment, Income, Inflation, Interest, Interest rate, Demand (economics), Fiscal policy, Output]. The seed nodes are black, and the size of the nodes increases linearly with their centrality illustrated in each graph. For comparison, the nodes in the four graphs change only their sizes and not their positions. 147
6.4. Kanopy: Topic Labelling on top of WSD with E-WSD or Sen-Dis 149
6.5. Document labelling with Kanopy .............................................. 150
6.6. Histogram of proportion of core concepts in the topic graph; ............... 154
6.7. Example shown in the Instructions section of the user interface of the user-study web application ................................................................. 155
6.8. UI of the user study web application ................................................. 155
6.9. Precision and Coverage (y axis) @top-k (x axis) for combined corpora. . . 156
6.10. Precision and Coverage (y axis) @top-k (x axis) for the three corpora. . . 157
6.11. Influence of number of seed nodes .................................................. 159
6.12. Convergence to the set of top-5 labels as the number of core concepts increases; x-axis: top-k core concepts of the topic; y-axis - the proportion of labels that are also in the final top-5 .......................................................... 161
6.13. Document labelling process, and the three potential sources of labels: (1) the document words (or topic words), (2) their DBpedia concepts counterpart obtained through linking and disambiguation (seeds), and (3) the DBpedia concepts from the seeds’ ego-networks; ................................................................. 162
6.14. Labelling purity scores per corpus; The shaded area illustrates the target interval; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k ................................................................. 168
6.15. The ground truth label frequency on the StackExchange document clusters; The shaded areas illustrate the target intervals; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k ................................................................. 170
6.16. The ground truth label frequency on the BBC document clusters; The shaded areas illustrate the target intervals; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k ................................................................. 172
6.17. Purity scores on BAWE clusters with positive gain; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k ; the shaded areas illustrate the target interval ................................................................. 174
6.18. Purity scores of the labelling methods, on BAWE clusters where they did not obtain positive gain; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k ; the shaded areas illustrate the target interval . . 175
A.1. Example of B&B search space ........................................ 194

A.2. Example of relatedness graph $G^R$ containing all senses and the edges between them ............................................................ 194

B.1. Index Entry for DBpedia resource: http://dbpedia.org/resource/International ........................................................................ 197

B.2. Index Entry for DBpedia resource: http://dbpedia.org/resource/International_(disambiguation) ........................................ 198

C.1. Example of paths connecting disambiguation and redirect DBpedia pages . . 200

E.1. UI of Kanopy Demo web application demonstrating document labelling . . 205

E.2. UI of Kanopy Demo web application demonstrating the possible exploration of the topic graphs and relation between concepts and labels .................. 206
## List of tables

2.1. Example namespaces and their usual prefixes. *DBpedia is a knowledge base, not an ontology* ................................................................. 29

4.1. A toy example for three target words, with a set of candidate meanings and their respective related words. ........................................................... 66

4.2. The adapted vectors-space representations of sense web#1 from the example in Table 4.1 .................................................................................. 67

4.3. A toy example for three target words, with a set of candidate meanings and their respective related words. The NE tag means the corresponding word is a named entity ................................................................. 70

4.4. Two feasible solutions containing senses g#2 and s#2. Note that the vector representations of these two senses strongly depend on the feasible solution they are part of, in this case due to being combined with the senses w#1 and SB#1 respectively ................................................................. 71

4.5. Algorithm performance assessment scheme ................................................. 78

4.6. Average number of correct senses per word as annotated by humans, and as retrieved by each method ................................................................ 79

4.7. Number of words with ground-truth annotations for the various settings; When Fuzzy Filter in ON, the number of words are approximate (≈) as the removal of fuzzy senses also causes removal of some words, depending on the algorithm. The reported values are the medians over the algorithms. ....................................................... 80

5.1. Connected pairs after DBpedia subgraph extraction ..................................... 110
5.2. Spearman Correlations between system assessment and human assessment in the four ground truth datasets ................................................. 111

5.3. Number of clusters per document and words per cluster for the two clustering algorithms .................................................. 126

5.4. Annotation test results for NYT10 dataset, using only the hierarchical links; P - precision; R - recall; F - F-measure ................................................. 127

5.5. Accuracy results for the disambiguation test on the two datasets, .................................................. 128


6.2. Correlation of the focused centrality measures .................................................. 151

6.3. Ground truth labels for document clusters .................................................. 165

6.4. Top-10 most frequent labels in WordPress cluster of StackExchange ............... 171

6.5. Top-10 most frequent labels in Gaming cluster of StackExchange ............... 171

6.6. Top-10 most frequent labels in Technology cluster of BBC ....................... 173

6.7. Top-10 most frequent labels in Sport cluster of BBC ............................. 173

A.1. Example values for pairwise relatedness between the senses of four words. . . 195

A.2. The sorted lists of pairwise relatedness .................................................. 195

A.3. The B&B WSD algorithm applied to example data ..................................... 196
Chapter 1.

Introduction

The main interest behind the research presented in this thesis is to understand the correspondence between the structure of semantic graphs and the meaning expressed by topics extracted through statistical text analysis. For example, do co-occurring words have meanings that are likewise related in semantic graphs? If so, can we identify the meanings of co-occurring words, by their proximity in the semantic graph? Can the semantic graph be used to summarise the essence of the topic in a simple, human interpretable way? By answering these questions, we can understand how applications can most effectively leverage background knowledge for text understanding, collection organising, summarisation and navigation as well as content filtering.

All the previous questions revolve around the connection between words and their meanings. While this connection is seamless for humans, it is a big challenge for computers, mostly due to polysemy. Polysemy is a feature of language whereby words can have different meanings in different contexts. For instance, the word orange can generally refer to the fruit, the colour, or the telecommunications company. Humans encounter no problem in grasping the contextual meaning of the word, but computers must use complex algorithms to correctly identify the intended sense, by interpreting the context in which the word is used. This automatic process is called word-sense disambiguation (WSD). A substantial part of this thesis deals with the
problem of word-sense disambiguation, which we study from two perspectives:

- First, we research a word-sense disambiguation approach that can work with any sense inventory (e.g., dictionary, thesaurus, semantic network, etc.), and also with multiple sense inventories simultaneously. Then we run this algorithm on a graph-structured knowledge base, DBpedia\(^1\), and analyse if the correctly disambiguated meanings of related words are related in the knowledge base;

- Using insights from part 1, we develop a second word-sense disambiguation approach, that is suitable for graph-structured knowledge bases. Its novelty lies in achieving disambiguation by exploiting solely the pre-given graph structure of the knowledge base.

By using a graph-structured external knowledge base, the meanings of words can be seen as nodes in the knowledge base graph. This opens the opportunity for analysing the network that connects the meanings or words known to be related, and that we conventionally refer to as topics. We explore this opportunity by defining graph measures that are suitable for finding a coherent phrase that captures the essence behind a topic. This problem is called topic labelling. Usually, topics are the result of automatic statistical analysis of text. The purpose of topic labelling is to make topics easily interpretable for humans. Solutions to this problem face the challenge that text in natural language assumes certain background knowledge of the message’s recipient. For example, if we consider the topic [patient, drug, hospital, health, professional], a document that contains all these words might not mention the term healthcare, which is a possible label for the topic. Linking topics to external knowledge bases can fill this knowledge gap by revealing the concepts that are best for labelling the main themes of a document or a set of documents.

1.1. Motivation

We are witnessing an intensive proliferation of digitised documents in all aspects of our lives. Textual content in digital form is being produced constantly at a rate never seen before, in academia, social media, digital books, news, etc. As our ability to store and share these amounts of data is increasing, its value is limited by our ability to analyse it and draw meaning out of it. There has been substantial progress in summarising text collections by capturing the latent themes using techniques such as probabilistic topic modelling. In order to fully avail of these powerful tools, we must also focus on how their results can be transmitted to humans so that

\(^1\)http://dbpedia.org
they are easily interpreted. This is a fundamental requirement for the knowledge discovery process as defined by Fayyad et al. [38], and shown in Figure 1.1. As opposed to the other stages of this process, the Interpretation / Evaluation step has been vastly neglected.

![Figure 1.1.: The Knowledge Discovery in Databases (KDD) process [38]](image)

For instance, while probabilistic topic modelling provides a great structure for collection browsing, the exploitation of this structure is strongly limited by the user-unfriendly way topics are typically displayed. Typically, they are presented to the user by listing the most frequent words of each topic, for example, a Genetics topic extracted from articles from the journal Science might be presented as the list of words [human, genome, dna, genetic, genes, sequence, gene, molecular, sequencing] [16]. Interpretation of the topics places a significant cognitive load on the reader as they must synthesise a higher level abstraction to capture the unifying meaning of each topic. Furthermore, in many scenarios, the user may be forced to align the topics manually to a pre-existing taxonomy of abstractions in their organisation or domain. Topic labelling supports humans in these tasks, and helps them make sense of the main themes of large amounts of text. Moreover, it is crucial for availing of topics in user interfaces of applications like document collection browsing.

Furthermore, when using topic modelling, users often face the so-called model-checking problem. This means that they cannot decide which model best fits their data. By linking topics to semantic knowledge bases, and labelling them, the user can be supported in deciding on the model that is most interpretable, and in line with the usage needs. The same problems exist with the interpretation of other data mining results, for example document clustering. Powerful algorithms have been devised for clustering massive amounts of documents. These clustering algorithms find patterns in the documents and use them to group the documents.
However, humans often struggle to interpret these patterns and to make sense of the results of the clustering process. As we later show, topic labelling also provides means for cluster labelling.

The first step towards the reconciliation between what humans expect, understand and is useful, and what machines identify as prominent patterns is to provide the machines with background knowledge, in a machine understandable format. In this direction, great progress has been achieved by automated knowledge extraction [98], as well as through the development of standard data formats such as RDF\(^2\) which are geared towards making data understandable to machines.

The second step is to bridge the human defined meaning from the background knowledge to the machine inferred meaning. One of the most important challenges here is word-sense disambiguation (WSD). In this thesis, our focus is on unsupervised word-sense disambiguation, which relies on the senses and knowledge provided by the sense inventory\(^3\) to achieve disambiguation.

The two main problems this thesis deals with are therefore unsupervised word-sense disambiguation and topic labelling. More specifically, it addresses a number of research questions that we are summarising in the following section.

\(^2\)Resource Description Framework. [http://www.w3.org/RDF/](http://www.w3.org/RDF/)

\(^3\)we use the term sense inventory to denote any collection of words linked to their meanings, like dictionaries, thesauri, glossaries, semantic networks, knowledge bases
1.2. Research Questions

1. *How can we effectively disambiguate words with multiple sense inventories simultaneously?*

Since knowledge bases are diverse and often complementary, WSD can benefit from the knowledge that can be uncovered by bringing senses from different inventories together. An important prerequisite to disambiguation is a common model for representing the possible senses of words. This is because an assumption behind word-sense disambiguation is that the senses of the words in the disambiguation context are related. While previous research focused on perfecting the WSD accuracy in the context of singular knowledge bases, there is a gap in understanding how senses from different inventories can be compared without introducing selection biases.

2. *Can the graph structure of DBpedia be used as the sole evidence for word sense disambiguation of topic words?*

The most commonly used sense inventory for unsupervised WSD is WordNet, a manually produced lexicon with a clean hierarchical structure. However, it suffers from several limitations, like the lack of information about named entities (e.g., persons, organisations) and a very limited range of relations between concepts. For instance, using WordNet’s relations, the association between hospital and nurse is impossible to capture. Wikipedia\(^4\) has more recently become very popular as a sense inventory due to its very wide topic coverage. Being an encyclopedia, its usage requires the processing of the content of its articles. While approaches that use Wikipedia perform well, they lack flexibility, portability and require heavy preprocessing. On the other side, DBpedia\(^5\) is a graph structured knowledge base, the structured replica of Wikipedia, and has lately attracted a lot of interest for linking to other knowledge bases, both structured and unstructured.

Although most commonly used structured knowledge bases are graph-based, there is a gap in understanding if their underlying graph structure can be used as cue for disambiguation. The network properties of WordNet have been vastly researched for WSD. However, methods employing WordNet rely on its hierarchical nature. While most knowledge bases have a hierarchy of concepts as a backbone, they also contain many other relations that connect these concepts, and are not hierarchical, for instance an authorship relation between a book and its author. These types of non-hierarchical relations give more information about how concepts are

\(^4\)http://wikipedia.org  
\(^5\)http://dbpedia.org
associated and arguably they can be used as evidence for disambiguation. Considering these relations, we can conclude that a graph is the common underlying structure of most structured knowledge bases. As such, we believe that WSD methods that are able to exploit any graph structure are likely to be portable and extendible to different graph topologies.

2. (a) **What graph-measures capture the strength of semantic relatedness and / or semantic similarity between DBpedia concepts?**

A premise of unsupervised word-sense disambiguation is that the senses of related words are themselves related. Therefore, before attempting word-sense disambiguation using the graph-structure of the knowledge base, we need to understand to what extent this graph captures semantic relatedness and / or semantic similarity as assessed by humans. As in most heterogeneous networks, the similarity or relatedness between concepts of DBpedia is not always directly represented. For example, a book concept is connected to the concept corresponding to its author, but it is less likely to be connected to the other books of the same author. As such, the similarity between two books of the same author can only be established by traversing two semantic connections. Furthermore, the same books might also belong to the same genre, in which case the similarity between them is even higher. Regarding relatedness, a book can be seen related to the country whose society it depicts, but also to the home-country of its writer, as well as to the countries where it had high sells. These relations are usually not directly captured, and they must be inferred by analysing the semantic network.

3. **How can we extract topic labels from DBpedia graph in an unsupervised manner?**

The problem of topic labelling has mostly been tackled by previous research under the assumption that the label can be found in the text of documents that the topic was extracted from, and can be uncovered by statistical analysis. As we later show in this thesis, this is not always the case, as often, the label is not mentioned in texts, because the documents often assume a certain background knowledge of the reader. Other related work in this topic uses Wikipedia and relies on analysis of its content and hyperlink structure. This type of approach requires preprocessing of Wikipedia. However, all approaches follow the intuition that the topic label is somehow strongly related to all the topic words. We therefore find it natural to exploit the semantic network that interconnects the topic concepts, and use their common neighbours as labelling candidates. Our intuition is therefore that the suitable labels can be revealed by analysing their graph properties relative to the topic concepts. In this thesis, we search for possible graph measures that can be used to find suitable topic labels.
1.3. Contributions

This thesis brings several contributions that we overview in this section, and address in detail in the following chapters.

1.3.1. Word Sense Disambiguation with Multiple Sense Inventories

The first WSD approach we propose considers the simultaneous use of multiple sense inventories. To this end, we focus on gloss-based WSD, as this type of approach is the only one suitable for multiple sense inventories. The advantage of gloss-based WSD systems lies in that they require little background knowledge (just a gloss per sense), and the algorithms are usually straightforward to implement and suitable for most sense inventories. Their disadvantage is that their performance is often surpassed by approaches that use deeper knowledge. However, we isolated a couple of limitations of related work in gloss-based WSD that we tackle and largely overcome. For example, most gloss-based WSD approaches establish sense relatedness by relying on the mere count of word overlap between sense glosses.

Nevertheless the challenge of computing relatedness between glosses is not trivial. The length of glosses can vary considerably between senses, and most importantly, between senses of different inventories. Some inventories provide very short glosses, of only several words, while others provide glosses of several paragraphs. Traditional text similarity measures tend in this case to either give a great weight to short glosses by length based normalisation, or advantage long glosses. Therefore one challenge is to reduce these biases. Furthermore, since the glosses are text based, they are also subject to noise due to ambiguity and polysemy. We show that by targeting these problems through a novel sense modelling scheme and a novel scoring function for joint disambiguation, we can strongly improve the performance of state-of-the-art gloss-based approaches to disambiguation.

Another novel aspect of this work is that we attempt word-sense disambiguation in the context of probabilistic topic models. Typically, the context for word disambiguation consists of the words in close proximity to the word in question, e.g.: the words from the sentence, paragraph, a window of words of a set size, or a whole document. For generality, we use topics resulted from topic modelling or word clustering as disambiguation context for WSD. We show that preliminary grouping of words based on their co-occurrence generates contexts that are highly suitable for WSD. In some of our experiments, this step improves the disambiguation
accuracy with more than 0.15 as compared to the use of full text, even on short texts (less than 100 words in total).

**Contributions**

- We propose a new vector space model for sense glosses, that addresses the limitations of previous work;
- We propose a new measure for scoring solutions in gloss-based word sense disambiguation;
- We provide a thorough evaluation, including multiple related work approaches, on disambiguation with multiple sense inventories;
- We show that although topic models do not provide any syntax clues, or part of speech tags, they can be used with success for word sense disambiguation.

### 1.3.2. Word Sense Disambiguation with a Graph-Structured Knowledge Base

The second WSD approach we study considers disambiguation with one graph-structured knowledge base, DBpedia. Since there is no previous evidence that the DBpedia graph can be used for disambiguation, this thesis brings several contributions in this direction. We first show that the correct senses of related words lie in close proximity in the DBpedia graph. Given a topic that requires linking to DBpedia, we define a *disambiguation solution*, as a possible combination of senses of the topic words, one sense per word. We then use pairwise graph proximity measures to rank the alternative disambiguation solutions. Our hypothesis is that the solution with highest aggregated proximity provides the correct word senses. We validate this hypothesis by experimenting with several graph proximity measures that have not been used before for joint WSD. Another challenge is related to the size of the knowledge base and noise. We handle the former by restricting the scope of a sense to a close neighbourhood. As for the latter, in this work we introduce the notion of *stop-URIs* to denote concepts whose scope is limited to the knowledge base they are part of (e.g., *Philosophy maintenance categories*) and have no meaning outside it. In this thesis, we highlight and quantify the negative influence that these stop-URIs have on assessing semantic relatedness between concepts with DBpedia’s graph structure, and consequently on word sense disambiguation.
Our experiments show that our method is superior to approaches in the related work that use Wikipedia.

Contributions

• We propose established as well as novel graph-based proximity measures to capture the semantic relatedness between DBpedia concepts, and show that meanings of topic words are closely related in the DBpedia graph structure;

• We propose a new graph-based approach to joint WSD that uses only the underlying graph of DBpedia and the proposed graph-based proximity measures;

• We introduce the concept of stop-URIs and show that their presence in the semantic network renders the structure of the network unusable for satisfactory assessment of semantic relatedness and WSD. As such, throughout our work with the DBpedia semantic network, a crucial step is the removal of the stop-URIs.

1.3.3. Topic Labelling

In this thesis, we tackle the problem of topic labelling by using external knowledge. Since topic labelling attempts to solve an interpretation problem for humans, the decisions that lead to the suggested solution must be tractable and if required, the system must be able to provide simple straightforward explanations for the choices, and ideally be able to receive and integrate feedback from the user.

Therefore, we think that traversal of semantic paths, rather than hyperlink paths, and a structured knowledge-base rather than encyclopedic text, provide a more suitable setting for the topic labelling problem. To this end, we focus on a graph-structured knowledge base, DBpedia, under the assumption that “ideal” label fulfils a certain structural role in the DBpedia graph. We therefore, propose novel focused graph centrality measures, that are able to uncover the concepts that are appropriate labels for topics. Following this process, the topics are not just labelled, but:

• Topics are assigned representative knowledge base subgraphs (topic graph) that contain the relations between their linked concepts;

• Topic labels are concepts in the topic graphs, connected through various relations to the topic concepts.
Furthermore, we show that this methodology gives much better labelling results than methods that look for topic labels within the text of the documents in the collection, and can be used for labelling both topic models and documents.

**Contributions:**

- We introduce the notion of *focused centralities* as adaptations of classic graph centrality measures, in scenarios when the centrality must be computed with respect to a subset of graph nodes;
- We propose a new method for labelling topics and documents that uses the concept of focused centrality.

### 1.4. Thesis Outline

Finally, we introduce the document processing pipeline, Kanopy\(^6\), which ties all our contributions together. The main idea of Kanopy is to use the results of word-sense disambiguation in order to extract the sub-graph of the knowledge base that represents the topic, and use this graph for topic labelling, as illustrated in Figure 1.2.

![Figure 1.2: Kanopy document processing pipeline](http://kanopy.deri.ie)

The reversed arrow between stages 3 and 2 of the pipeline highlights that the semantic graph underlying the knowledge base can be used for disambiguation, as we later show. Due to the topic-centric approach we take throughout this thesis, the document processing pipeline in Figure 1.2, can be applied to any input that can be processed into groups of related words: one single document, a corpus of documents, a single topic, a tag cloud or a sentence. For the last

\(^6\)http://kanopy.deri.ie
three cases, the first stage of the pipeline is usually skipped. Each stage of the Kanopy pipeline includes a deployment of the contributions described. As such, it provides a useful organising principle for the thesis. In Figure 1.3, we show what stages of the pipeline concern each of the core contribution chapters.

This thesis is divided in four parts. The first part, Background contains the research foundations that this work relies on, in Chapter 2. Chapter 3, contains the literature survey on the related work on WSD as well as topic labelling, focusing on the approaches that fall in the same category as ours. The second part, Core contains the three core contribution chapters of this thesis.

Chapter 4 proposes a novel approach to gloss-based disambiguation that can accommodate multiple gloss sources. We evaluate our approach in a comparative evaluation that consists of a user study for gathering the correct senses of words. Chapter 5 presents our approach to word-sense disambiguation when the sense inventory is a graph-based knowledge base. It introduces the graph extraction process, and also discusses and evaluates the influence of topic size on the disambiguation results. Chapter 6 contains the presentation and evaluation of our proposed approach to unsupervised topic labelling. As shown in Figure 1.2, this implies the extraction of the subgraph of interest, and its analysis for the extraction of suitable labels.

We evaluate the quality of the labels in two scenarios: topic labelling and document labelling. In Chapter 6, we also present a demonstration web application that implements all the aspects of the work presented in this thesis.

Finally, Chapter 7 presents the main research directions that this work has opened.
Part I.

Background
Chapter 2.

Foundations

In this chapter, we briefly summarise the fundamental concepts and work which ground the research presented in this thesis. We start by introducing some basic notions in text content modelling, graph-analysis and word-sense disambiguation. Then, we present a general overview of semantic networks and their properties. Then, in Section 2.3 we introduce probabilistic topic models, which are able to capture the main themes of vast document collections. We continue in the fourth section with fundamental concepts of the Semantic Web. Last, in Section 2.5, we present the knowledge bases that we use in this thesis as sources of external semantic knowledge: WordNet, and DBpedia.

2.1. Basic Notions

2.1.1. Notions of Word-Sense Disambiguation

Word-Sense Disambiguation The problem of word-sense disambiguation (WSD) is the problem of automatically identifying which sense of a word is intended in a particular context, for words which independently of the text have multiple senses. Words that have multiple senses are called ambiguous, and the process of word-sense disambiguation assigns them the intended sense. The main requirement for solving word-sense disambiguation is the existence of a sense inventory (e.g, dictionary).

Sense A sense is a possible meaning of a word or phrase, captured in a sense inventory.
**Sense inventory**  A *sense inventory* is any collection of objects that can be represented as a collection of words or phrases that are mapped to their possible meanings, called *senses*. Example: dictionaries, lexicons, knowledge bases, semantic networks.

**Gloss**  A *gloss* is a textual brief definition of the meaning (sense) of a word or phrase.

**Target word / Target phrase**  In a word-sense disambiguation problem, a *target word* is the word that at a particular moment in time is considered for disambiguation. Similarly a *target phrase* is the phrase that at a particular moment is considered for disambiguation.

**Noun phrase**  A *noun phrase* is a word group with a noun or pronoun as its head. In a sentence, a noun phrase functions as subject, object of prepositional object. The simplest noun phrase consists of a single noun.

**Named entity**  A *named entity* is a textual element, usually a noun phrase, that can be classified into pre-defined categories such as names of persons, organisations, locations, quantities, monetary values, expressions of times, etc.

**Context**  In a word-sense disambiguation problem, the *context* is the text in which the target word is considered for disambiguation. The exact definition of the *context* depends on the application, for example: the phrase containing the target word, or other relevant words scattered around the text where the target word is used, selected by various measures of relevance to the word-sense disambiguation problem.

**Semantic network**  A *semantic network* is a network of *concepts*, that models semantic relations between them. In word-sense disambiguation, the concepts from semantic networks can be seen as *senses* of the words that name them, therefore semantic networks are considered a type of sense inventory. Like most networks, semantic networks are typically represented as *graphs*. Section 2.2 provides more details and examples of semantic networks.
**Semantic relatedness**  *Semantic relatedness* denotes a metric defined over two or more senses\(^1\) that quantifies the strength of the semantic relation between them. The notion of semantic relatedness, covers any type of semantic relation.

**Semantic similarity**  *Semantic similarity* denotes a metric defined over two or more senses that quantifies their likeness. Semantic similarity is a type of semantic relatedness and as such it is more specific. For instance the concepts “rake” and “leaves” can be thought of having a much higher semantic relatedness score as compared to their semantic similarity score.

### 2.1.2. Basic Notions of Text Document Representation

**Vector-Space Model**  The *vector-space model* is a model for representing documents as vectors, where each dimension corresponds to a *term*. The terms are defined depending on the application. They can be single words, phrases of at most \(n\) words (n-grams), they can be stemmed, and so on. The value for each dimension corresponds to the *term weight*, a numerical value that generally represents how important the term is with respect to the document. There are many ways of computing term weights, depending on the application, for example term frequency in the document. The number of dimensions (terms) in a vector-space representation of a document usually depends on a predefined vocabulary, and the terms that exist in the vocabulary but not in the document, have *zero* weight. Using vector space models, document \(d\) is represented as

\[
d = [weight_1, weight_2, ..., weight_n]
\]

Where \(n\) is the size of the vocabulary and \(weight_i, i \in \{1, ..., n\}\) is the weight of \(i^{th}\) term in the vocabulary.

**Term Frequency - Inverse Document Frequency (TF-IDF)**  *TF-IDF* is a method for weighting terms of a document, that intends to capture how important they are to the document with respect to the other documents in the corpus. The intuition is that a term should have a high weight in a document if it has high frequency in the document and low frequency in the corpus. This measure is mostly used as a term weighting scheme in vector space models for document representation. The computation consists of multiplying two factors, the *term frequency (TF)* and the *inverse document frequency (IDF)*. For a document \(d\) in corpus \(D\) that

\(^1\)in this work, we only consider relatedness between senses, but related work also considers relatedness between words;
contains \( N \) documents, the \( \text{TF} - \text{IDF} \) score of term \( t \) is typically computed as:

\[
\text{TF} - \text{IDF}(t, d, D) = f(t, d) \times \log \frac{N}{|\{d' \in D | t \in d'\}|}
\]

where \( f(t, d) \) denotes the number of times the term \( t \) occurs in the document \( d \), and can also be normalised, by dividing it to the total number of terms in the document.

**Document Scalar Product** The *scalar product*, also called *dot product* is often used to measure similarity between two documents that are represented in vector-space models. It is a number, computed as the scalar product between the vector representations of the two documents. Having two documents \( d_1 \) and \( d_2 \) represented as vectors over the same dimensions, their scalar product is computed as:

\[
d_1 = [\text{weight}_{1,1}, \text{weight}_{1,2}, ..., \text{weight}_{1,n}]
\]

\[
d_2 = [\text{weight}_{2,1}, \text{weight}_{2,2}, ..., \text{weight}_{2,n}]
\]

\[
d_1 \cdot d_2 = \sum_{i=1}^{n} (\text{weight}_{1,i} \times \text{weight}_{2,i})
\]

**Document Cosine Similarity** The *cosine similarity* of documents is an alternative to the scalar product similarity, that normalises the scalar product with respect to the magnitudes of the document vectors. The idea is that two vectors are more similar if the angle between them is small, checked by a large cosine of the angle. This measure is normalised, belonging to \((0, 1]\). The magnitude of the vector of a document \( d \) is denoted by \(||d||\) and is computed as:

\[
||d_j|| = \sqrt{\sum_{i=1}^{n} \text{weight}_{j,i}^2}
\]

Then, the cosine similarity between document \( d_i \) and \( d_j \) is:

\[
\text{cos}(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i|| \times ||d_j||}
\]
2.1.3. Basic Notions of Graphs

Graph A graph is a representation for networks of objects. The objects are called nodes or vertices and are connected by edges. As such, a graph \( G = (V, E) \) has the set of nodes \( V \) and the set of edges \( E \), where \( n = |V| \) denotes the number of nodes and \( e = |E| \) denotes the number of edges. A typical way of representing graphs is by the adjacency matrix \( A \), of dimension \( n \times n \), whose elements \( A_{ij} \) are equal to 1, if there is an edge in \( E \) between nodes \( i \) and \( j \), and 0 otherwise.

Directed and undirected graphs The edges of a graph can be either directed or undirected, leading to a directed graph or undirected graph. The choice is usually made depending on the objects and relations between them that the graph represents. For example, a telecommunications network might use directed edges to represent calls between two people, with the edge directed from the called to the callee. A social network might be represented with undirected edges when the relations between the actors are considered mutual (i.e., friendship). In this thesis we mainly deal with undirected graphs, therefore in the following we restrict the definitions only to undirected graphs.

Weighted and unweighted graphs A graph is weighted if edges bare weights, that represent the relation between the nodes. For example, for a network of pipes, the weight might represent the distance between nodes. In this case, the graph is represented by the \( W \) matrix, whose elements are real numbers.

Node degree In a graph \( G = (V, E) \) the degree of a node is the number of edges that are adjacent to the node. In directed graphs, the concepts of in-degree and out-degree are used to signify the counts of edges that point to the node and edges that point away from the node.

Bipartite graph A graph \( G = (V, E) \) is a bipartite graph if the vertices can be split in two subsets, \( V_1 \subset V \) and \( V_2 \subset V \), so that all the edges in \( E \) connect only vertices from \( V_1 \) to vertices from \( V_2 \). Therefore, there are no edges in \( E \) between two nodes in \( V_1 \), or two nodes in \( V_2 \).

Subgraph A subgraph of graph \( G = (V, E) \) is a graph \( G' = (V', E') \) such that \( V' \subseteq V \) and \( E' \subseteq E \).
Path  A path through a graph $G = (V, E)$ is a sequence of a subset of edges in $E$ that connect a sequence of nodes in $V$. The nodes on the path are all distinct from one another.

Connected graph  A graph is connected if there is at least one path between all pairs of nodes.

Connected component  A connected component of graph $G = (V, E)$ is a subgraph of $G$, $G_c = (V_c, E_c)$ such that there is at least one path connecting all pairs of nodes in $V_c$, and there is no path connecting any node in $V_c$ to any node in $V \setminus V_c$.

Cliques  A clique of graph $G = (V, E)$ is a subgraph of $G$, $G' = (V', E')$ such that there is an edge in $E'$ between all pairs of nodes in $V'$.

Maximal cliques  A maximal clique in graph $G = (V, E)$ is a clique of $G$, $G' = (V', E')$ such that there is no vertex in $V \setminus V'$ whose addition to $G'$ would preserve the clique property of $G'$. Therefore, the maximal cliques are not included in any other cliques.

Maximum edge weight clique  In a weighted graph, a maximum edge weight clique is the clique whose sum of edge weights is maximum among all other cliques in the graph.

2.2. Semantic Networks

Previous research in semantic networks is one of the most important foundations that grounds the work in this thesis. Semantic network have been studied for more than half a century, and they gained even more momentum in the last decade due to the advancements of the Semantic Web. Some of the earliest works in this domain belong to Quillian [112, 113] who introduced and studied the notion of semantic memory. This work contains one of the first attempts to model semantic associations between words in the form of a network. The semantic memory contains words and their meanings, and each meaning has a definition from a dictionary which is itself represented as a directed network of words. This work was later extended by Collins and Quillian [29], by adding inference capability on top of hierarchical semantic networks. By using a structure similar to that illustrated in Figure 2.1(a), a system should be able to infer that a canary can fly, or that it has skin.
Collins and Loftus [28] develop further the idea of semantic memory. They see the hierarchical semantic structures like the one in Figure 2.1(a) as a lexical memory, while the semantic memory as a more flexible structure, organised along the lines of semantic similarity, such as that illustrated in Figure 2.1(b). In this figure, the shorter the edge between two concepts, the higher their relatedness. The two types of networks are highly compatible because the semantic network can be obtained to some extent from applying spreading activation on the lexical network. The advantage of the semantic networks of the form in Figure 2.1(b) is that they can model a greater variation of domains, while the hierarchical one from Figure 2.1(a) is restricted to domains whose concepts are taxonomically structured.

Various other semantic networks have been used like Roget’s Thesaurus [119], and WordNet [93]. Other works create semantic networks automatically starting from language dictionaries like Collins English Dictionary [138] or Longman Dictionary of Contemporary English [50, 71]. Although not a semantic network in the traditional sense, Wikipedia’s hyperlink structure has also inspired researchers [94] to use it as a semantic network. DBpedia\(^2\), which is structured knowledge base created from Wikipedia, has also attracted attention towards its potential of being used as a semantic network, by Nunes et al. [109]. We take a similar route, and in this work explore DBpedia as a semantic network.

Regarding the graph structure of semantic networks, Steyvers and Tenenbaum [127] published an analysis of the structure of semantic networks. The authors analysed three semantic networks that have different sources and were generated with different purposes. The first is a

\(^2\text{http://dbpedia.org}\)
word-association network \cite{104} that was crowd-sourced by asking the participants to note the first word that came to mind when served with a cue word. The other two semantic networks were WordNet and Roget’s thesaurus. Steyvers and Tenenbaum \cite{127} analysed these networks from a graph-based perspective and learnt that all of them presented the characteristics of scale-free, small-world networks. This means that words tend to cluster more than nodes in random graphs, the average shortest paths are quite small, and the node degrees follow a power law distribution. The last property means that there are very few words that are related to many other words, while most of the words have few neighbours.

Graphs with these properties are quite common, for example social networks and the World Wide Web, or citation networks \cite{10}. This is a very interesting insight because these properties of words and their meanings, found by mining the semantic networks, are inconsistent with properties found by analysing other types of semantic representation, like hierarchies or latent semantic representation (LSA) \cite{48, 73, 127}, but are consistent with topic model representation \cite{48}. Topic models are the focus of the next section.

2.3. Probabilistic Topic Models

The common idea behind probabilistic topic models is that documents can be represented as mixtures of topics, while topics are multinomial distributions over words. These models are generative as they try to describe how words in documents might have been generated, on the basis of latent variables \cite{126}. The topics are the latent variables that need to be estimated, as well as their distribution over documents. From the perspective of a generative process, the topic models can be defined as follows: given a document $d$, each word $w_i$ of $d$ is generated by first sampling a topic $z$ from the topic distribution of document $d$, and then by sampling a word from the topic-word distribution of topic $z$. If $P(z_i = j)$ denotes the probability that the topic of the $i$-th word is $j$, and $P(w_i|z_i = j)$ denotes the probability of word $w_i$ in the topic $j$, the distribution of the words over a document can be described as:

$$P(w_i|d) = \sum_{j \in T} P(w_i|z_i = j)P(z_i = j|d); \quad (2.1)$$

One of the first models that considered this type of generative process was probabilistic latent semantic analysis (pLSA) \cite{60}. This model makes no assumption about how the mixture
weights \( P(z|d) \) are generated, therefore the model learns these distributions only for the documents it is trained on, and there is no way to generalise the model to new, unseen documents. In order to overcome this drawback, Blei et al. [15] suggest Latent Dirichlet Allocation (LDA), a generative model that considers a Dirichlet prior on the mixture of topics. Figure 2.2 illustrates four topics extracted with LDA.

In this representation, the similarity between two words is usually computed as conditional probability [48, 126]:

\[
sim(w_1, w_2) = P(w_1|w_2) = \sum_{j=1}^{T} P(w_2|z = j)P(z = j|w_1)
\] (2.2)

This computation leads to asymmetric similarity. A discussion about its implications, as well as correlations to human assessment of similarity, which often proved to be asymmetric, can be found in Griffiths et al. [48] work.

Lately, many approaches to topic modelling have been researched, that extract hierarchical topics [18, 82] or that consider the ordering of words [47, 141], and also dynamic topic models that capture the evolution of topic models over time [17].

A relevant issue with respect to topic models is that of polysemy. These models work with words only, therefore an interesting question is if and how do they handle words that can have distinct meanings. In these models, words are part of topics. Therefore a polysemous word will occur with comparatively high probability in multiple topics, and the other words in the topic will implicitly determine the word’s meaning. For instance, the word \textit{cell} might occur in a topic with words \texttt{[biology, cell, life, biochemistry]}, or in a topic

\begin{table}
\begin{tabular}{|l|c|}
\hline
\textbf{word} & \textbf{prob.} \\
\hline
DRUGS & .069 \\
DRUG & .060 \\
MEDICINE & .027 \\
EFFECTS & .026 \\
BODY & .023 \\
MEDICINES & .019 \\
PAIN & .016 \\
PERSON & .016 \\
MARIJUANA & .014 \\
LABEL & .012 \\
ALCOHOL & .012 \\
DANGEROUS & .011 \\
ABUSE & .009 \\
EFFECT & .009 \\
KNOWN & .008 \\
PILLS & .008 \\
\hline
\end{tabular}
\begin{tabular}{|l|c|}
\hline
\textbf{word} & \textbf{prob.} \\
\hline
RED & .020 \\
BLUE & .099 \\
GREEN & .096 \\
YELLOW & .073 \\
WHITE & .048 \\
COLOR & .048 \\
BRIGHT & .030 \\
COLORS & .029 \\
ORANGE & .027 \\
BROWN & .027 \\
PINK & .017 \\
LOOK & .017 \\
BLACK & .016 \\
PURPLE & .015 \\
CROSS & .011 \\
COLORED & .009 \\
\hline
\end{tabular}
\begin{tabular}{|l|c|}
\hline
\textbf{word} & \textbf{prob.} \\
\hline
MIND & .081 \\
THOUGHT & .066 \\
REMEMBER & .064 \\
MEMORY & .057 \\
THINKING & .050 \\
PROFESSOR & .028 \\
FELT & .025 \\
REMEMBERED & .022 \\
THOUGHTS & .020 \\
FORGOTTEN & .020 \\
MOMENT & .020 \\
THINK & .019 \\
THING & .016 \\
WONDER & .014 \\
FORGET & .012 \\
RECALL & .012 \\
\hline
\end{tabular}
\begin{tabular}{|l|c|}
\hline
\textbf{word} & \textbf{prob.} \\
\hline
DOCTOR & .074 \\
DR. & .063 \\
PATIENT & .061 \\
HOSPITAL & .049 \\
CARE & .046 \\
MEDICAL & .042 \\
NURSE & .041 \\
PATIENTS & .029 \\
DOCTORS & .028 \\
HEALTH & .025 \\
MEDICINE & .017 \\
NURSING & .017 \\
DENTAL & .015 \\
NURSES & .013 \\
PHYSICIAN & .012 \\
HOSPITALS & .011 \\
\hline
\end{tabular}
\end{table}
The words around the ambiguous word are able to implicitly disambiguate it. In this thesis, we exploit this “implicit disambiguation”, and use the topics in order to explicitly disambiguate the words of topics, by finding their senses in external knowledge base.

Another aspect of topic modelling that is relevant for this thesis, is that topic models capture semantics from the perspective of word usage in natural language. With topic modelling, the semantics emerges from the contexts in which words are used in text without any other external source. This contrasts with the semantics of the semantic networks, whose aim is to understand and model how concepts are organised in human memory from a conceptual, rather than a linguistic perspective. While words’ meanings are implicit in the case of topic models, they are explicit in the case of most semantic networks.

Nevertheless, Steyvers and Tenenbaum [127] and Griffiths et al. [48] have researched if topic models capture word associations similarly to semantic networks. The basic idea of these experiments was to compare the structure of semantic networks to the structure of networks that result by connecting words based on their similarity in topic model representation. They report that the networks obtained from LDA topic models have the same structure as the semantic networks. This finding shows that the two representations of semantics are compatible, while they also give different insights. This is highly relevant to the topic model labelling work presented in this thesis, in which we try to map the words in topics to their corresponding explicit meaning in semantic networks, through disambiguation. This mapping reveals a subgraph of the semantic network that represents the topic, and we further analyse the graph to extract other related concepts, to better understand the relations between words and to automatically label the topic.

2.4. Semantic Web and Linked Data

The vision of the Semantic Web takes the idea of semantic networks to the scale of the World Wide Web. It inherits from the semantic networks the graphical representation of knowledge, and extends it by incorporating more reasoning capabilities on top of it, and by creating an environment where semantic networks are interconnected, and easily accessible over the World Wide Web. The aim of Semantic Web is for web-pages to contain computer understandable data, that can be interpreted and used automatically by machines, facilitating sharing and usage of large amounts of information.
2.4.1. Semantic-Web Notions

The extension the Semantic Web brings to the Web, revolves around the idea of “meaning” that is understandable by machines. This requires that data be semantically structured. Moreover, the data and its structure need to be understood by the publisher as well as by consumers. This entails the need of a common way of expressing and sharing meaning, and leads to a standardisation effort under the care of the World Wide Web Consortium\(^3\) (W3C).

The traditional semantic networks relate either words or meanings, but the vision of the Semantic Web is that anything can be represented, including people, communities, projects, web pages and artefacts. In the Semantic Web all these heterogeneous entities are called resources and they are uniquely identified by Universal Resource Identifiers (URI), a generalisation of URLs. They are interconnected by properties that are themselves URIs. The building block of the knowledge representation is the triple, a formation that corresponds to a statement, formed by a subject, a predicate and an object. While the subject and predicate are URIs, the object can be either a URI or a literal. This data format is called Resource Description Framework (RDF) \(^{140}\), and results in a graph representation of data, where the subject and object are nodes, and the property is a labelled edge between them. URI and RDF are W3C standardised and can be seen together with the other Semantic Web technologies in the architecture from Figure 2.3, that is referred to as the Semantic Web stack.

While RDF and URI provide means for expressing meaning, the standardised syntax for sharing this meaning is XML. In an analogy to human communication, the equivalent is that humans speak in words that form sentences, and they communicate orally or by written text. Therefore, these technologies alone do not guarantee that transmitted data is understood, because multiple URIs can be used to describe the same object \(^{13}\), just as things have different names in different languages. The Semantic Web tackles this by the use of ontologies. When a party publishes semantic data, it provides the used ontology, which consists of a vocabulary of terms, their relations, as well as the inference rules that can be used for reasoning on top of the data. An important desideratum of the Semantic Web community is the usage of the ontologies standardised by W3C, or the provision of mappings to them, to limit the efforts involved in “translating” data.

One of the most commonly used ontologies standardised by W3C is RDF Schema (RDFS), a basic ontology for creating classes and properties, organised in simpler structures, like hierarchies. The other mostly used ontology is Web Ontology Language (OWL) which supports the

\(^3\)http://www.w3.org/
definition of more complex relations between entities, governed by constraints or characterised by certain properties like for example transitivity. The Rule Interchange Format (RIF) adds even more flexibility to defining rules that govern the data.

URIs are defined within **namespaces** that correspond to ontologies or knowledge bases, and the namespace is used as part of the URI for identification of the URI source. For example, the URI `http://www.w3.org/2004/02/skos/core#broader` identifies the namespace `http://www.w3.org/2004/02/skos/core#` and the resource name `broader`. For simplification of syntax, namespaces are often assigned short names, that in the context of URIs are referred to as **namespace prefixes**. For example, the previous URI, written in the prefixed format becomes `skos:broader`.

We now give some examples of ontologies and vocabularies, that are commonly used and that we will refer to throughout this thesis. Their corresponding namespaces and the used prefixes are shown in Table 2.1:

**RDF Syntax** defines basic classes like `rdf:Class, rdf:Resource, rdf:Property`, and properties, for example `rdf:type` which is used to link instances to their classes.

**RDF Schema** defines classes and properties that are useful for creating taxonomies like `rdfs:subClassOf, rdfs:label`, etc.
Table 2.1.: Example namespaces and their usual prefixes. *DBpedia is a knowledge base, not an ontology.

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Prefix</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a></td>
<td>rdf:</td>
<td>RDF Syntax</td>
</tr>
<tr>
<td><a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a></td>
<td>rdfs:</td>
<td>RDF Schema</td>
</tr>
<tr>
<td><a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a></td>
<td>owl:</td>
<td>OWL</td>
</tr>
<tr>
<td><a href="http://www.w3.org/2004/02/skos/core#">http://www.w3.org/2004/02/skos/core#</a></td>
<td>skos:</td>
<td>SKOS</td>
</tr>
<tr>
<td><a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a></td>
<td>dc:</td>
<td>Dublin Core</td>
</tr>
<tr>
<td><a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a></td>
<td>dcterms:</td>
<td>Dublin Core Terms</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a></td>
<td>dbpedia:</td>
<td>DBpedia*</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a></td>
<td>dbpedia-owl:</td>
<td>DBpedia Ontology</td>
</tr>
</tbody>
</table>

**OWL** defines classes and properties to be used for creating complex ontologies, including rules. It contains the class `owl:Thing`, which is the class that all OWL instances are part of. Other OWL class examples are `owl:ReflexiveProperty`, `owl:Ontology`, `owl:SymmetricProperty` and properties like `owl:disjointWith` and others.

**SKOS** defines classes and properties that can be used to express the basic structure and content of concepts from thesauri, taxonomies, glossaries and other types of controlled vocabulary. Examples of classes contain: `skos:Concept`, `skos:Collection` and some of the properties that link SKOS Concepts are: `skos:broader`, `skos:narrower` and `skos:related`, which define the type of relation between concepts.

**Dublin Core** defines fifteen core properties that can be used to describe both web resources, and physical resources. Some examples are: `dc:creator`, `dc:title`, `dc:language`, `dc:publisher`, etc.

**Dublin Core Terms** extends Dublin Core by adding new classes like `dcterms:Agent` and more properties like `dcterms:subject`, which links a resource for instance a document, to the resources or concepts it is about.

**DBpedia Ontology** contains classes and properties that can be used to express encyclopedic knowledge. It was created based on the most commonly used infoboxes within Wikipedia. Some classes are `dbpedia-owl:Person`, `dbpedia-owl:Place`, `dbpedia-owl:Species`, `dbpedia-owl:Organisation` and some properties are `dbpedia-owl:profession`, `dbpedia-owl:religion`, `dbpedia-owl:spouse`, etc.

The upper layers of the Semantic Web architecture deal with reasoning, provenance of data and trust, and applications. Another important component of the architecture is the data encryption component.
By populating ontologies with instances, semi-structured datasets can be created. Some examples of datasets are GeoNames\(^4\) which contains RDF descriptions to all its geographical names and DBpedia\(^5\) which is the RDF version of Wikipedia. The RDF datasets are usually stored in so-called **triple stores** that can typically be queried with SPARQL (SPARQL Protocol and RDF Query Language)\(^6\), a query language for RDF closely related to SQL.

### 2.4.2. Linked Data

One of the main outcomes of the research towards a Semantic Web, is Linked Data \(^7\). The Semantic Web provides the technical environment for datasets to be shared and linked, and its value can only be determined if data providers transform their data in RDF, publish and interlink it to other previously published datasets. Simply publishing data in RDF format is not enough for realising the potential of Semantic Web. The data sources should be linked, so that knowledge from multiple datasets can be integrated. This massive, interconnected network of structured datasets is the so-called Linked Data Cloud. Great value is expected from interconnectivity of this dataset: “By enabling seamless connections between data sets, we can transform the way drugs are discovered, create rich pathways through diverse learning resources, spot previously unseen factors in road traffic accidents, and scrutinise more effectively the operation of our democratic systems.”\[^{54}\]. Part of Linked Data is the Linked Open Data which, as the name suggests, contains the RDF representation of non-proprietary data sources. There are more than 300 datasets published and linked as part of the Linked Open Data, in domains like media, geography, publications, social networking and fora, government and life sciences \[^{34}\]. The most linked-to dataset is DBpedia, which offers encyclopedic knowledge extracted from Wikipedia in RDF format.

In this thesis, we are using DBpedia, WordNet and YAGO as sources of external semantic knowledge. We describe in more detail these knowledge bases in Section 2.5. We are not aiming at either producing or publishing data as part of Linked Data, but we are interested in solving particular problems related to topic labelling and polysemy of words, by exploring this freely available data. Among the semantic technologies we used, the main ones are the OpenRDF Sesame\(^8\) triple store, HDT (Header Dictionary Triples)\(^9\)[6] for storing and querying

---

\(^4\)http://www.geonames.org  
\(^5\)http://dbpedia.org  
\(^6\)http://www.w3.org/TR/rdf-sparql-query/  
\(^7\)http://linkeddata.org  
\(^8\)http://www.openrdf.org  
\(^9\)http://www.rdfhdt.org
the used DBpedia datasets on the local machines, and SPARQL for querying the public DBpedia SPARQL endpoint\textsuperscript{10}.

### 2.4.3. Semantic Web and Topic Models

The Semantic Web provides semantic networks whose main building blocks are human-defined concepts. At the same time, statistical learning techniques like topic modelling, provide an alternative approach to concept definition, driven by data rather than prior knowledge \cite{24}. As seen in Section 2.3, the idea of complementarity between semantic networks and latent semantics, has developed independently from the Semantic Web, but the Semantic Web has since become the most important “provider” of semantic networks.

In 2008, Chemudugunta et al. suggest the Concept-Topic Model (CTM), which is a generative probabilistic model that combines topic modelling and ontologies. Given an ontology and a corpus, the aim is to model the documents in the corpus as probability distribution over predefined concepts from the ontology, and also model the ontology concepts as probability distributions over words. An extension of the model builds the bag-of-words from the definition of the concept itself, merged with the definitions of all its descendants in the ontology. Therefore the model is similar to the topic models, but it contains the constraint that a word can only have a non-zero probability within a concept, if it belongs to the bag-of-words representation of that concept.

Another generative probabilistic model that combines predefined semantic knowledge to latent semantics is the Entity-Topic Model (ETM) \cite{69}. An entity in this work denotes a concept, from an external knowledge base, that a word in the text has been annotated with. Therefore the input to this model is a corpus of documents, that have already been annotated with concepts, either manually or automatically. The aim of ETM is to capture the word distribution for topics, for entities as well as for topic-entity pairs. The intuition is that for example an author, considered an entity, can be represented by a probability distribution over words. At the same time, a scientific topic can be similarly represented, and the two representations differ from the probability distribution that represents the pair of author and scientific topic.

In this thesis, we are also aiming of utilising both these sources of semantic knowledge: prior knowledge and latent topic models. However, we research to what extent the association of words in topic models corresponds to the association of their meanings in the semantic network. As Chemudugunta et al.’s approach \cite{24}, our work annotates documents with semantic

\textsuperscript{10}http://dbpedia.org/sparql
concepts, but in our case there is a one-to-one correspondence between semantic concepts and topic words. The actual topic model is assigned an additional graph representation, that is extracted directly from the semantic network. We use this graph representation to learn more about the topic, for example other concepts that are related to it, the explicit relations between the topic’s words, and also a suitable label for the topic, that is selected based on its centrality in the graph. We use DBpedia as semantic network.

2.5. Knowledge Bases

2.5.1. DBpedia

DBpedia\textsuperscript{11} is a knowledge base resulted from the extraction of structured information from Wikipedia\textsuperscript{12} [7, 14]. As a consequence, every Wikipedia article has its corresponding DBpedia concept, which is assigned a URI. The properties of the DBpedia concepts are populated automatically by extracting structured information from Wikipedia articles. This information is obtained by parsing semi-structured content like the Wikipedia infoboxes, lists, tables and categorisation.

Currently, DBpedia contains approximately 4.0 million concepts classified based on three classification schemata:

**Categories** correspond to the Wikipedia categories. There are approximately 415,000 categories, organised by using the SKOS vocabulary. As such, the category concepts are of type \texttt{skos:Concept} and the relations between categories are the ones provided by SKOS vocabulary. The predicate connecting DBpedia concepts to their categories is \texttt{dcterms:subject}. As a classification schema, this dataset poses several challenges: the categories do not form a hierarchy, but a graph that contains cycles [14], and as they are manually assigned by the Wikipedia editors, they are not evenly distributed. In the work presented in this thesis, the Categories graph plays a very important role, because, as we later show, they are crucial for establishing relatedness between concepts.

**DBpedia Ontology** contains 529 manually created classes, organised in a hierarchy. 3.22 million DBpedia concepts are classified to these classes through the \texttt{rdf:type} property.

\textsuperscript{11}http://www.dbpedia.org
\textsuperscript{12}http://www.wikipedia.org
The classes are interrelated with the `rdfs:subClassOf` property. This ontology also defines the concept properties automatically extracted from the Wikipedia Infoboxes.

**YAGO** is an automatically created knowledge base [58], derived from Wikipedia, WordNet and GeoNames[13]. Its backbone is a hierarchy of about 350,000 classes, derived from mapping Wikipedia Categories to WordNet. Since DBpedia assigns to concepts their Wikipedia Categories, this facilitated automatic mapping between the DBpedia concepts and the YAGO classes [14]. The used predicate is `rdf:type`, and the subsumption property used by YAGO is `rdfs:subClassOf`.

Besides the concept properties captured in the previous datasets, in this thesis, particularly in the work that addresses WSD, we also use the following properties [14]:

- **labels** by using the `rdfs:label` property, with the value being the Wikipedia article title.
- **definition and abstract** which are the first paragraph of the Wikipedia article and the entire text before the table of contents, respectively. The corresponding properties are `rdfs:comment` and `dbpedia-owl:abstract`;
- **redirect concepts** which are the DBpedia concepts that correspond to Wikipedia articles that have automated redirect links to the Wikipedia article in question. These redirecting articles act as synonyms to their targeted article. Similarly, in DBpedia the redirect concepts are the synonyms to the targeted DBpedia concept. The used property is `dbpedia-owl:wikiPageRedirects`;
- **disambiguation pages** are the concepts that refer different meanings of a homonym word. The DBpedia predicate for this property is `dbpedia-owl:wikiPageDisambiguates`;

The DBpedia knowledge base is publicly available and stored in RDF format. It can be accessed either online, by using available services like the DBpedia SPARQL Endpoint[14], or it can be downloaded and accessed locally[15].

### 2.5.2. WordNet

WordNet [92] is the most widely used lexical dataset in natural language processing and in information retrieval. It combines words in synonymy groups that are called *synsets*. The

---

synsets are organised hierarchically and also linked by other semantic relations. The hierarchical properties of nouns are hypernymy/hyponymy (ISA relation) or meronymy (PARTOF relation). Verbs are also arranged hierarchically, while adjectives have links to their antonyms. Currently WordNet contains 117000 synsets.

WordNet as a dataset has served and strongly influenced the research and advancements in WSD. In the following chapter, we detail some of the fundamental approaches that used this dataset for WSD. WordNet is available for download\footnote{http://wordnet.princeton.edu/wordnet/download/}, and is also available in RDF/OWL format\footnote{http://www.w3.org/TR/wordnet-rdf/}.
Chapter 3.

Related Work

In this chapter, we review the work that tackles the same problems as this thesis, and the methods used, their particularities, advantages and limitations. In the first part we focus on unsupervised WSD, while in the second part we focus on topic labelling.

3.1. Related Work on Unsupervised Word Sense Disambiguation

The WSD problem has been targeted through a multitude of diverse approaches, some being even half a century old. Agirre and Edmonds [1] and Navigli [101] provide a comprehensive account of this field.

In our work, we have identified a couple of limitations of the approaches in previous literature, with respect to the following two scenarios that are highly relevant for document linking to structured knowledge: disambiguation with multiple sense inventories simultaneously, and disambiguation with general graph-structured knowledge bases. To the best of our knowledge, there is no work to tackle the first scenario, and existing methods that might be suitable, such as previous gloss-based WSD methods, suffer from limitations that make them suboptimal for this task. In Chapter 4 of this thesis, we propose an approach to gloss-based WSD that overcomes some of these limitations.

For the second scenario, while semantic networks have been vastly studied for WSD, they are usually assumed to be taxonomies (IS-A hierarchies). DBpedia is a graph-structured knowledge base extracted from Wikipedia and it is more than 40 times bigger than the most studied semantic network, WordNet. It is the result of Wikipedia community effort as well as
Related Work

automatic information extraction. Its structure is therefore not as regular as that of WordNet and other manually created sense inventories, and contains much more noise. Therefore, works that tackle WSD with DBpedia rely on its connections to Wikipedia, and use the content of Wikipedia articles as well as the hyperlinks between the articles. The main limitations of these works is that they require preprocessing of entire Wikipedia, and lack portability to other graph-based knowledge bases. In Chapter 5 of this thesis, we propose an approach for disambiguation with DBpedia that makes exclusive use of DBpedia’s underlying graph structure. It overcomes the limitations of Wikipedia based approaches, and is related to the previous work in WSD with semantic networks.

In this section, we first give an overview of the most relevant WSD approaches, and highlight the typical directions, as well as the contributions brought by our work. Afterwards, we focus in more detail on the approaches that are most similar to ours, and highlight their particularities and limitations, to further contrast them to the approaches we propose in this thesis.

3.1.1. Word Sense Disambiguation - The Big Picture

In literature on WSD, approaches are mostly grouped in several classes as follows. Some of the very first approaches are the semantic network based WSD [113] and rule-based WSD [32]. As we later show in this chapter, semantic network based WSD has been vastly studied, especially with WordNet as a sense inventory. As for the rule-based approaches, their adoption was limited due to the great amount of rules that need to be defined, which renders this class of methods very laborious compared to other WSD methods. Gloss-based WSD was introduced in 1986 by Lesk [80], and this class of methods requires the least human effort, as the only knowledge they need about the words is their glosses. With the availability of big text document corpora, statistical WSD developed. These approaches model senses based on statistics on their use in sense annotated text corpora [50, 143]. The development of machine learning also lead to a variety of supervised WSD [77] methods, as well as bootstrapping WSD [144] methods that require very little human input and use bootstrapping for automatic supervision. This direction gained more momentum once Wikipedia became available, because the hyperlinks of articles can be used directly for system training [62]. Another wide class of WSD approaches is that of domain specific WSD, that use domain knowledge to enhance disambiguation [12, 21, 52, 67, 120, 133, 139]. The use of specialised knowledge for disambiguation is outside the scope of this thesis as we are rather interested in disambiguation with general domain knowledge.
However, most unsupervised WSD approaches contain several elements that help identify distinctions and similarities between them beyond the borders of their assigned classes. These are: (i) the way they model the senses, (ii) the context they consider for disambiguation, (iii) the way of measuring how senses fit the context, (iv) the way of measuring how senses are related to each other, and (v) the strategy in which all these pieces are put together to achieve disambiguation. We therefore propose a conceptual framework illustrated in Figure 3.1, expressed by means of the following properties:

**Sense model** decides how senses are represented;

**Context model** specifies in what context the words are disambiguated (e.g., sentence, paragraph, etc.) and how it is modelled;

**Context fitness measure** compares the possible senses to the context;

**Semantic relatedness measure** compares senses; it is used either together with or as an alternative to the context fitness;

**Disambiguation strategy** uses the context fitness and/or the semantic relatedness to decide the intended senses of ambiguous words.

![Figure 3.1: Conceptual framework of word-sense disambiguation](image)

We now use this framework and review the main features of unsupervised WSD approaches in literature, and show how they relate to the work presented in this thesis.

**Sense model**

One of the decisive factors in WSD is the way senses are formally modelled. This decision strongly depends on the sense inventory and influences how the senses can then be scored
and ranked. In the case of semantic networks for instance, senses are usually nodes in the network \([2, 3, 11, 91, 136, 138]\) while in statistical models, senses are typically represented as multi-dimensional vectors of terms \([22, 43, 72, 89]\), for instance TF-IDF, or other weighting schemes. Other approaches use the set of words that belong to the gloss \([31, 80]\). In many approaches that disambiguate to Wikipedia, the senses are the Wikipedia articles, and they are modelled as nodes in the hyperlink network \([36, 44, 46, 81, 95, 116, 145]\). In our work, in Chapter 4 we introduce a novel vector space model for representing glosses, that addresses some of the challenges encountered when comparing glosses with high length difference. In Chapter 5, as we look into an approach for WSD with semantic networks, the senses are nodes in the semantic network.

**Context model**

Words are always disambiguated in a context. *Word meanings do not exist in isolation. Each word must be interpreted in its context.* \([99]\). Some typical contexts used in literature are: the sentence or paragraph the ambiguous target word occurs in, the whole document or a fixed window of text around the target word. Thus, most WSD methods use the approach of selecting the disambiguation context as the text in the vicinity of the word. Other works follow the intuition that proximity of words in text does not necessarily reflect their similarity/relatedness. Typically, these works use manually created sets of related words \([117, 118, 138]\), or group the words in a text on the fly to create the so-called *lexical chains*, and use these groups as disambiguation contexts \([11, 57]\). Where disambiguation of tags is required, a tag is disambiguated in the context of the set of all other tags used by the same user to tag the same resource \([43]\).

In our work, we follow the intuition from distributional semantics that the meaning of a word is better emphasised by the words in whose vicinity it *often* occurs. We therefore disambiguate words in the context of topics obtained through probabilistic topic modelling and through co-occurrence based clustering. Thus, our work bridges the explicit semantics from knowledge bases to the distributional semantics of words. To the best of our knowledge, this is the first work that attempts disambiguation in the context of topic models. Nevertheless, from the context representation perspective, the context is usually modelled as a set of terms or a vector of terms, regardless of its source. From this perspective, our methods are generalisable to any context that can be modelled as a set of terms.
Context Fitness

As previously mentioned, disambiguation of words is dependent on the context in which the words are used. It is typically computed as a similarity or relatedness metric between the senses and the context. One of the most basic measures is to count the overlapping words between the gloss of the sense and the context \([80, 90]\). Approaches that model senses as multi dimensional vectors usually employ cosine similarity \([22, 33, 43, 89]\). In our work we do not use any explicit measure of context fitness, but attempt to optimise the relatedness between the meanings of the target words.

Semantic relatedness

Besides context fitness, most approaches consider the need for the chosen senses for ambiguous words in the same context to be related. Also, if non-ambiguous words can be mapped to their senses in the inventory, these senses are good clues for the disambiguation of the ambiguous words. Therefore semantic relatedness measures between senses have been widely used for WSD.

A recurring distinction in literature is between semantic similarity and semantic relatedness. Works that consider any type of relations between concepts usually refer to “semantic relatedness” or “association”, while works that only use the IS-A relation as evidence for disambiguation, refer to “semantic similarity”. In this thesis, we call this dimension “relatedness” because it is more generic and also includes the “similarity” relationship. However, when we discuss related work found in literature, we use the term originally used by the authors. The disadvantage of algorithms considering only hierarchical relations, and as such assessing similarity, is that for example, they find no evidence for disambiguating the pair doctor and health due to their dissimilarity, despite their strong association.

The semantic relatedness measure is by far the property that best distinguishes the various WSD approaches. Typically, when senses are modelled as multi-dimensional vectors, the relatedness measure used is the cosine similarity \([42, 145]\) or the scalar product \([33]\). Methods that model senses as sets of words or phrases use measures of word overlap \([8]\). Dice coefficient \([137]\) as well as Jaccard coefficient \([72, 137]\). A multitude of semantic relatedness measures have been devised for hierarchical semantic networks, mostly for IS-A hierarchies, based on the paths between senses \([65, 76, 83, 84, 114, 117, 124, 130, 142]\). While many of them were not introduced necessarily for WSD, they have been used for WSD \([123]\). With respect to works that disambiguate with Wikipedia and model senses based on the hyperlink
network between articles, some of the most popular relatedness measures follow the intuition that two senses (i.e., Wikipedia articles) are related if many other senses contain hyperlinks pointing to both [95, 116]. Other works consider two Wikipedia articles more related if they contain hyperlinks to the same articles [46].

In this thesis, in Chapter 4 we use the scalar product between senses modelled with a novel vector space sense model, and we also propose a measure that captures relatedness between two or more senses. In Chapter 5, we measure sense semantic relatedness in semantic networks using established as well as novel measures that do not need a hierarchical knowledge base, but work on general graphs.

**Disambiguation strategy**

The *disambiguation strategy* refers to how the context fitness and / or the semantic relatedness measures are used to decide which of the candidate senses is most likely the intended sense of the ambiguous word. Much of related work views WSD as a “classification” problem: given a word occurrence in a context and multiple possible classes, decide what class the word belongs to [1, p. 2]. Approaches that follow this reasoning attempt disambiguation one-word-at-a-time, and ignore the fact that the correct senses of the ambiguous words should be all related.

To unify these two views, we view the WSD problem as an optimisation problem, and distinguish between approaches that achieve a **local optimisation** (the “classification” perspective), and approaches that achieve **global optimisation**. Among works that achieve local optimisation, most of them extend the context with all the possible meanings of all ambiguous and non-ambiguous words. Since the optimisation in these cases is local, but also considers the senses of other words, we call this class of approaches as **extended-local optimisation**.

To help us illustrate the distinction between these three types of systems, we use a toy example, of disambiguating the words *orange* and *apple* in the group of words [orange, apple, fruit]. We consider apple to have two possible senses, the one referring to the fruit, and the one referring to the Apple Inc. company. We also consider orange to be ambiguous and have three possible senses, the fruit sense, the colour sense and Orange, the telecommunications company. We consider fruit to have exactly one sense, and therefore unambiguous.

**Local optimisation** - In this type of approaches, given a context that contains multiple ambiguous words, the correct sense for each word is the one that has the maximum fitness to the context. Therefore, these approaches only use context fitness measures. Given a
word \( w \), the disambiguated sense \( s^*_w \) is the one that optimises the function:

\[
s^*_w = \arg\max_{s \in \text{Senses}(w)} \text{contextFitness}(s, \text{Context})
\]

(3.1)

where \( \text{Senses}(w) \) represents the set of senses of word \( w \), \( \text{Context} \) is the context the word \( w \) occurs in, and the function \( \text{contextFitness} \) is the measure used to compute the context fitness.

In the toy example, when disambiguating the word \text{apple}, the sense is chosen that has highest context fitness to the context \{\text{orange, fruit}\}. This type of approaches have the advantage of being very straightforward, but they are highly influenced by the choice of words in the context. With development of big knowledge bases like Wikipedia, where senses are text articles modelled in multi-dimensional vectors, disambiguation can be broadly interpreted as a search task where the query is the context, and the searched dataset is the set of candidate articles. Approaches of this type achieve local optimisation, and have been used with Wikipedia \[22, 90\] as well as DBpedia \[43, 89\].

\textbf{Extended-local optimisation} - In this type of approaches, given a context \text{Context} with multiple words, the correct sense for each word is the one that maximises a function of relatedness to all possible senses of all other context words. Sense \( s^*_w \) that disambiguates the word \( w \) is selected from the possible senses of word \( w \), \( \text{Senses}(w) \), so that it optimises the following function:

\[
s^*_w = \arg\max_{s \in \text{Senses}(w)} \sum_{w' \in \text{Context} \setminus w} \sum_{s' \in \text{Senses}(w')} \text{rel}(s, s')
\]

(3.2)

where \( \text{rel}(s, s') \) represents the function that computes the relatedness between two senses. For simplicity, Formula (3.2) uses the sum as the aggregate function for the pairwise relatedness values, but other aggregate functions can also be used.

Taking the same example, the disambiguation of word \text{apple} considers the relatedness of the senses of \text{apple} to the three possible senses of word \text{orange}, and to the sense of the unambiguous \text{fruit}. It selects the sense for \text{apple} that maximises the aggregate of these three scores. The word \text{orange} is disambiguated similarly.

Thus, all \text{apple} senses influence the disambiguation of \text{orange} including the Apple Inc. sense. Therefore, while these approaches are sometimes called global because they consider all the possible senses of all ambiguous words, the actual performed optimisation
is local, as no global function is optimised. These approaches are quite fast, but the selection of senses is contaminated by the wrong senses of the other ambiguous words. This type of disambiguation strategy is the most common. A very good example of extended-local strategies to disambiguation are the graph centrality based approaches, that we discuss in more detail in Section 3.1.3.

**Global optimisation** - In this type of approaches, disambiguation is treated as a combinatorial optimisation problem. Given multiple ambiguous words, the correct sense for all ambiguous words are selected *simultaneously*, by maximising a function of relatedness between the selected senses. This function that scores a combination of senses is called *coherence*. Therefore, in these approaches we encounter the notion of solution that denotes a combination of senses. Having a context of \( n \) words, with each word \( w_i \) having \( m_i \) possible senses, a solution \( R \) contains \( n \) senses, one sense for each word. There are \( \prod_{i \in [1..n]} m_i \) solutions. We denote the set of all solutions as \( R \). A solution \( R^* \) is chosen that has the highest coherence. The most common way of computing the coherence of a solution \( R \) is by summing up all the pairwise relatedness scores of the senses in \( R \), as shown in Formula (3.3). In this approach, the disambiguated sense \( s^*_w \) of word \( w \) is the sense of \( w \) that belongs to solution \( R^* \) as shown in Formula (3.4)

\[
R^* = \arg \max_{R \in R} \sum_{s \in R} \sum_{s' \in R; s' \neq s} \text{rel}(s, s'); \tag{3.3}
\]

\[
s^*_w = R^*[w]; \tag{3.4}
\]

In the same example, the system would consider all six sense combinations of orange and apple and fruit. It then selects the combination with the highest score. The final senses are selected simultaneously, thus ensuring global optimisation.

Global optimisation has the advantage of producing good results but the selection of the best combination of senses is NP-hard. It was first used by Cowie et al. [31], who suggested approximate search with simulated annealing. Sussna [130] also evaluated this strategy and they show that it produces better results than other methods, but that it only works for small contexts (less than 15 ambiguous words) due to the computational complexity. Global optimisation has been used more often recently, mainly with Wikipedia [36, 59, 72]. In order to reduce the computational complexity, all works propose an approximate search. Kulkarni et al. [72] write it as an integer programming problem and relax it to a linear problem. Hoffart et al. [59] and Fahrni et al. [36] use a
A greedy strategy to remove senses that are least likely to be correct, and then Hoffart et al. [59] perform an exhaustive search on the remaining senses, while Fahrni et al. [36] select an approximate solution with beam search strategy.

Therefore, the **local** and **extended-local** strategies disambiguate words one-by-one, while the **global** strategy disambiguates all words simultaneously. This is why we also refer to it as **joint disambiguation**. The WSD approaches we propose in this thesis use a **global optimisation** strategy. The first, in Chapter 4 proposes a new coherence measure for gloss-based global disambiguation that performs better than the traditional sum of pairwise relatedness. The second approach that we propose in Chapter 5, proposes a branch-and-bound (B&B) strategy that achieves a complete search for the optimal solution. In case the size of the search space overpasses a preset threshold, we use an approximate search on top of the B&B strategy that splits the problem into multiple smaller subproblems by splitting the disambiguation context.

We now analyse in more detail the approaches that are most related to ours.

### 3.1.2. Gloss-based WSD

Gloss-based approaches only rely on the glosses of senses for deciding which sense of an ambiguous word is appropriate in a given context. These approaches to disambiguation are versatile as they do not rely on a pre-given background structure that unifies the possible senses. As such they only require the senses to have glosses, a condition that is easily fulfilled by most thesauri and dictionaries. These methods have the advantage that they can disambiguate with multiple sense inventories simultaneously, an advantage which, to the best of our knowledge, no previous approach exploited. The relatedness is established by measuring the text similarity of the glosses. Since they only rely on the texts of glosses, their performance is often surpassed by other methods that use deeper knowledge. However, their study is important for the scenario of disambiguation with multiple sense inventories, as well as for the case when the glosses are the only knowledge available for disambiguation.

Lesk [80] published the first gloss-based approach, that relies on the intuition that the definition of the correct sense of an ambiguous word should have high overlap with the text where the ambiguous word is used. Therefore, the senses as well as the context are modelled as sets of words. The method uses word overlap as a measure for context fitness, and achieves a **local optimisation**. Cowie et al. [31] recognized the need of optimising the relatedness between the final selected senses. They therefore proposed a **global optimisation** method that scores all possible combinations of senses based on the number of words their definitions have in
common, and picks the combination with the highest sum of pairwise scores. It uses simulated annealing for finding the approximate best solution.

Banerjee and Pedersen [8, 9] bring several enhancements to the gloss overlap algorithms of Lesk [80] and Cowie et al. [31]. First, the sense model is enhanced by also including the sets of words of the glosses of all the associated synsets in WordNet: hyponyms, hypernyms, holonyms, meronyms, troponyms and attributes. Second, they define a more complex relatedness measure between glosses, $rel_{Ban}$, where the overlapping gloss phrase $o$ is weighted based on the number of words it contains, $length(o)$.

$$rel_{Ban}(gloss_1, gloss_2) = \sum_{o \in gloss_1 \cap gloss_2} length^2(o)$$

This comes from the intuition that glosses that share multi-word phrases are more similar than senses that share singular tokens. Given two WordNet synsets, their relatedness is computed by scoring all the possible pairs of glosses from their representations respectively.

Another direction for WSD relying on sense glosses is to use sense clustering [5]. This method relies on the intuition that the correct senses exhibit mutual similarity therefore they can be uncovered by clustering. Anaya-Sánchez et al. [5] model senses as multi-dimensional vectors of words extracted from the glosses. The relatedness between senses is then computed with cosine similarity. For clustering, Anaya-Sánchez et al. [5] use the extended star clustering algorithm. The obtained clusters are iteratively scored and ranked and the correct sense of each word is the sense that belongs to the highest ranked cluster. Clustering based methods achieve an extended-local disambiguation as they optimise the relatedness among senses that are not necessarily the selected ones.

Glosses have also been used in graph-based methods for disambiguation [123]. Possible senses of target words are the nodes, linked by edges that are weighted by Lesk [80]’s gloss-overlap measure. Once the sense graph is built, Sinha and Mihalcea [123] apply centrality metrics to the graph. For each word, the sense with highest centrality is selected as the correct sense. The centrality measures that the authors experimented with are PageRank, in-degree, betweenness and closeness. By creating a graph between all possible senses of target words, this method achieves an extended-local optimisation, as each sense influences all the other senses’ centralities.

Sinha and Mihalcea [123] also experiment with many non-gloss-based pairwise relatedness measures besides the gloss-based (i.e., Lesk’s [80]), for instance path-based semantic relatedness measures [65, 76, 80, 84, 118, 142], used with WordNet. Their results lead to some very
interesting conclusions that impacted this thesis. The best overall results were obtained when a combination of relatedness measures was used, as well as combining all the centralities through a voting scheme. However, the best performance for all words, with a singular relatedness measure and a singular centrality was achieved by the gloss overlap with in-degree centrality. Therefore, while having minimum requirements, the gloss-based approaches can compete with more complex approaches.

Our approach for gloss-based disambiguation is related to the work of Sinha and Mihalcea [123] as we also create an ad-hoc network, but our graphs contain both the senses and the words of the glosses. Furthermore, we aim at optimising the overall connectivity of the obtained graph, rather than local centralities, in order to address the issues caused by the local optimisation.

In this thesis, in Chapter 4, for comparison with our approach, we implement, compare and discuss in detail the performances of the following gloss-based approaches:

- Sinha and Mihalcea’s [123] WSD framework with Lesk’s [80] gloss-overlap and in-degree centrality;
- Cowie et al.’s [31] method, but with a complete search rather than approximate;
- Banerjee and Pedersen’s [9] measure for gloss overlap shown in Formula (3.5), that we evaluate with a global disambiguation approach;
- global disambiguation with gloss cosine similarity;

We show that the gloss-overlap method as proposed by Lesk [80], and used by Cowie et al. [31] and Sinha and Mihalcea [123], as well as the enhanced measure proposed by Banerjee and Pedersen [9], have a bias towards senses with long glosses. At the same time, the length normalisation achieved by cosine similarity biases the sense selection towards senses with very short glosses. The algorithm we propose in Chapter 4 strongly reduces these biases by placing the focus on the importance of the overlapping gloss-words rather than on their count.

3.1.3. WSD with Semantic Networks

In this section, we summarise the main approaches to WSD that use a semantic network as sense inventory. In all these works, the senses correspond to nodes in the semantic network. The underlying intuition of these approaches is that the way the senses are connected in the semantic network can shed light on the relations between them and emphasise the correct senses. In this section, we use the term semantic network as an umbrella term for the types of
networks that are created independently of the disambiguation problem, but have been used for WSD. They include:

- manually built semantic networks [113]
- automatically built networks of words and concepts from dictionaries [138]
- a frame knowledge base [56]
- WordNet [3, 91, 102, 136]

**Graph Centrality-based WSD**  Many WSD researchers [3, 91, 102] explored the extent to which the centrality of a sense in a semantic network can indicate if it is the correct sense of a word used in a particular context. The main idea is that given a semantic network, the correct senses can be identified because they will exhibit a high centrality with respect to the candidate senses. Since semantic networks are typically used as static structures, the centralities of nodes are constant. Therefore the context must be used to amplify the properties of the targeted words and senses. This is the main aspect that distinguishes the various methods, apart from the used centrality measure.

Mihalcea et al. [91] extract from WordNet all the synsets that correspond to possible senses of the ambiguous words, and builds a network where these synsets are connected by edges that represent either direct or indirect relations in WordNet. Then they score all synsets by their PageRank centrality in this network and select for each word, the sense with the highest centrality. Agirre and Soroa [3] use the whole WordNet semantic network but use Personalised PageRank [53] with the teleport vector corresponding to the uniform distribution over the context words. Navigli and Lapata [102] extract a subnetwork from WordNet, containing all the possible senses of targeted words, as well as all the other synsets that belong to paths shorter than or equal to 6 between any two possible senses. All these extracted nodes are connected by the existing relations in WordNet. They experiment with the in-degree centrality, eigenvector centrality, closeness centrality, betweenness centrality and maximum flow centrality.

**WSD with Spreading Activation**  A class of approaches that are closely related to the graph-centrality ones, are using spreading activation [28]. The main idea is that by activating all the possible senses of target words, which in turn activate their neighbouring nodes (senses or words depending on the network) and so on, the correct senses of the words will receive high amounts of activation. All the activated nodes in this process keep a score of how much activation they received. The amount of activation passing from a source node to its neighbours
is function of the source node’s activation and might also be influenced by the similarity between the source and sink nodes [71] (e.g., cosine similarity) or by a fan-out factor [56, 136]. The amount of activation is also often decaying with time [136]. The process ends either when due to the time decay there is no more activation flowing in the network, or after a preset number of iterations. Typically, after the process finishes, the senses with the highest activation among the possible senses of each word, are selected. Some of the most representative works for word sense disambiguation with spreading activation are the works of Hirst [56], Quillian [112, 113], Tsatsaronis et al. [136], Veronis and Ide [138]. By computing centralities, or the activation of senses in networks that contain all the possible senses of ambiguous words, these methods achieve extended-local optimisation.

Graph properties for WSD Navigli and Lapata [102] also experiment with global optimisation, by measuring graph properties rather than node properties. For each global disambiguation solution, they extract a corresponding WordNet subgraph. On this graph, they measure compactness, graph entropy and edge density. The disambiguation would select the combination whose graph has the highest score. The authors use simulated annealing for reducing the complexity of the problem. Nevertheless, these methods achieve worse results than the extended-local measures.

This shows that the use of properties of a subgraph of the semantic network, as extracted by Navigli and Lapata [102] leads to suboptimal results. This indicates that the subgraph contains many nodes and edges that are not relevant to the relatedness between the targeted synsets. In this thesis, we also extract subgraphs from the semantic network, but we do not use graph properties of these subgraphs. Rather, we only analyse the paths between the nodes of interest in these subgraphs, as in Chapter 5, or we adapt graph measures so that they only consider the relations between our nodes of interest, as in Chapter 6.

Path-based WSD Another type of approaches that use the semantic network for disambiguation, compute relatedness between senses based on the paths connecting them in the semantic network. This direction was suggested in 1966 by Quillian [113]: “a useful measure of ‘semantic similarity’ can be obtained from the length and number of paths connecting two patriarchs. [...] but must remain an hypothesis until a really large semantic memory is set up and tested” [113, p. 79]. In his work on disambiguation, Quillian [113] weights direct links three times higher than neighbour-of-a-neighbour paths. The actual disambiguation algorithm is inspired from spreading activation.
One of the most straightforward measures that follow the same principle that paths in semantic networks can capture semantic relatedness was implemented by Rada et al. [114], and states that semantic distance between concepts can be quantified as the shortest distance between them in the semantic network.

$$dist_{rada}(c_1, c_2) = len(c_1, c_2)$$

where $len(c_1, c_2)$ represents the length of the shortest path between concepts $c_1$ and $c_2$, in a IS-A hierarchy. The external knowledge base they used was the MeSH\(^1\) (Medical Subject Heading) vocabulary. While Rada et al. [114] did not use this semantic distance for disambiguation, it has later been adapted and used for disambiguation [46, 128].

There are many other path-based semantic relatedness measures that have been proposed and subsequently used for disambiguation, but they are suitable only for hierarchies [65, 76, 83, 84, 124, 142].

In our work in Chapter 5 we focus on graph measures that might capture pairwise semantic relatedness between DBpedia senses, for use in a global disambiguation setting. Centrality measures and spreading activation do not capture pairwise similarities. However, Personalised PageRank centrality has been later used on WordNet by Agirre et al. for computing semantic relatedness between pairs of senses by combining it with cosine similarity. We therefore experiment with this combined method for capturing sense relatedness on DBpedia. We also evaluate the inverse of $dist_{rada}$ (Formula (3.6)) as a relatedness measure for use in global disambiguation. Furthermore, we experiment with path-based relatedness measures that have not been used before for WSDs, and that are applicable to any types of graphs.

### 3.1.4. Word-Sense Disambiguation with DBpedia

Although DBpedia can be seen as a semantic network, WSD approaches that use it do not use its structure. We therefore treat WSD with DBpedia separately from the semantic network based approaches.

DBpedia became available in 2007, as the structured knowledge version of Wikipedia. For its advantage of being structured and query-able, and at the core of the Linked Data Cloud due to its broad knowledge coverage, a great interest emerged into linking both structured and

\(^1\)http://www.nlm.nih.gov/mesh/meshhome.html
unstructured data sets to it. In this context, disambiguation is highly needed, to ensure the correct DBpedia concepts are linked to.

However, to the best of our knowledge, there is no work that uses DBpedia for disambiguation independently from Wikipedia. As such, these approaches rely on the content and hyperlink structure of Wikipedia. Garcia et al. [43] link and disambiguate tags from social networks to DBpedia. The authors use the mapping from the candidate DBpedia concepts to their Wikipedia articles to extract the full text of the Wikipedia article. The senses are modelled as vectors of terms, where the value for each term is its frequency in the text of the targeted sense’s Wikipedia article. Given a tag, the authors use as context for disambiguation the set of all other tags used by the same user to tag the same resource. The disambiguation strategy is local as the cosine similarity is computed as a context fitness for each candidate sense, and the sense with highest fitness is selected.

Regarding WSD in text with DBpedia, the most prominent approach is DBpedia Spotlight [89], a local disambiguation algorithm. The authors map the DBpedia concepts to their Wikipedia articles, and in order to model a sense, Mendes et al. [89] collect all the paragraphs in all Wikipedia that contain anchors to the sense’s Wikipedia article. The sense is modelled as a multi-dimensional vector comprising of the terms from all the collected paragraphs. The terms are weighted with a new scheme, called TF-ICF (term frequency - inverse candidate frequency), inspired by the commonly used TF-IDF measure. TF-ICF scores higher the terms that are discriminating between the possible senses of an ambiguous word.

Using this sense model, Mendes et al. [89] compute for each candidate sense, the cosine similarity to the context (the paragraph of the target document where the ambiguous word occurs). The authors then select the sense whose vector space representation has the highest cosine similarity to the context, among the candidate senses of the ambiguous word.

Another recent work on WSD to DBpedia is that of Hakimov et al. [51]. The authors use the network of Wikipedia hyperlinks, and weight the candidate senses based on a graph centrality measure, in an approach reminiscent of centrality-based WSD with semantic networks.

To sum up, DBpedia has not yet been used as a semantic network for word-sense disambiguation, but only as an entity dictionary for disambiguation with Wikipedia. These approaches require preprocessing, and furthermore, new senses (i.e., Wikipedia articles) cannot be considered for disambiguation until other Wikipedia articles will have referred to them. These limitations can be overcome if we develop WSD approaches that only require the DBpedia underlying graph. In Chapter 5, we propose such an approach, that does not require preprocessing, nor hyperlink analysis. Furthermore, compared with reported results in related
literature, it achieves better results than state-of-the-art WSD approaches with Wikipedia or DBpedia.

3.2. Related Work on Topic Labelling

In texts, as well as in topic models, the domain or theme is often not explicitly stated, but it is implicit and requires human interpretation. The problem of automatic topic labelling refers to automatically selecting a coherent phrase that names the otherwise implicit topics.

Several works [75, 85, 88] consider topic labelling in the same scenario as we do, where topics represented by a set of words have to be labelled. A second relevant area considers labelling of document clusters [23, 100, 108, 111, 135]. Similar to the first scenario, document clusters are often summarised as a collection of the most prominent words they contain. The third related direction deals with annotations for document indexing [30, 49, 87, 131], also called automatic topic identification [30]. Despite these different application domains, the various approaches are better distinguished by the techniques they use.

3.2.1. Label Extraction from Text

A significant part of the approaches extract the most likely label from the text [66, 88, 100, 111, 135]. In many works, the idea is that a good label for a topic or cluster of documents is a phrase that has high frequency in the topic or cluster, and is also very specific to it therefore it has a lower frequency over all topics or clusters respectively. Popescul and Ungar [111] defines such a measure for cluster labelling as

\[
\text{score} = p(\text{label}|\text{cluster}) \times \frac{p(\text{label}|\text{cluster})}{p(\text{label})}
\]

With respect to multinomial topic models, Mei et al. [88] defines a measure following a similar intuition:

\[
\text{score} = \log \frac{p(\text{label}|\text{topic})}{p(\text{label})} = \sum_{w_i \in \text{label}} \log \frac{p(w_i|\text{topic})}{p(w_i)}
\]

In this measure, called “zero-order relevance”, Mei et al. [88] therefore consider the score of a label composed of multiple words, as the sum of the scores of each word. “Basically, a phrase containing more important words in the topic distribution is assumed to be a good label” [88].
Another measure proposed by Mei et al. [88], called the “first-order relevance” has as central point the representation of candidate labels $l$ as multinomial distribution of words $p(w|l)$. This probability represents the percentage of documents in the corpus that contain the word $w$ out of the documents containing the label $l$. Then, a good topic label shows a distribution over the words of the corpus that is similar to the latent topic’s distribution, measured using the Kullback-Leibler divergence (zero if a label perfectly matches the distribution of a topic). This value is computed as the expectation $E$ of point-wise mutual information ($PMI$) between the label $l$ and the topic words given the context $D$ (i.e. the document corpus). The score $score_{Mei}$ of a label is thus computed as:

$$score_{Mei}(l, \theta) = E_{\theta}[PMI(w, l|D)] = \sum_w (p(w|\theta)PMI(w, l|D))$$  \hspace{1cm} (3.7)

$$PMI(w, l|D) = log \frac{p(w, l|D)}{p(w|D)p(l|D)}$$  \hspace{1cm} (3.8)

The most relevant noun-phrases of the corpus (Mei et al. [88] use top-1000), as measured with t-test are all scored for all topics with this measure, and the top one for each topic is selected as the topic label.

Another class of approaches attempt labelling of hierarchical clusters [100, 111, 135]. In this case, the aim is to prevent selecting the same label for sibling clusters, or for parent and child clusters. In order to prevent the same label for siblings, the most discriminative labels are selected. As for parent and child clusters, a label is suitable for a cluster if it has high relevance score for that cluster, and a lower relevance score for the parent. The higher this difference the better the label. At this stage of this work, we are not interested in labelling hierarchical topics, but we certainly envisage this scenario for future work.

The methods that attempt to find the topic/document/corpus labels in the text make the assumption that (i) the correct label can be found in the documents, and that (ii) the document/corpus is rich enough to identify a label with confidence. However, this is not always the case. For example, a cluster of documents might be about artificial intelligence without mentioning the phrase. On the other hand, it might contain many more specialised phrases that cannot be related just based on the text (e.g., probabilistic reasoning and first-order logic). This problem can be overcome by the use of external data sources, that we further explore in the next section.
3.2.2. Keyphrase Extraction from Text with Support from External Knowledge Bases

Besides the work in hand, a wide range of recent research [23, 30, 49, 75, 85, 87, 108, 131] focused on labelling topics, documents or clusters of documents with phrases extracted from external data sources. The probably most popular external knowledge base for this purpose is Wikipedia.

A research direction close to ours is focused on extracting the main keyphrases from text, by using guidance from an external knowledge base on the candidate phrase’s suitability as a keyphrase. The typical knowledge base used for this task is Wikipedia. One underlying assumption that makes Wikipedia attractive for this task is that phrases that are good keyphrases in general, are used in Wikipedia articles as texts of hyperlinks. This property has been called keyphraseness [49, 87, 90].

In order to score text phrases from documents based on their properties derived from Wikipedia, the first stage involves linking and disambiguation [49, 87]. Afterwards, the linked concepts are scored, in a supervised [87] or unsupervised manner [49]. Besides the keyphraseness, the methods also use properties extracted from the analysis of the hyperlink graph of Wikipedia.

A very recent work [121], done in parallel with ours, extracts the most representative DBpedia concepts representing discussion fora. Rowe et al. [121] analyse community generated content like forum posts, in order to gain insight into the topical specificity of communities. They extract from the posts the DBpedia concepts with Zemanta\(^2\). These concepts are weighted based on their frequency in the forum, or on a weighting scheme similar to \(tf-idf\), that the authors name concept frequency-inverse forum frequency (cf-iff). The top concepts as ranked with these schemes are then used to represent the forum.

If used for topic labelling, all these methods are limited to the concepts that the phrases of the document are directly linked to. As such, despite using Wikipedia as external knowledge base, if used for topic labelling, these methods still suffer from the drawback of assuming that the input document contains the keyphrases that are best for labelling it. In Chapter 6, we compute the frequency with which suitable ground-truth labels occur in the actual set of disambiguated concepts of the document, as this is the upper bound of labelling accuracy achievable by using keyphrase extraction. We greatly improve this upper bound by extracting

---

\(^2\) http://www.zemanta.com
label candidates from the knowledge base as concepts that are semantically related to the document concepts.

### 3.2.3. Label Finding in External Knowledge Bases

Many works that attempt labelling texts (document clusters, documents, topics, posts) including the one we propose in Chapter 6, are not restricting the space of possible labels to the concepts named in the documents. They all have in common the fact that they look for labels outside the borders of their own content, by scoring candidate labels from external knowledge bases.

WordNet has been used by extracting all hypernyms of the topic words [66] and this method obtained very poor results in the published comparative study. Open Directory Project\(^3\) was used for topic-labelling by Magatti et al. [85]. This approach differs from our work and the aforementioned ones by relying on the tree-structure of Open Directory Project. Magatti et al. [85] model each cluster of the hierarchy as a list of words and use the names of the clusters as label candidates for topics. They compare the topics to label with the nodes in the hierarchy based on various relatedness measures and reuse the label of the most similar node in the hierarchy. Magatti et al.’s approach is particularly suited for use-cases providing a given hierarchy that has to match the clusters of the corpus.

A multitude of approaches tackled the problem of topic or document labelling with Wikipedia, and we focus on them in the remainder of this section. Most methods that use Wikipedia’s full text as a knowledge base for labelling extraction [23, 74, 75] usually employ a full text index of Wikipedia, and the titles of Wikipedia articles and categories are the label candidate pool. Nomoto [108] take a distinct approach, as they de-construct the full text of each Wikipedia article into a collection of minipages corresponding to their sections. The sections are indexed and it is their headings that form the pool of candidate labels.

Having such an index the methods proceed by querying it with the set of target words (the topic words, or cluster representative words), to retrieve the set of relevant entries. Nomoto [108] use the headings of the top retrieved sections as labels. Carmel et al. [23] and Lau et al. [75] use the retrieved Wikipedia articles for further scoring. Carmel et al. [23] for example use statistical methods similar to Mei et al.’s scoring function, Formula (3.7).

Lau et al. [75] use the article titles and their main noun-phrases, as candidate labels. For each candidate label in the final set, they compute an association score to the topic terms.

\(^3\)[http://www.dmoz.org]
Several measures are used like pointwise mutual information, t-test, Dice coefficient, either independently in a unsupervised manner, or combined in a supervised machine learning system that learns each method’s weight. The authors compare these methods to Mei et al.’s text-based method [88], and the unsupervised methods achieve worse results than Mei et al.’s.

Other approaches use the Wikipedia content as well as the hyperlink graphs [30, 131]. Both approaches consider the task of document labelling. They have a similar starting point as the previous approaches: to identify the Wikipedia articles that are most representative for the document. This is achieved either by querying a Wikipedia full text index with the text of the document [131], or by linking and disambiguating the document phrases to Wikipedia [30]. The obtained Wikipedia articles, that we call “seeds” are used to extract the hyperlink graph connecting them on Wikipedia. The graph is then used by Syed et al. [131] for spreading activation from the seeds. The activation spread from the source article to the sink article increases as the cosine similarity between the two Wikipedia articles increases.

Coursey et al. [30] use the graph by running Personalised PageRank (PPR) [53] with the probability mass distributed over the seeds [30]. The probability of these articles is influenced by their keyphraseness and distance from the root of the Wikipedia Categories hierarchy, http://en.wikipedia.org/wiki/Category:Fundamental. The articles and categories with highest PPR score obtained are then considered the most relevant with respect to the input document.

From a methodology perspective, our work is closely related to Coursey et al.’s [30], as we also perform disambiguation followed by graph centrality analysis for label extraction. However, all the aforementioned approaches using Wikipedia strongly differ from our approach, as they analyse the content of Wikipedia articles in order to decide on the proper label. This makes the algorithms hard to adapt to data sources that are less rich in content and do not contain encyclopedic text about concepts. Our approach is fully structured and independent of the content of Wikipedia articles. It differs from all the above works from three perspectives: First, it uses only structured data in order to identify the labels. Second, the analysed graphs are not pre-processed off-line. Thus, it can be used entirely on-line by querying knowledge bases, such as the DBpedia SPARQL endpoint. Third, for identifying suitable labels, we adapt and experiment with popular graph-based centrality measures that have not been used before for this task.

---

4 this Wikipedia Category has been deleted in 2010.
5 http://dbpedia.sparql.endpoint
Part II.

Core
Chapter 4.

Analysis of On-the-fly Graphs for Gloss-based WSD

While gloss-based approaches to word-sense disambiguation take advantage neither of the vast amount of structured knowledge that is currently available, nor of the encyclopedic knowledge of Wikipedia, they are portable and independent of the sense inventory. This property makes them desirable when multiple datasets provide candidates for disambiguation, when interlinking resources from different knowledge bases, and also when the structure of the targeted knowledge base is not rich enough to provide good disambiguation cues.

This chapter presents a novel approach to unsupervised, gloss-based word-sense disambiguation. It starts by formalising the problem, and then by describing the suggested approach. Afterwards, in Section 4.2 we present the first core contribution of this work, a novel measure for quantifying relatedness between multiple senses. Section 4.3 describes the second core contribution of this chapter, an innovative gloss-based sense model that adapts to the other considered senses, in order to emphasize relevant gloss-words. Then, in Section 4.4, we detail the experiments we conducted and show that our proposed algorithm strongly surpasses the performance of state-of-the-art gloss-based methods. We finally conclude in Section 4.5.

4.1. Introduction

In this chapter, we identify and exploit some opportunities for improving the performance of unsupervised, gloss-based word-sense disambiguation. As previously stated in Section 3.1.1,

\footnote{Parts of the research reported in this chapter have been published as [63]}
gloss-based WSD denotes the class of WSD approaches that use the glosses of senses as the only sense description and evidence for their meaning.

The contribution of the work presented in this chapter is multi-fold. First, we propose an eigenvalue-based measure, E-WSD, inspired by HITS [70], to measure the coherence of a group of senses for achieving global disambiguation. Previous global optimisation approaches, presented in Chapter 3, typically use the sum of pairwise relatedness scores as a measure of coherence for a group of senses. E-WSD is an alternative measure we study because it is able to capture more varied and subtle properties of the relations between multiple senses than the sum of their pairwise relatedness scores. Second, we propose a novel sense modelling approach, called adaptive sense vectors (ASV), that scores the relevance of gloss-words used in the glosses of the senses.

Third, we base our approach for WSD on two inventories: WordNet [92] and DBpedia\(^2\). Previous work that considers simultaneous use of DBpedia concepts and WordNet senses usually start with a preliminary step of integrating the two sense inventories. These methods therefore produce a new knowledge base like YAGO [58] and BabelNet [103]. While this expensive step brings the benefit of producing a new knowledge base, it is not practical for scenarios where linking to various inventories is the main goal. We regard this scenario highly relevant in the context of Linked Data. In this work, we therefore propose to use gloss-based WSD in order to simultaneously link to different inventories that are not necessarily integrated.

In this case, an important challenge is that inventories differ in terms of the length of their sense glosses. For instance, DBpedia glosses are longer than WordNet glosses, and we show that this difference in gloss lengths of the two sense inventories generates biases either towards long, or short glosses, depending on the used gloss similarity measure. We reduce this bias by proposing a simple heuristic that is easily integrated in the ASV model. We show that the eigenvalue-based measure of coherence for a group of senses, when paired with the proposed sense modelling scheme, strongly improves the current state-of-the-art. We also show that ASV can be used with success independently of our eigenvalue-based measure.

Another novel aspect of our work is that we exploit WSD via topic-models [126]. This helps to reduce the complexity of WSD tasks for large text corpora. Although topic models have been used for document representation and extraction of related words for many years, word-sense disambiguation focuses mainly on disambiguation of words in the context of phrases. We show that the context created by a topic can be used to improve disambiguation performance, by

\(^2\)http://dbpedia.org
amplifying the relatedness between target words appearing in the same context. Our evaluations are performed on ground truth data collected via a two week user study.

4.1.1. The Joint WSD Problem

Let us consider the set of \( n \) words that are targeted by joint word-sense disambiguation, \( W = \{w_1, \ldots, w_n\} \). A set \( M_i \) of \( m_i \) possible senses corresponds to every word \( w_i \). We refer by \( s_{i,k} \) to the \( k^{th} \) sense of word \( w_i \), so that \( M_i = \{s_{i,1}, \ldots, s_{i,m_i}\}, i \in \{1, \ldots, n\} \). In a joint WSD problem, the space of feasible solutions is formed by all the possible sense combinations, taking exactly one sense for each target word. Therefore, if \( FS \) denotes the set of feasible solutions, there are \( |FS| = \prod_{i=1}^{n} m_i \) feasible solutions.

**Definition 1.** Given \( n \) words to be disambiguated, with \( m_i \) possible senses each, \( i \in \{1, \ldots, n\} \), we define a feasible solution \( FS \), as a combination of possible senses, containing exactly one possible sense for each target word.

\[
FS = \{s_{i,j} | s_{i,j} \in M_i, \forall i \in \{1, \ldots, n\} \land j \in \{1, |M_i|\} \land \exists s_{a,b}, s_{c,d} \in FS, \text{ s.t. } a = c\}, \text{ and } FS = \bigcup\{FS\}
\]

**Example 2.** Let us consider four words: \( W = \{A, B, C, D\} \), with the following possible senses: \( M_A = \{s_{a,1}, s_{a,2}, s_{a,3}\}, M_B = \{s_{b,1}, s_{b,2}\}, M_C = \{s_{c,1}, s_{c,2}, s_{c,3}\} \) and \( M_D = \{s_{d,1}, s_{d,2}\} \) Then for WSD, the feasible solutions set \( FS \) contains 36 feasible solutions:

\[
FS = \{\{s_{a,1}, s_{b,1}, s_{c,1}, s_{d,1}\}, \{s_{a,1}, s_{b,1}, s_{c,2}, s_{d,1}\}, \{s_{a,1}, s_{b,1}, s_{c,3}, s_{d,2}\}, \ldots\}.
\]

The problem of joint WSD is to find the feasible solution \( FS^* \) that maximises a function \( f(FS) \) over the set of feasible solutions:

\[
FS^* = \arg \max_{FS \in FS} f(FS) \tag{4.1}
\]

As WSD is an optimisation problem, the function \( f : FS \rightarrow \mathbb{R}^+ \) is from this perspective the objective function. From the word-sense disambiguation perspective, it has been called coherence [59, 72], as it measures the coherence of the senses in the analysed solution.

In this chapter, we look into gloss-based disambiguation, therefore the senses are represented by their glosses. As such, the feasible solution \( FS \) can be seen as a set of glosses and the function \( f(FS) \) must capture its coherence.
4.1.2. Approach Overview

The E-WSD approach, that is the focus of this chapter, can be summarised as follows:

**Sense Model** Senses are multi-dimensional vectors, where the dimensions correspond to words extracted from their glosses. However, only the dimensions corresponding to gloss-words that overlap with the other senses in the same feasible solution have a non-zero value. Section 4.3.1 describes in detail the gloss-word weighting scheme. For illustration, we represent feasible solutions as bipartite graphs connecting senses to their gloss words. Hence, senses are nodes in these bipartite graphs, and the weights of the gloss-words with respect to senses are represented as weighted edges.

**Context Model** Words are jointly disambiguated in the context of topic models. To the best of our knowledge, this is the first work that attempts disambiguation of words in LDA topic models. We experiment with various sizes of context, taking the top-k words from topics, based on their probability, and analyse the effect of the context size in Section 4.4.5.

**Semantic Relatedness** Pairwise semantic relatedness is implicitly calculated in the computation of the hubs matrix, and it corresponds to the scalar product between the senses.

**Disambiguation** For all feasible solutions, the hubs matrix of the bipartite graph containing senses and their gloss-words is computed. Then, the principal eigenvalue of this matrix is computed, and the solution with highest eigenvalue is returned as the optimal solution. The rationale behind this computation is detailed in Section 4.2.

We now move on to introducing the main idea and computation of the HITS [70] algorithm that inspired this work. Afterwards, we show how the intuitions behind this algorithm resonate with the intuitions behind gloss-based joint WSD, and propose an eigenvalue based measure for the coherence of a WSD solution. We then propose a novel sense modelling approach, called adaptive sense vectors (ASV), that weights senses and their gloss-words with respect to the feasible solution they are part of. We use this model together with a named entity-based heuristic for capturing the importance of gloss-words with respect to the disambiguation solution. Finally, we report the main findings from the experiments we conducted, in order to understand the performance of the eigenvalue-based coherence computation, and of the adaptive sense vectors.
4.2. Eigenvalue-Based Word Sense Disambiguation

In our gloss-based disambiguation problem, we can represent the relation between a sense and the words contained in its gloss, as directed edges pointing from the sense to the gloss words. Then we can represent a feasible solution as a bipartite graph as in Figure 4.1.

Let $A$ be the adjacency matrix corresponding to the bipartite graph of the feasible solution. $A_{ij}$ is equal to 1 if the gloss of sense $s_i$ contains the gloss-word $\omega_j$, or 0 otherwise. The fact that the glosses of related concepts have words in common has been used as a cue for WSD since 1986, by Lesk [80]. The strength of the relation between two concepts can be measured by counting their common gloss-words. Furthermore, a sense is important if it contains many gloss-words that also belong in the glosses of many other senses. Most gloss-based WSD approaches [31, 80, 123] use this intuition.

In this work, we extend this intuition and claim that two gloss-words $\omega_i$ and $\omega_j$, are related if they co-occur in the definitions of many concepts. The strength of this relation can be seen as the number of concepts which use both gloss-words. Overall, a feasible solution is better if it contains glosses that contain many common gloss-words that are themselves related. This mutually recursive relation between glosses and gloss-words resonates with the HITS algorithm, and forms the basis of our eigenvalue-based function for solution coherence.

4.2.1. The HITS Algorithm

The HITS (Hyperlink-Induced Topic Search) algorithm, also called Hubs and Authorities, was introduced in 1998 by Kleinberg [70]. It has been proposed as a solution for identifying the most authoritative web pages with respect to a given query. It defines an authority page as a page which is pointed to by other pages, and a hub page as a page which points out to other pages. The intuition is that a good authority page is one which is pointed to by many good
hub pages. Formally, let us consider the directed graph $G(V,E)$, represented by the adjacency matrix $A$, and that the notation $(p,q)$ represents an edge in $E$ pointing from node $p$ to node $q$. HITS starts by assigning each node in the graph an authority and a hub weight. In the first step, all these weights are assigned to 1. Afterwards, the algorithms iteratively updates the two weights of each node in a mutual recursion operation:

$$a_{p} \leftarrow \sum_{q:(q,p) \in E} h_{q}, \quad h_{p} \leftarrow \sum_{q:(p,q) \in E} a_{q}, \quad (4.2)$$

After each iteration the weights are normalised, by dividing the authority weights to the magnitude of the authorities vector, and the hubs weights to the magnitude of the hubs vector:

$$a_{p} \leftarrow \frac{a_{p}}{||a||}; \quad h_{p} \leftarrow \frac{h_{p}}{||h||}. \quad (4.3)$$

Using the adjacency matrix $A$ of graph $G(V,E)$, and applying the two operations for $k$ times, we obtain:

$$a^{(1)} = \phi_{1} A^\top h^{(0)} \text{ by (4.2);} \quad h^{(1)} = \varphi_{1} A a^{(1)} \text{ by (4.2)} \quad (4.4)$$
$$a^{(k)} = \phi_{k} \varphi_{k-1} A^\top A a^{(k-1)}; \quad h^{(k)} = \varphi_{k} \phi_{k} A A^\top h^{(k-1)}. \quad (4.5)$$

where $\phi_{i}$ is the normalisation constant for the authorities vector at iteration $i$, and $\varphi_{i}$ is the hubs vector normalisation constant at iteration $i$.

The two vectors converge to $a^*$ and $h^*$, the principal eigenvectors of $A^\top A$ and $AA^\top$, respectively:

$$a^* = \frac{1}{\lambda^*} A^\top A a^*; \quad h^* = \frac{1}{\lambda^*} A A^\top h^*. \quad (4.6)$$

The matrix $A^\top A$ is often referred to as the authorities matrix, and the matrix $AA^\top$ as the hubs matrix. These matrices are both symmetric and they have the same eigenvalues. Furthermore, $\lambda^*$ is the principal eigenvalue of $A^\top A$ and of $AA^\top$ [37]. From Formulas (4.3), (4.5) and (4.6), $\lambda^*$ increases as the magnitude of the authorities vector increases and as the magnitude of the hubs vector increases. Therefore the higher the value of $\lambda^*$, the more connected the bipartite graph $G$. In this work, we exploit this property in order to identify the most densely linked combination of senses and their corresponding gloss-words.
4.2.2. Eigenvalue-based WSD Solution Coherence

In the context of a feasible solution, word-senses are the hubs, and the gloss-words are the authorities. Given a feasible solution, we are interested to score how inter-connected its senses and gloss-words are. Our intuition is that the magnitude of the principal eigenvalue of the hubs and authorities matrices of this graph can be used as a score for the feasible solution.

The hypothesis we formulate and test in this chapter is that the WSD problem can be solved by selecting, out of the space of feasible solutions, the one whose hubs matrix has the highest principal eigenvalue. Equation (4.7) formalises the problem.

$$FS^* = \arg \max_{FS \in FS} \text{eig}_1(A_{FS}A_{FS}^T)$$

where $\text{eig}_1 : \mathbf{R}^{n \times n} \rightarrow \mathbf{R}$ denotes the function computing the principal eigenvalue of a matrix, and $A_{FS}$ is the adjacency matrix of the bipartite graph that represents the feasible solution $FS$. In addition, this method also supports edge-weighted bipartite graphs. In the bipartite graphs between word-senses and gloss-words, edge weights can be used to emphasize the relation strength between a sense and a gloss-word. We exploit this by proposing in Section 4.3, a method for capturing the relevance of gloss-words to senses, and show that it further improves the disambiguation performance. In the following, we refer to the method of using the principal eigenvalue of the hubs matrix for WSD, as E-WSD.

4.2.3. Theoretical Comparison to Sum of Overlaps

As shown in the previous section, the eigenvalue of the hubs matrix might be used as a measure of collective sense relatedness. The measure from related work that E-WSD is closest to, is the sum of scalar products (SSP). In the case of unweighted relations between senses and gloss-words, leading to $A_{ij} \in \{0, 1\}$, the sum of scalar products is equal to the sum of pairwise gloss-words overlap. Cowie et al. [31] firstly used the sum of pairwise gloss-words overlap between senses in order to score feasible solutions. Its weighted generalisation, namely the sum of scalar products, has later been used in WSD with Wikipedia [33, 72]. Figure 4.2 shows that in its unweighted form, eigenvalue actually captures better the subtleties of the relatedness between multiple senses.
Figure 4.2 shows four types of bipartite graphs, all containing three senses. In all cases, the senses have the same sum of pairwise word overlap. However, the principal eigenvalues of their hubs matrices differ, so that they capture the magnitude of the feedback effect between hubs and authorities. From a WSD perspective, the eigenvalue measure scores higher the combination of senses from Figure 4.2(a) than Figure 4.2(b) because senses \( s_a \) and \( s_b \) have two words in common rather than just one. Actually, as seen in Figure 4.2(c), an overlap of three words between two senses, is already producing a higher eigenvalue than both Figure 4.2(a) and 4.2(b), even if sense \( s_c \) is completely isolated. In this work, we exploit to what extend capturing these network subtleties can help in solving the WSD problem.

<table>
<thead>
<tr>
<th>Senses-words bipartite graph</th>
<th>( A )</th>
<th>( AA^T )</th>
<th>Measure</th>
</tr>
</thead>
</table>
| (a) | \[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\] | \[
\begin{pmatrix}
3 & 2 & 1 \\
2 & 2 & 0 \\
1 & 0 & 1
\end{pmatrix}
\] | \( \text{sum} = 3; \) \( \lambda_1 = 4.7321; \) |
| (b) | \[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 1
\end{pmatrix}
\] | \[
\begin{pmatrix}
2 & 1 & 1 \\
1 & 2 & 1 \\
1 & 1 & 2
\end{pmatrix}
\] | \( \text{sum} = 3; \) \( \lambda_1 = 4; \) |
| (c) | \[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
0 & 0 & 0
\end{pmatrix}
\] | \[
\begin{pmatrix}
3 & 3 & 0 \\
3 & 3 & 0 \\
0 & 0 & 0
\end{pmatrix}
\] | \( \text{sum} = 3; \) \( \lambda_1 = 6; \) |
| (d) | \[
\begin{pmatrix}
1 & 1 & 1
\end{pmatrix}
\] | \[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{pmatrix}
\] | \( \text{sum} = 3; \) \( \lambda_1 = 3; \) |
Nevertheless, the graph in Figure 4.2(d) is of particular interest. The small degrees of the senses lead to poor principal eigenvalue. A feasible solution having the graph pattern in Figure 4.2(d) contains a gloss-word that is common to all the targeted senses. This is usually a very strong indication that the three senses are all related, not just pairwise related. In other words, in WSD feasible solutions, a gloss-word that is common to multiple senses (i.e., has high in-degree) suggests stronger solution coherence than a sense with multiple common gloss-words (i.e., a sense with high out-degree). This is simply because senses might have many gloss-words in common by virtue of having long glosses. A straightforward solution that comes to mind is to normalise the scores based on gloss lengths, and therefore use the cosine similarity rather than the scalar product. But as we show in the evaluation section, this strategy fails to solve the problem, because the length of senses is not the core issue to target.

We show in the remainder of this chapter that a problem of uttermost importance when scoring feasible solutions is that the gloss-words that senses have in common are relevant to the gist of the feasible solution. The use of the principal eigenvalue addresses this problem, but we further stress the importance of gloss-words relevance by proposing the adaptive sense model (ASV), that we present in the next section.

### 4.3. Adaptive Sense Modelling for WSD

#### 4.3.1. Adaptive Sense Vectors (ASV)

Until now, we have considered the bipartite graph representing a feasible solution as unweighted. This means that all gloss-words represented in a feasible solution are equally relevant. However, our intuition is that the gloss-words that belong to many glosses in the feasible solution are more relevant. We illustrate this intuition with the example in Table 4.1, where we want to disambiguate the three target words in the topic: [web, internet, page]. web#1 denotes the 1st sense of word web and so on, and the right column contains the terms extracted from the glosses of the senses.

For humans it is not hard to recognise that web#1, internet#1 and page#1 are the most likely senses of the target words. However, the gloss of page#1 is much shorter compared to the glosses of page#2 and page#3, and it only has one gloss-word, web, in common with the senses web#1 and internet#1. The gloss of sense page#2 has two words in common
Table 4.1: A toy example for three target words, with a set of candidate meanings and their respective related words.

<table>
<thead>
<tr>
<th>Word</th>
<th>Sense</th>
<th>Gloss words</th>
</tr>
</thead>
<tbody>
<tr>
<td>web</td>
<td>web#1</td>
<td>web, internet, document, contain, hypertext</td>
</tr>
<tr>
<td></td>
<td>web#2</td>
<td>web, feather, net, flat, side</td>
</tr>
<tr>
<td></td>
<td>web#3</td>
<td>web, world, english, bible, book</td>
</tr>
<tr>
<td>internet</td>
<td>internet#1</td>
<td>internet, connected, web</td>
</tr>
<tr>
<td>page</td>
<td>page#1</td>
<td>page, web, browser</td>
</tr>
<tr>
<td></td>
<td>page#2</td>
<td>page, paper, leaf, side, contain, document, book</td>
</tr>
<tr>
<td></td>
<td>page#3</td>
<td>page, boy, work, servant, service</td>
</tr>
</tbody>
</table>

\[ \lambda_{AA}^T = 4.56 \quad \text{SSP} = 4 \]

\[ \lambda_{AA}^T = 6 \quad \text{SSP} = 4 \]

(a) Bipartite graph of feasible solution \{web#1, internet#1, page#1\}
(b) Bipartite graph of feasible solution \{web#1, internet#1, page#2\}

Figure 4.3: Top two feasible solutions of the WSD problem in Table 4.1

with the gloss of web#1. The graphs of the two feasible solutions of interest are shown in Figure 4.3.

As seen in Figure 4.3, the sum of scalar products (SSP) cannot discriminate between the two feasible solutions, while the eigenvalue of the hubs matrix scores the feasible solution with more gloss-words higher. Therefore, we recognise the need of a score that emphasises the fact that the gloss-word web in the feasible solution in Figure 4.3(a) belongs to all three senses. We consider that a gloss-word’s relevance to the coherence of the solution increases as the number of glosses it belongs to increases.

We achieve this by weighting the bipartite graph’s edges. In feasible solution \( FS \), we emphasise gloss-words’ relevance by making the weights of their incoming edges linear with
their in-degree. As such, the weight of an edge from sense \( i \) to its gloss-word \( j \) is weighted as:

\[
\text{weight}^{FS}_{(i,j)} = \text{indegree}^{FS}(j) - 1
\]  

We subtract 1 in Formula (4.8) because we only consider gloss-words that belong to at least two senses, therefore for gloss-words that belong to only one sense, the incoming edge weights 0.

From a vector-space model perspective, given a feasible solution \( FS \), and sense \( i \) belonging to \( FS \) and having \( t \) gloss-words, we define the adaptive sense vector representation (ASV) of sense \( i \) as:

\[
\text{ASV}^{FS}_i = [f^{FS}(\omega_1) - 1, f^{FS}(\omega_2) - 1, ..., f^{FS}(\omega_t) - 1]
\]  

where \( f^{FS}(\omega_1) \) denotes the number of senses of \( FS \) that the gloss-word \( \omega_1 \) belongs to.

Using the toy example from Table 4.1, we illustrate the effects of this weighting scheme in Figure 4.4. Figure 4.4 shows the bipartite graphs of the 9 feasible solutions, and their coherence as scored by the principal eigenvalue of the hubs matrix, \( \lambda_{WW^T} \), and the sum of scalar products, \( SSP \). The correct meanings for the two ambiguous target words \( web \) and \( page \) are \( web\#1 \) and \( page\#1 \) respectively, therefore the feasible solution \( A \) is correctly selected.

Table 4.2 shows the ASVs of the sense \( web\#1 \) as represented in the three different feasible solutions this sense is part of.
Figure 4.4.: Bipartite graphs generated by each combination of meanings in the toy disambiguation example in Table 4.1, with the corresponding eigenvalue scores ($\lambda$).

4.3.2. Using ASV with Heuristics

In this work, we consider overlapping named entities referred to in the glosses as stronger evidence for senses’ relatedness than other words. This heuristic is close to that of Banerjee and Pedersen [9], who considered the length of the overlap as the stronger evidence. We use our heuristic in order to devise an edge weighting scheme for the bipartite graph, with the support of ASV. The purpose of our scheme is to penalise senses whose glosses refer to many named entities that do not overlap with the glosses of the other senses in the same feasible solution.

To this end, we introduce sense weights. We then make the weight of each sense whose gloss contains many named entities, linear to the proportion of named entities it shares with the other glosses in the current feasible solution. Thus, if $NE_i$ denotes the set of named entities that the gloss of sense $i$ contains, and $t$ is a predefined threshold, the weight of sense $i$ is calculated as:
Analysis of On-the-fly Graphs for Gloss-based WSD

\[ \text{weight}^{FS}_i = \begin{cases} \frac{|NE_i| \cap (\bigcup_{j \neq i} NE_j)|}{|NE_i|} & \text{if } |NE_i| \geq t \\ 1 & \text{otherwise} \end{cases} \] (4.10)

Therefore weight\(^{FS}_i \in [0, 1]\. In other words, if the number of named entities contained in the gloss of sense \(i\), is bigger than \(t\), then sense \(i\) is penalised so that its weight is equal to the proportion of named entities that are also shared by other glosses in the same feasible solution. The actual penalty equals \(1 - \text{weight}^{FS}_i\). Since the fundamental idea behind ASV is that gloss-words are relevant to the solution coherence if they belong to multiple glosses, we introduce a weighted in-degree of gloss-words:

\[ \text{weighted}_{indegree}^{FS}(\omega_j) = \sum_{(i,j) \in E} \text{weight}^{FS}_i; \] (4.11)

Using the weighted in-degree of gloss words, and the weights of senses, we generalise Formula (4.8), and obtain the generalised weight of an edge \((i,j)\), denoted by \(\text{\hat{weight}}^{FS}_{(i,j)}\) as:

\[ \text{\hat{weight}}^{FS}_{(i,j)} = \text{weight}^{FS}_i \times (\text{weighted}_{indegree}^{FS}(\omega_j) - \text{weight}^{FS}_i), \] (4.12)

which is equivalent to:

\[ \text{\hat{weight}}^{FS}_{(i,j)} = \text{weight}^{FS}_i \sum_{(k,j) \in E, k \neq i} \text{weight}^{FS}_k; \] (4.13)

Then, rewriting Formula 4.9 we obtain the adaptive sense vector representation of sense \(i\) as:

\[ \text{ASV}^{FS}_i = [\text{\hat{weight}}^{FS}_{(i,1)}, \text{\hat{weight}}^{FS}_{(i,2)}, ..., \text{\hat{weight}}^{FS}_{(i,t)}]; \] (4.14)

Figure 4.5 shows examples on how sense weights influence the weights of graph edges, and finally the hubs matrix eigenvalue. Figure 4.5(a) shows how the weight for the sense \(s_c\) is propagated to the edges in the graph. Figure 4.5(b) shows the case when all weights are 1, while in Figure 4.5(c) the sense \(s_c\) has a 0.8 weight. Finally, Figure 4.5(d) shows the case when the weight is 0, effectively removing the sense from the computation. An important aspect
which makes this strategy applicable in our approach is that a sense will only be penalised in a
group of senses it shares very few entities with. At the same time, in another combination of
senses, the same sense might be better connected and not attract any penalty.

We conclude this section with an example that illustrates how the adaptive sense modelling,
together with the named entity based weighting scheme manage to weight the gloss words of a
gloss differently, depending on the analysed feasible solution.

**Example 3.** Consider two groups of words each needing joint disambiguation, one being from
the physics domain \([gravity, speed, weight]\), while the other being from the cinema
domain \([gravity, speed, sandra bullock]\). Consider the tokenized glosses of
their possible senses as shown in Table 4.3. Note that the gloss word gravity of senses \(g\#1\)
and \(w\#1\) is a common noun, while gloss-word Gravity in sense \(g\#2\) is a named entity, as
corresponds to the name of the film. As such, gravity and Gravity(NE) are considered two
different tokens. The same holds for gloss-words speed and Speed(NE).

<table>
<thead>
<tr>
<th>Target word</th>
<th>Sense</th>
<th>Gloss words</th>
</tr>
</thead>
<tbody>
<tr>
<td>gravity</td>
<td>(g#1)</td>
<td>gravity, object, mass, physics, ...</td>
</tr>
<tr>
<td></td>
<td>(g#2)</td>
<td>Gravity(NE), film, producer, screenwriter, space, Sandra Bullock(NE), Alfonso Cuaron(NE), George Clooney(NE),...</td>
</tr>
<tr>
<td>speed</td>
<td>(s#1)</td>
<td>speed, object, velocity, ...</td>
</tr>
<tr>
<td></td>
<td>(s#2)</td>
<td>Speed(NE), film, producer, screenwriter, Sandra Bullock(NE), Keanu Reeves(NE), Dennis Hopper(NE),...</td>
</tr>
<tr>
<td>weight</td>
<td>(w#1)</td>
<td>science, physics, gravity, force, mass ...</td>
</tr>
<tr>
<td>Sandra Bullock</td>
<td>(SB#1)</td>
<td>Sandra Bullock(NE), film, producer, star, act, Gravity(NE), Keanu Reeves(NE), George Clooney(NE), US(NE), Crash(NE) ...</td>
</tr>
</tbody>
</table>

**Table 4.3:** A toy example for three target words, with a set of candidate meanings and their respective related
words. The NE tag means the corresponding word is a named entity.
Analysis of On-the-fly Graphs for Gloss-based WSD

Target words gravity and speed require disambiguation, and in the physics topic the correct senses are g#1 and s#1, while in the cinema topic the correct senses are g#2 and s#2 corresponding to the two films. By considering weight 1 of the senses, even in the physics topic, the two ambiguous words would be disambiguated to g#2 and s#2 due to their high overlap. But by using the weighting scheme as described in Formula (4.13), in the feasible solution {g#2, s#2, w#1}, the senses g#2 and s#2 are assigned 0.25 weight each. Then, the vectors of the three senses in the feasible solution {g#2, s#2, SB#1} are as shown in Table 4.4(a), leading to a very small score as computed by E-WSD of 0.0313. For this topic, the feasible solution {g#1, s#1, w#1} does not attract any penalty, and obtains a score of 6.64. The other two feasible solutions for this topic are also penalised and obtain lower scores.

Because we consider the senses’ weights proportional to the number of named entities they share, the senses g#2 and s#2, when used in the cinema topic, in the feasible solution {g#2, s#2, SB#1}, have higher weights as shown in Table 4.4(b). These weights lead to a E-WSD score of 6.85 for this feasible solution. In the same cinema topic, the feasible solution {g#1, s#1, SB#1} is penalised because the sense SB#1 obtains a very low weight. The other two feasible solutions {g#1, s#2, SB#1} and {g#2, s#1, SB#1} also obtain poorer scores than the correct combination represented in Table 4.4(b).

To sum up, the adaptive sense vectors adapt the weights of their gloss-words with respect to the feasible solution they are part of. This schema therefore places emphasis on the coherence...
of the feasible solution as a whole. As such, the model of senses depends on the feasible solution they are part of, and consequently the relatedness measure between two senses depends on the feasible solution where the senses are considered.

In the following section we describe the experiments we carried out for evaluating our methods, the results we obtained and we discuss the effects of all our decisions in detail.

4.4. Evaluation

In this evaluation, we follow several objectives:

- The first objective is to assess the effectiveness of the eigenvalue-based measure of coherence, and of the adaptive sense vectors. We evaluate their performances separately as well as together, and compare them to implementations of multiple related work approaches;

- The second objective is to verify if topics obtained through probabilistic topic modelling provide good disambiguation context;

- The third objective is to simultaneously use two sense inventories, and understand how this affects our disambiguation methods as compared to related work methods;

- The fourth objective is to show that global disambiguation is more effective than local-extended disambiguation, although the latter is by far the most popular strategy;

In order to attain our first objective, we have implemented three disambiguation methods based on the contributions of this chapter: Basic E-WSD which used the eigenvalue-based coherence score, without ASV, ASV+SSP which models senses as adaptive sense vectors, and uses them in combination with sum of scalar products as a coherence score, and third ASV+E-WSD, which models senses as adaptive sense vectors and uses the eigenvalue-based measure of coherence. Besides these three methods, we also implement five methods inspired from related work.

In order to meet the second objective, we ran the Latent Dirichlet Allocation (LDA) topic modelling algorithm and we used for disambiguation a selection of the resulted topics. We meet the third objective by simultaneously using WordNet and DBpedia senses for the disambiguation of words in the topics. As for the last objective, we compare ours, as well as other global disambiguation methods to a local-extended approach that uses the same pairwise relatedness measure.
We base all our assessments and results on ground-truth data obtained through a user study for WSD. We now describe the datasets, experiment settings, the user study and methods in more detail, and then we get to reporting and interpreting the results.

### 4.4.1. Datasets

Since the Senseval³ corpora for WSD do not contain annotations for topic models, we organised a user study to gather ground truth benchmark data for WSD. In order to generate the data, we ran LDA [86] on three corpora: (i) British Academic Written English Corpus (BAWE) [105], (ii) BBC [45], and (iii) Semantic Web Corpus ⁴. We extracted 500 topics, 250 topics, and 100 topics from these three corpora respectively. We randomly selected a subset of 130 topics: 70 from BAWE, 40 from BBC and 30 from the Semantic Web Corpus, to match the feasibility of a user study.

### Sense Inventories

In order to evaluate this work, we used two sense inventories, WordNet version 2.1 and DBpedia.

### Possible Sense Extraction

For using WordNet, we downloaded⁵ and installed it locally, and for all target words we retrieved the definition, synonyms, hypernyms, and meronyms, if available. As for extracting possible senses from DBPedia, we used the DBpedia Lookup Service⁶, YAGO [58], and the DBpedia SPARQL Endpoint⁷. For each target word, queries were issued automatically. For the DBpedia Lookup Service, the query URL we used was http://lookup.dbpedia.org/api/search.asmx/KeywordSearch?QueryString=query&MaxHits=15, with the query text set to be the target word. We extracted the top 15 results. As for the DBpedia SPARQL Endpoint, direct SPARQL queries for labels could not be used because the endpoint would timeout. Therefore, we used YAGO⁸ [58] as a workaround. We downloaded

³http://www.senseval.org/
⁴http://data.semanticweb.org
⁵http://wordnet.princeton.edu/wordnet/download/
⁶http://wiki.dbpedia.org/lookup
⁷http://dbpedia.org/sparql
the YAGO knowledge base in its Jena TDB\(^9\) format. Jena TDB is a RDF store that supports the Jena API. Using the Jena API for Java, we queried the YAGO TDB repository to extract the YAGO entities that have the target word as label. For the extracted YAGO entities, we retrieved the corresponding Wikipedia article, because YAGO, being automatically created from parsing Wikipedia and WordNet, contains entities that correspond to most Wikipedia articles. Then we used the obtained Wikipedia article to query the DBpedia SPARQL Endpoint and retrieve the corresponding DBpedia entity.

For each retrieved DBpedia resource, if it points to a redirect concept, then we use that concept only if it has the same label as the targeted word. If the DBpedia resource points to disambiguation concepts, we extract those concepts, and select only the ones that have the same label as the target word. For all the remaining DBpedia concepts, we extract the abstract by querying for the `dbont:abstract` property, as well as the DBpedia categories and classes it belongs to.

All the fields collected about each sense are then passed through a text processing phase consisting of stopword removal, word stemming, 2-gram and 3-gram extraction, and named entity extraction \(^3\). For named-entity extraction we used the Stanford CoreNLP\(^10\) toolbox. Then we aggregated all the resulted tokens, obtaining a bag-of-words representation for each sense. We saved the topics, their words, their senses and corresponding bags-of-words to a MySQL database that we then used for the evaluation of the systems and for obtaining sense annotations through a user study that is described in the following section.

### 4.4.2. User Study

To gather the human input, we created an evaluation webpage, that the users could access any time during the data gathering period. The annotators were given randomly selected topics from the topics targeted for evaluation. The topics were presented as groups of words. We targeted for annotation the top seven most probable words of the topics. For each topic, the system would give the user one word at a time, with the definitions of the extracted possible senses. For each sense, they could assign a score on a scale from 5 to 1 where 5 represents “Perfect match”, 3 represents ”Acceptable”, and 1 represents “Not related at all”. They could also label a word as being too ambiguous in the given context, could label the whole topic as too ambiguous, or just skip the whole topic. Figure 4.6 shows a screenshot of the web user interface.

77 annotators participated in the experiment, most of them being PhD students and postdoctoral researchers. Each sense has been annotated by three different annotators. After removing topics and words marked as too ambiguous by at least one user, this resulted in 55 topics from the BAVE corpus, 28 from BBC, and 33 from the Semantic Web corpus. The final set contained a total of 116 topics, 500 unique words, 633 words occurrences to be disambiguated, and 2578 annotated senses.

### 4.4.3. Compared Methods

For understanding the independent effects of using the E-WSD for disambiguation, and of using the adaptive sense modelling scheme described in Section 4.3.2, we implemented three novel joint WSD methods:
**Basic E-WSD** In this method, given a feasible solution to score, the system generates the bipartite graph between the senses and their common gloss-words, weighting all edges to 1. Then it computes the eigenvalue of the matrix obtained by multiplying the adjacency matrix of the bipartite graph to its transpose. The eigenvalue of this matrix is then used to score the feasible solution. Figure 4.2 on page 64 illustrates exactly this computation.

**ASV+SSP** This method implements the other contribution of this chapter, the adaptive sense vectors (ASV), together with the heuristic presented in Section 4.3.2. Rather than using the eigenvalue as a measure of solution coherence, this method uses the traditionally used sum of scalar products (SSP). After sense modelling with ASV, a weighted bipartite graph is obtained, containing the senses and their common gloss-words. We name the weight matrix of the graph $W$. Then, the hubs matrix is computed as $WW^T$, whose cells contain the pairwise scalar products of the senses. The sum of scalar products is obtained by summing up the values in the upper triangle of the hubs matrix, excluding the diagonal. This value is then used to score a feasible solution.

**ASV+E-WSD** This method implements both contributions of this chapter, by computing the hubs matrix exactly like ASV+SSP, but scoring the feasible solution by the eigenvalue of the $WW^T$ matrix.

We compared our results with methods from related work:

**Cowie (1992)** The method of Cowie et al. [31], discussed in Section 3.1.2, scores a feasible solution by summing up the pairwise word-overlap of the senses. It creates the adjacency matrix similar to **Basic E-WSD**, but rather than computing the eigenvalue of the adjacency matrix, it computes the sum of scalar products by summing up the values in the upper triangle of the hubs matrix (excluding the diagonal values).

**Banerjee (2002)** In Section 3.1.2, we describe Banerjee and Pedersen’s WSD method [8]. In this evaluation, we use the scoring of the overlap between two senses, as in Formula (3.5) on page 44. We use this measure as an alternative pairwise sense relatedness measure. We use it for feasible solution coherence by summing up the pairwise sense relatedness.

**Sinha (2007)** The third related work we implemented is the work of Sinha and Mihalcea [123]. This is also the only WSD method we evaluate that does not use a global disambiguation approach, but extended-local optimisation. Sinha and Mihalcea [123] propose a WSD framework that we summarised in Section 3.1.2. We create a graph containing all senses of all target words, and connect the senses by edges weighted by the number of
overlapping words. Then, the senses are scored based on their weighted degree centrality in this graph.

**CosSim** Another approach we find relevant for comparison computes the pairwise sense relatedness with the cosine similarity of their bag of words representations. Cosine similarity has been widely used with Wikipedia, so we are interested to see if it can also perform well with descriptions of senses that are much shorter than the Wikipedia text of the corresponding article. We have also experimented with the cosine similarity of the ASV representation of senses (ASV+CosSim).

We use the three methods we introduce as well as Cowie (1992), Banerjee (2002), CosSim and ASV+CosSim in global WSD. Sinha (2007) is the only local-extended approach we evaluate. For global disambiguation, the identification of the optimal solution out of the set of feasible solutions is NP-hard. However, in these experiments, we did compute all feasible solutions, because we want to estimate an upper bound of the performance of the measures. Of course approximation techniques can be used, but for the purpose of a clean comparison we do not use approximation in these experiments. In Chapter 5, we will describe a frozen-past approximation technique [130] that we later use with E-WSD. We will also propose a branch-and-bound complete search strategy, that suits any joint WSD that uses the sum of pairwise relatedness for scoring the feasible solutions.

### 4.4.4. Tests and Performance Measures

**Tests** For accuracy assessment of the WSD algorithms, we collapsed the five sense annotation categories used in the user study into two categories, correct and wrong. We used two settings, shown in Table 4.5: one in which the “3 - Acceptable” annotation counts as correct, and one in which it counts as wrong. We name the first setting the **Relaxed test**, and the second setting the **Strict test**. Thus, in the Relaxed test, a returned sense is a hit if it obtained at least two human annotations of 3, 4, or 5. For the Strict test, we count a hit only if at least two out of the three human annotations are 4 or 5. We refer to a sense that has two or more annotations of 3, 4 or 5 as **CorrectRelaxed**, and to a sense that has two or more annotations of 4 and 5 as **CorrectStrict**.

**Computationally Ambiguous Words (CAWs)** Because we gathered senses from two sense inventories, there is a high chance that more senses of one word are correct by human’s
Table 4.5: Algorithm performance assessment scheme.

<table>
<thead>
<tr>
<th></th>
<th>Relaxed</th>
<th>Strict</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>3, 4, or 5</td>
<td>4 or 5</td>
</tr>
<tr>
<td>Wrong</td>
<td>1 or 2</td>
<td>1, 2, or 3</td>
</tr>
</tbody>
</table>

For example, the word apple contains the “fruit” sense in WordNet as well as in DBpedia. Thus, in our case, rather than distinguish between polysemous and monosemous words, it is of more importance to distinguish between (i) words that were annotated with both right and wrong senses, and (ii) words annotated only with right or only with wrong senses. Only words annotated with both wrong and right senses pose an actual disambiguation challenge to the WSD algorithms, so we simply call them computationally ambiguous (CAW). Words whose senses were all either annotated right or all wrong are consequently called computationally non-ambiguous.

**Fuzzy Filter** We define fuzzy senses as senses where human annotators disagreed (for example, a sense assessed by an annotator with 1—“No relation at all” and by the other 2 annotators with 4—“Good”). Because these senses introduce much inter-annotator disagreement, we devise a test where we remove these senses from the computation, in which case we say that the fuzzy filter is ON. As seen in Table 4.7, the inter-annotator agreement greatly improves when the fuzzy filter is ON and these senses are not considered. If a removed fuzzy sense is the sense an algorithm chose as correct for a word, we ignore that word in the evaluation of that algorithm. Also, a fuzzy sense removal might cause a word that was computationally ambiguous, to become computationally unambiguous. These two effects explain the lower, and approximate numbers in Table 4.7 for this setting.

**WSD Accuracy Measure** Typically, in WSD, if a system retrieves more than one sense for a word, it is considered a miss, because the system was not able to discriminate the correct sense from the wrong one. However, in our case, a word might have more than one correct sense, because we simultaneously use two sense inventories. As seen in Table 4.6, humans annotated on average more than 2 senses per word as being CorrectRelaxed, and 1.79 senses per word as being CorrectStrict.

Consequently, while the retrieval of two or more senses for a word shows that the system was not able to discriminate between the senses, the retrieved senses might actually be all
Table 4.6: Average number of correct senses per word as annotated by humans, and as retrieved by each method

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#senses per word</td>
<td>2.36</td>
<td>1.79</td>
<td>1.02</td>
<td>1.16</td>
<td>1.04</td>
<td>1.14</td>
<td>1.14</td>
<td>1.07</td>
<td>1.02</td>
<td>2.48</td>
</tr>
</tbody>
</table>

Under this assumption, we count a hit for the system for a particular word, only if all returned senses are correct. Therefore, we define accuracy as in the following formula:

\[
\text{Accuracy}^{<\text{Test}>} = \frac{\# \text{ of words with all returned senses being Correct}^{<\text{Test}>}}{\text{total number of CAWs with at least one sense that is Correct}^{<\text{Test}>}}
\]  

(4.15)

where \(<\text{Test}>\) is the place-holder for \textit{Relaxed} or \textit{Strict}. As seen in Table 4.6, \textbf{ASV+CosSim} has very poor discrimination power retrieving many more senses than human annotators considered correct. This is due to the small dimensionality of the adaptive sense vectors. This method achieves therefore very poor accuracy results, constantly lower than 30%. We conclude that the combination of ASV with cosine similarity is not suitable for disambiguation and in the following reports we ignore this setting.

**Context Size** As previously mentioned, the users annotated the words in the context of top-7 words from LDA topics. However, we assume that the assessment a user decided for in the context of 7 words, is the same assessment they would have decided if they had seen only the top-3, top-4 words of the topic and so on. Making this assumption, we can evaluate the performance of the word-sense disambiguation methods, with various context sizes. We therefore ran the disambiguation methods, for top-3 to top-10 words in the topic.

Table 4.7 shows the four tests we ran, and the number of total evaluated words, for each context size. We also report for the Fleiss Kappa value for inter-annotator agreement, for the 7 annotated words per topic.
Table 4.7: Number of words with ground-truth annotations for the various settings; When Fuzzy Filter in ON, the number of words are approximate (=) as the removal of fuzzy senses also causes removal of some words, depending on the algorithm. The reported values are the medians over the algorithms.

<table>
<thead>
<tr>
<th>Test</th>
<th>Only CAW</th>
<th>Fuzzy Filter</th>
<th>Fleiss Kappa</th>
<th>Total number of words</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Relaxed</td>
<td>YES</td>
<td>ON</td>
<td>0.83</td>
<td>≈131</td>
</tr>
<tr>
<td>Strict</td>
<td>YES</td>
<td>ON</td>
<td>0.73</td>
<td>≈151</td>
</tr>
<tr>
<td>Relaxed</td>
<td>YES</td>
<td>OFF</td>
<td>0.66</td>
<td>172</td>
</tr>
<tr>
<td>Strict</td>
<td>YES</td>
<td>OFF</td>
<td>0.63</td>
<td>192</td>
</tr>
</tbody>
</table>

4.4.5. Results and Discussion

We now report and interpret the results. Figure 4.7 shows the accuracies obtained on the computationally ambiguous words. As the highest inter-annotator agreement is obtained when the fuzzy filter is applied, Figures 4.7(a) and 4.7(b) are the most conclusive. All the illustrated results were obtained on computationally ambiguous words, for capturing the methods’ disambiguation capability. As most words that were non-computationally ambiguous have only correct senses, their addition leads to an approximately 5% more accuracy than the results obtained in the tests with the fuzzy filters set to OFF.

The results clearly show that ASV+E-WSD achieve the same performance over all test. This method is closely followed by ASV+SSP. When the fuzzy filter is ON, these two methods have an advantage of more than 0.2 accuracy over all other methods, especially in contexts bigger of 8 words for the Strict test, and 6 words for the Relaxed test. This shows that the adaptive sense vectors bring a great improvement over methods that do not adapt the sense models to the feasible solution. Also the better performance of ASV+E-WSD shows that the eigenvalue, by capturing the connectivity between senses and gloss-words, is superior to the traditional sum of scalar products.

We now interpret the effects of each novel aspect that our work brings, as well as other interesting factors.

The effect of eigenvalue is noticed when comparing the performance of ASV+E-WSD to that of ASV+SSP, and also the performance of Basic E-WSD to that of Cowie (1992). ASV+E-WSD constantly performs better than ASV+SSP with 2 to 5 percent. Still, Basic E-WSD only achieves statistically significant better results than Cowie (1992) in the Strict test, as measured with paired t-test at a significance level $\alpha = 0.05$. An important benefit of using
the eigenvalue of the hubs matrix for WSD is brought in small contexts. In the Strict test, Basic E-WSD has the best performance in the smallest evaluated context, of 3 words. This trend is also present in the Relaxed tests, where Basic E-WSD achieves in both settings an accuracy of more than 0.7 when 3 words are considered (Figure 4.7(b)). In the Relaxed tests, the other eigenvalue based method, ASV+E-WSD starts at 3 words with an accuracy of 0.76 when the fuzzy senses are removed, and 0.7 on all senses.

**The effect of the adaptive sense modelling (ASV)** can be observed by comparing ASV+SSP to Cowie (1992), and also by comparing ASV+E-WSD to Basic E-WSD, its unweighted analogue. ASV brings a strong improvement, as the two methods employing it delimit themselves from all the other methods, by up to 0.15 in the Relaxed test with applied fuzzy filter, as seen in Figure 4.7(b). The highest benefit of ASV is in contexts larger than 5. In the Strict tests, when 3 words are used as context, the ASV-based methods show no benefit
as compared to the others. But as more words are being considered, while the other methods either lose in performance or stay constant, the disambiguation accuracy of the ASV methods increases. This shows that the ASV methods are able to make use of the words in the context as cues for disambiguation, and are not affected by the noise caused by the wrong senses of the new words. This comes to reinforce the importance of capturing the relevance of gloss-words in the computation of solution coherence.

The effect of joint disambiguation is best observed when comparing Cowie(1992) to Sinha(2007), as these two measures use the same pairwise relatedness measure (the word overlap), and only differ in the disambiguation strategy: Cowie(1992) uses global disambiguation and Sinha(2007) uses it in local-extended disambiguation. Sinha(2007) performs slightly better, but not statistically significant. As a local-extended disambiguation approach, Sinha(2007) uses for the disambiguation of one word, all the senses of all other words simultaneously, therefore, the scores are “contaminated” by the wrong senses. While this criticism is well justified, it does not mean that global disambiguation is much more noise resilient. Actually, global disambiguation entails the risk of a feasible solution containing mostly alien senses to be retrieved. Therefore, used trivially, global disambiguation does not perform better than local-extended disambiguation, and also comes at the expense of a higher computational cost. But, the advantage of global disambiguation stands in its ability of isolating noise from signal, by treating the feasible solutions separately. As such, as shown by the high performance of ASV+SSP and ASV+E-WSD, by optimising the solution coherence, global disambiguation clearly outperforms state-of-the-art local-extended disambiguation.

The effect of the context size An important thing to notice is that the variation due to the size of context is quite small, with a standard deviation averaged over all methods of 0.021. All joint disambiguation methods except ASV+E-WSD and ASV+-SSP, achieve their best performances in contexts of only 4 words. This high performance with small contexts is very likely due to the use of topic models as contexts. This is a highly valuable insight because it appears reasonable to believe that an approximate search technique for global optimisation can split the context and solve multiple smaller subproblems, drastically reducing the search space without sacrificing disambiguation performance. For big contexts, this would work by jointly disambiguating for instance the top-4 words, then proceed to jointly disambiguating the following 4 words, by also adding to the sense combinations the already disambiguated senses, and so on. In Chapter 5 we exploit exactly this potential and propose a new approximation technique that overcomes the exponential complexity of global optimisation.
Another noticeable point regarding context size, is that *Sinha (2007)* is the most stable of all evaluated methods, with respect to the number of words in the context. Its stability, and also its relatively good performance, can be explained by the fact that the addition of a new word in the computation, adds its senses to the graph that already contains all possible senses of all other words. Thus, the impact of only one word over the centralities in the sense graph is quite limited, having a smoothing effect on the performance curve.

**The effect of gloss length** Using gloss-overlap based methods for word-sense disambiguation, entails the risk of a bias towards senses with long glosses. By weighting the overlapping gloss-words based on their relevance through ASV, we aim to reduce this bias.

![Box-plot representation of the distributions of senses over gloss lengths](image)

**Figure 4.8:** Box-plot representation of the distributions of senses over gloss lengths

Therefore, an important angle from which to analyse the performance of the evaluated methods, is the lengths of the glosses of the senses they retrieve. The box-plot in Figure 4.8 shows the distributions of the number of gloss-words of the senses retrieved as correct by the evaluated methods, the length of the senses assessed by human annotators as *Correct*\(^{Strict}\) and *Correct*\(^{Relaxed}\), the lengths of all annotated senses, and the lengths of the WordNet and DBpedia senses. This figure reinforces that the DBpedia glosses are much longer than the WordNet glosses. Also, the human annotators assessed as “Perfect” and “Good” slightly
longer glosses than the “Acceptable” senses. As for the methods, this figure explains why \textbf{CosSim} performs much worse than all other methods: the normalisation by length of gloss that is performed by the computation of cosine similarity, gives high weight only to solutions containing very short sense glosses. We also notice that the ASV based methods are least biased to the length of senses. It is also noticeable that \textbf{Sinha(2007)} gives highest preference to long glosses, since it measures the overlap between all senses of all words. \textbf{Basic E-WSD} is the method with second longest glosses, and this is because, as opposed to the other joint-based disambiguation methods, it takes into consideration the values on the diagonal of the hubs matrix - the magnitudes of the senses. \textbf{Banerjee(2002)} retrieves senses slightly shorter than \textbf{Cowie(1992)}, as the only difference between these two methods is that the former uses a gloss-word weighting scheme.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4-9}
\caption{The proportion of WordNet senses out of all retrieved senses}
\end{figure}

\textbf{The effect of using multiple knowledge bases} An analysis related to that of gloss length, is the analysis of the distribution of senses over the two used knowledge bases, WordNet and DBpedia. This is shown in Figure 4.9. Approximately 70\% of the senses annotated as Correct by humans are WordNet senses. As expected from the previous results, \textbf{CosSim} retrieves almost only WordNet senses, in a percentage of more than 90\%. The method closest to human’s assessment is \textbf{ASV+SSP}, followed by \textbf{ASV+E-WSD}, therefore, ASV fulfils with success its purpose of strongly reducing the bias in the selection of the disambiguation senses. \textbf{Cowie(1992)} and \textbf{Banerjee(2002)} are highly correlated, with the latter choosing slightly more WordNet senses. \textbf{Sinha(2007)} selects the smallest percentage of WordNet senses, and \textbf{Basic}}
E-WSD retrieves slightly more, but still less than all the other joint measures. It is important to notice that except for CosSim, the proportion of WordNet senses decreases as the number of words in context increases.

Therefore, the traditional as well as state of the art methods for gloss-based WSD present a high bias either towards very short glosses, like CosSim, or towards long glosses. This becomes a problem when multiple knowledge bases, with different gloss length distributions need to be simultaneously used. The proposed sense weighting method, ASV, reduces the sensitivity to gloss-length of disambiguation methods that use it.

To sum up, the proposed sense modelling and the eigenvalue-based measure for WSD, achieve very good results, while reducing the gloss-length bias of the other methods. ASV fulfils the role of weighting the overlapping gloss-words, while E-WSD supports a more discriminative selection of disambiguation senses. Used together, they achieve the best results, almost reaching 80% accuracy in both Relaxed and Strict tests on ambiguous words, on ground truth data that exhibits 0.83 and 0.73 inter-annotator agreement. ASV, when used with sum of scalar products achieves the second best performance. When E-WSD is used without ASV, it manages to perform third, but its score improvement is not very high, ranging from 1 to 3 percent.

4.5. Conclusions

Currently, the unsupervised WSD scene contains mostly complex knowledge-based approaches. While they can achieve very good performances, they suffer from lack of portability. Gloss-based approaches only require a bag of words representation of senses that can be obtained from any typical sense inventory. In the context of Linked Data, gloss-based disambiguation can provide the means for simultaneous linking to multiple knowledge bases, without the requirement that they are interconnected. Establishing relatedness of words, concepts and topics based on limited textual representations can assist the process of interlinking heterogeneous datasets.

In this chapter, we introduced a novel gloss-based disambiguation method, based on the intuition that the gloss-words differ in terms of relevance to the disambiguation solution. We approach this idea in a two-step manner. First, gloss-words are weighted based on their frequency in the joint disambiguation solution. Second, the solution’s coherence is computed with the principal eigenvalue of the hubs matrix of the solution’s graph representation. The
eigenvalue computation implicitly emphasizes the importance of frequent gloss-words. We also propose a very basic heuristic considers overlapping named entities as a strong evidence for sense relatedness or lack of it. Putting these three ingredients together, we obtain a WSD approach that clearly improves state-of-the-art gloss-based disambiguation approaches.

In the following chapter, we propose another joint WSD approach, Sen-Dis, that only relies on the underlying graph-structure of the knowledge base. For comparison, we also revisit ASV+E-WSD, and use it for disambiguating words from text documents, rather than isolated topics. Furthermore, in the following chapter we address the computational complexity limitation of global disambiguation. ASV+E-WSD also plays an important part in Chapter 6, as we use its disambiguation results in the topic labelling experiments.
Chapter 5.

Proximity Measures on the DBpedia Graph for WSD

This chapter proposes a novel approach to word sense disambiguation, that uses the DBpedia underlying graph structure. As discussed in Section 3.1.4, the current approaches that use DBpedia as a sense inventory for word-sense disambiguation, rely on Wikipedia text and hyperlink structure. In the work presented here, we research to what extend DBpedia can be used as a stand-alone knowledge base. To this end, we view DBpedia as a semantic network, and propose existing as well as novel graph-based proximity measures to capture semantic relatedness based on this network. After we show that these measures correlate with human assessment of semantic similarity, we integrate them into a global WSD system, called Sen-Dis. Global disambiguation based on the structure of the semantic network has only been previously attempted in 1993 [130], with WordNet, using relatedness measures that cannot be directly applied on DBpedia. In this chapter, we show that this vastly ignored direction is very effective on the DBpedia semantic network, which is much bigger and whose structure is much less regular than that of WordNet. Our experiments show that our approach achieves better results than much more complex approaches that require the preprocessing of whole Wikipedia.

5.1. Introduction

The development of Linked Data in the last decade, has resulted in a multitude of structured knowledge bases becoming available, and many applications for text enrichment, text summarisation and indexing can avail of this knowledge. In most cases, automatic word-sense disambiguation is required to solve language ambiguity. The structure of the targeted knowl-
edge bases can be used as an important source of disambiguation cues. In this chapter, we focus on graph structured knowledge bases, and since we analyse them from the perspective of the semantics of their nodes and edges, we call them semantic networks. From the word-sense disambiguation point of view, semantic networks, together with dictionaries, glossaries, thesauri and encyclopedias are types of sense inventories.

In the previous chapter, we presented a novel approach to gloss-based word sense disambiguation. Gloss-based word sense disambiguation approaches are very versatile because their dependency on the sense inventory is reduced compared to other types of approaches. They can be used on any sense inventory that provides a definition, or the possibility of bag-of-words representation of senses. Gloss-based approaches also permit disambiguation using multiple sense inventories simultaneously, as we showed in the previous chapter. Nevertheless, they depend on the wording of the glosses, and therefore performance is affected by the use of different words to convey similar meanings, and by the use of ambiguous words. WSD approaches that use semantic networks do not suffer from this drawback, as in semantic networks, sense representation does not require its textual gloss.

In semantic networks, senses are represented by nodes and their properties and relations to other senses are explicitly stated in a structured, unambiguous way. The semantic relations that are represented in semantic networks are often part of domain knowledge that would require considerable effort to deduce out of the text of the glosses. These explicit relations between senses provided by semantic networks have been widely researched for WSD as shown in Section 3.1.3 of the Chapter 3. The intuition is that words that co-occur in language have senses that exhibit relations that are either directly stated in the semantic network, or can be inferred from its structure. The existence or non-existence of such relations provide evidence for word-sense disambiguation, shifting the balance towards senses that are most related to the other senses in the context.

In this chapter, we research how the background semantic network can be used from a graph-analysis perspective. As such, apart from controlling what types of semantic properties are considered, we do not differentiate between these properties. Once the considered properties are selected, we only analyse the structure of the obtained underlying graph. The key idea is to extract, from the semantic network, those senses of the target words, that are most closely connected. However, this is a challenge itself as it requires the definition of a network proximity measure that would capture semantic relatedness outside the scope of a predefined, targeted domain. Nevertheless, as we showed in Chapter 3, most previous semantic relatedness measures used for WSD for the general domain, assume hierarchical sense inventories like WordNet. On the other side, measures that use generic graph structures rely on hyperlinks extracted from
encyclopedic web content like Wikipedia. Arguably, the network produced by the hyperlinks between Wikipedia articles is not a semantic network, as the connections between articles have no knowledge-based meaning. Therefore, there is a gap in the analysis, understanding and evaluation of how word sense disambiguation can be performed on non-hierarchical semantic networks that are not backed up by a massive encyclopedia like Wikipedia. Another challenge is that, once such a relatedness measure is defined for pairs of senses, the problem of optimising the relatedness between multiple senses simultaneously is NP-hard, as it reduces to the maximum edge weighted clique problem [110].

In this chapter, we address both challenges. To this end, we relax the requirement of having senses organised in a IS-A hierarchy, and at the same time we do not exploit Wikipedia as an encyclopedia, or its hyperlinks. As such, our proposed WSD method, called Sen-Dis (Semantic Network Word-Sense Disambiguation), only makes use of the DBpedia structured knowledge. Regarding the NP-hard optimisation problem, we propose a branch-and-bound (B&B) methodology that returns the optimum solution without explicitly evaluating all feasible solutions. In addition, for the disambiguation problems whose search space overpasses a set threshold, we wrap our B&B algorithm in an approximate search routine. This routine uses the insights from the previous chapter, that the number of words in context has limited influence on the performance of disambiguation. We therefore reduce the complexity of the problem by splitting the context, so that we can run complete optimisation on multiple smaller problems.

As discussed in Chapter 3, DBpedia has been used before as a sense inventory for WSD, but the actual disambiguation algorithms model the senses and achieve disambiguation with Wikipedia. DBpedia as a semantic network presents several characteristics that make it attractive for word-sense disambiguation research. First, it is very broad, containing approximately 4 million nodes with a fair representation of most domains: scientific, entertainment, news, art, and includes a very wide range of named entities like persons, organisations, places, events, artwork and so on. Second, it is the result of collaborative effort of the Wikipedia editors community, therefore the facts it contains have reached a certain degree of agreement within the community. Third, it is also the result of automatic extraction of facts from Wikipedia pages. Being the result of an open community effort and also of an automatic process, DBpedia is also noisy, unlike the semantic networks used in previous approaches. As most of the public structured knowledge bases are the result of either community effort or automatic processes, WSD methods must gracefully deal with fair amounts of noise in order to be useful. Fourth, disambiguation results can be easily compared to systems that use Wikipedia due to the mapping between DBpedia concepts and Wikipedia articles.

This being the motivation of this work, we now present the main contributions.
5.1.1. Contributions

This chapter brings several contributions:

- a graph-based approach to WSD with the DBpedia semantic network. To the best of our knowledge, this is the first approach that uses only the DBpedia underlying graph to achieve disambiguation. It achieves better results than previous approaches like DBpedia Spotlight, that make use of the entire Wikipedia knowledge;

- a comparative evaluation of established as well as novel graph-based semantic relatedness measures, on DBpedia;

- a branch-and-bound methodology for joint WSD that improves joint WSD efficiency without sacrificing optimality of the result. We also propose an approximate search method that we use to back up B&B if search space size overpasses a certain threshold.

- we show that traditional word clustering algorithms can greatly improve joint WSD performance by creating coherent disambiguation contexts.

- we introduce the notion of *stop-URIs* to denote narrow-scoped concepts that have no semantics outside the scope of the dataset they are part of. Our experiments show that these stop-URIs have a major impact on the performance of WSD, and their removal is crucial for the success of WSD with DBpedia as a semantic network.

We expect several benefits from attempting word-sense disambiguation with DBpedia network of concepts while breaking their ties to the Wikipedia articles. An important benefit is that the implementation and reproduction of graph-based methods is more accessible and requires less resources than that of methods requiring entire Wikipedia preprocessing. The amount and variety of noise that the disambiguation methods need to deal with is also reduced. In addition, the proposed graph-based word-sense disambiguation methodology has a great potential of being ported to other semantic networks.

5.2. Approach Overview

The intuition behind this work is similar to the one behind many previous approaches. It is the idea that the words that co-occur in language refer to concepts that lie in close proximity in a predefined semantic network. Given a particular topic and a semantic network that covers that
topic, one can spot the senses to which the words of the topic refer, because they exhibit higher interconnectivity than the wrong senses of the same words.

In the previous chapter, in the absence of a background semantic network, we were building an ad-hoc network as a bipartite graph connecting targeted senses to their gloss-words. Connections between senses could only be established through the common gloss-words. Natural language characteristics like synonymy and polysemy, as well as the particular choice of words in glosses, negatively affect the reliability of such ad-hoc networks.

In the scenario we consider in this chapter, a background semantic network is provided (i.e., DBpedia), that unambiguously, connects related senses, nonetheless with no claim to completeness. As already stated, the disambiguation consists of finding in the semantic network, the senses of the targeted ambiguous words that exhibit highest interconnectivity. We illustrate this intuition with the following toy example.

Another distinction between the previous chapter and this one is that here, we only analyse noun-phrases. In the previous chapter, our approach attempted to disambiguate all top words of topic models, irrespective of their part of speech. However, in this chapter we extract and analyse only the noun-phrases from the input text. Ultimately, our goal is to link and disambiguate all noun-phrases that have a corresponding DBpedia page.

### 5.2.1. Toy Example

Let us consider we need to simultaneously disambiguate words $A$, $B$ and $C$, which are known to be related. Senses $A_1$ and $A_2$ are possible disambiguation senses of word $A$, that $B_1$ and $B_2$ are possible senses of word $B$ and that $C_1$ and $C_2$ are the candidate senses of word $C$. Let us also consider that the background semantic network containing these target senses is as shown on the left side of Figure 5.1. All the other nodes in the graph are other concepts that belong to the semantic network. The edges are the relations defined by the semantic network. We consider all relations undirected, and ignore their actual semantics.

Measures for semantic relatedness between senses usually consider the network paths between them. The more and shorter the paths between nodes, the more interconnected they are and the more related the corresponding senses. We can intuitively see semantic relatedness measures as “summaries” of the network paths between senses. This summarisation translates into weighted edges between pairs of senses, resulting in a “condensed” form of the original graph as illustrated on the right side of Figure 5.1. The thickness of the edges is proportional...
Following the intuition that related senses of words lie close in the semantic network, the solution that seems most likely to be correct is \{A_2, B_2, C_2\}. In the toy example presented in this figure, the feasible solutions are all represented by triangles in the “condensed” graph. Generalising to any number of target words, each feasible solution corresponds to a maximal clique. The challenge for joint disambiguation is to extract from the “condensed” graph, the clique that has the maximum sum of edge weights. This problem is NP-hard, and in Section 5.6, we suggest a solution for finding the optimal solution.

This being the main idea behind the WSD approach we present in this chapter, we now describe how we apply this intuition in a system that takes documents as input and uses this method to disambiguate the document’s phrases with DBpedia.

### 5.2.2. The Blueprint of Sen-Dis

Using the properties of unsupervised WSD systems introduced in Chapter 3, we summarise the proposed WSD system, Sen-Dis, as follows:
Proximity Measures on the DBpedia Graph for WSD

**Sense model** Senses are nodes in the semantic network used as sense inventory. As we are using DBpedia as a semantic network, and it contains more than 4 mil. nodes, we limit the scope of a sense to a neighbourhood sub-network. Thus, each sense has an ego-network associated to it, which is extracted from the semantic network. In Section 5.3, we describe the extraction of these networks. The nodes in the DBpedia semantic network are called in related literature DBpedia concepts [7]. In our work, words have senses, and senses in this particular case are DBpedia concepts. Therefore, throughout this chapter, we will refer to word senses as senses, DBpedia concepts or simply concepts.

**Context model** In this chapter, we research disambiguation contexts extracted from the targeted document. The main idea is to cluster words from the document based on their co-occurrence in an external corpus. Thus, we aim at obtaining groups of related words. The idea is reminiscent of lexical chains [11] and topic models [126], and we detail it in Section 5.4. We also experiment with the context being formed by all words from the text that have one or more possible senses in DBpedia, without clustering. We show in Section 5.8 that clustering improves disambiguation performance.

**Semantic Relatedness** In this chapter, we aim at quantifying the semantic relatedness between the senses in the extracted contexts, based on their proximity in the semantic network. We look into two types of graph-based proximity measures: path-based measures that consider length and number of paths between nodes to assess proximity, and neighbourhood-based measures that consider the similarity between neighbours of nodes. For the first class of measures, we use the shortest path, and also tailor two measures that account for longer paths. As for the second class of measures, we use a semantic relatedness measure proposed by Agirre et al. [4]. It weights the neighbours of a node by their Personalised PageRank score, and assesses similarity between nodes by cosine similarity between the vectors of neighbours. We describe the methods in more details in Section 5.5. As previously mentioned, the scope of a sense is limited to its ego-network, so we compute these measures on the sub-network resulted from merging the ego-nets of the targeted concepts. If the ego-nets do not connect, concepts are considered unrelated. The process of ego-net extraction is further explained in Section 5.3.

**Disambiguation** After computing the pairwise semantic relatedness between senses, we use joint WSD for identifying the optimal feasible solution. In the previous chapter, we report results obtained through a complete search over the feasible solution search space. In this chapter, in Section 5.6 we present a branch and bound methodology that also achieves optimal solution finding, but without explicitly analysing all feasible solutions. We also propose an approximate search method based on the frozen-past technique, that we use as
a wrapper for the branch and bound algorithm when the search space is too big for timely optimal solution finding.

5.2.3. Sen-Dis Process Overview

Given a document, the disambiguation of its noun phrases to DBpedia with Sen-Dis requires several stages of processing:

1. Extract noun phrases from document;

2a. For the resulting noun-phrases, extract candidate senses in DBpedia and their ego-networks;

2b. Create disambiguation contexts for the noun-phrases by clustering them based on co-occurrence in an external corpus;

3. Compute pairwise relatedness between sense candidates of different words using their ego-networks;

4. Use the pairwise relatedness scores to select the most inter-related feasible solution;

For the first step, we extract the noun-phrases from the input document with Stanford CoreNLP toolkit. In the remainder of this chapter, we describe our proposed solutions to stages 2-4 of the process. Section 5.3 explains the notion of ego-networks and how we extract them from DBpedia. The clustering algorithms we use for grouping words are presented in Section 5.4. Afterwards, Section 5.5 presents our proposed relatedness measures, and provides a thorough evaluation in which we assess the measures’ ability to capture semantic similarity/relatedness, compared to humans. Section 5.6 presents the joint disambiguation methodology, and its subsequent section puts all these pieces together in the Sen-Dis system architecture. Section 5.8 presents the evaluation of the whole disambiguation system, and then in the final section we draw the main conclusions of this work.

5.3. Sense Ego-Networks

As mentioned above, senses are nodes in the semantic network of DBpedia. DBpedia contains more than 4 mil. concepts, therefore working with the entire graph in order to establish relations

\[\text{http://nlp.stanford.edu/software/corenlp.shtml}\]
between senses would be very inefficient. We restrict the space around the concepts of interest to a 2-step neighbourhood. We use the concept of ego-networks [41] from social network analysis. The idea is to extract for a targeted actor, called “ego”, all the neighbours and also the relations between the neighbours. The ways in which the nodes in the ego-network interact sheds light on the properties of the “ego”. In our semantic network scenario, we use concepts’ ego networks in order to establish relations between them.

**Definition 4.** Given a set of DBpedia relation types \( \mathbf{R} \) we denote by \( \text{DBpedia}^{\mathbf{R}} \) the subgraph of the DBpedia semantic network formed only by the relations of type included in \( \mathbf{R} \). Then we define the sense ego-network or ego-graph of \( k^{th} \) degree of a sense \( s_i \), as an undirected graph \( G^k_i = (V^k_i \cup \{s_i\}, E_i, s_i, \mathbf{R}) \), where \( V^k_i \) is the set of DBpedia concepts lying at most \( k \) steps away from \( s_i \) in \( \text{DBpedia}^{\mathbf{R}} \) and \( E_i \) is the set of undirected edges between pairs of concepts in \( V^k_i \cup \{s_i\} \), that are connected by an edge in \( \text{DBpedia}^{\mathbf{R}} \). \( s_i \) is called the seed concept or seed node of graph \( G^k_i \).

The edges in \( E_i \) correspond to the semantic relations between the concepts in DBpedia. As we consider that semantic relations have reciprocals (broader-narrower, worksAt-employs, etc), and we do not differentiate among types of relations, all edges in \( E_i \) are undirected. For sake of simplicity and generality, regardless of how many relations exist between two nodes, we only connect them by one edge.

Now that we have defined sense ego-networks, we show how we use them to represent feasible solutions. Given a feasible solution, we create its solution network by merging the ego-networks of its senses. In Figure 5.2, we use the example semantic network and senses from Figure 5.1. The three graphs in Figure 5.2(a) are the ego-networks of 2nd degree of senses \( A_1, B_1 \) and \( C_1 \). The graph in Figure 5.2(b) represents the solution network of the feasible solution \( \{A_1, B_1, C_1\} \), created by merging the three sense ego-graphs, and adding any additional edges that exist between nodes, for example edge \( H \rightarrow L \) that does not belong to any of the senses ego-graphs. In the context of a feasible solution, the pairwise relatedness between its senses is computed on the solution’s network.

Regarding the degree of the ego-networks, we have experimented with 2-step and 3-step neighbourhoods. The size of the ego networks in terms of number of nodes grows exponentially with the average node degree. In our experience, the potential gain from the analysis of the bigger network does not justify the cost of the computational difference between 2-step neighbourhoods and 3-step neighbourhoods. We therefore use, throughout this chapter, 2nd degree ego networks.
5.3.1. Preliminary Check on Sense Ego-Network Overlap

The core assumption we make in our work is that the sense ego-graphs of related words are more likely to become connected than random concepts in DBpedia. In order to validate this, we ran some preliminary experiments on the ground truth data we collected in the user study presented in Chapter 4. This data consists of 116 topics that had all the top-7 words manually linked and disambiguated by human users against DBpedia concepts or WordNet synsets. Out of the 116 disambiguated topic, 111 contained at least two DBpedia senses annotated as correct. We used only the DBpedia concepts and computed for each topic a measure of pairwise concept
seed connectivity after two hops.

\[
PairConnectivity(topic) = \frac{\text{\# sense pairs whose egonets have at least one common node}}{\text{total number of sense pairs}}
\] (5.1)

We compared the obtained average pair connectivity over all 111 topics, to the same measure applied to identical 111 groups of DBpedia concepts formed randomly from the set of all DBpedia senses in the dataset. For the case of correct concepts, we obtained a mean connectivity of 0.46 with standard deviation 0.09, while the random groups had the mean of pair connectivity of 0.07 and standard deviation 0.02. These values indicate that the connectivity of seed concepts obtained from correctly linking and disambiguating related words to DBpedia is not accidental.

5.4. Word Clusters as Disambiguation Context

Previous work on disambiguation with lexical chains [11, 57] shows that proximity of words in sentences does not necessarily imply their senses’ relatedness, but that words that refer to the same topic occur scattered in text. We drew a similar conclusion in the previous chapter that shows that words can be disambiguated with success in the context of LDA topic models, and LDA assumes bag-of-word representation of documents [15], ignoring word ordering. Topic models group words based on their co-occurrence in multiple documents. In this chapter, we follow a similar route but tailored to documents. We group the words in the target document based on their co-occurrence in an external corpus. These groups then serve as context for disambiguation.

The main phrases and words from a document can be grouped using clustering algorithms or and topic inference approaches. Clustering firstly involves the definition of a relatedness or a similarity measure on the phrases; then the phrases are grouped into clusters that maximise intra-cluster relatedness and minimise inter-cluster relatedness. This measure of relatedness between phrases usually considers their co-occurrence in language, estimated by counting occurrences over big text corpora. On the other side, topic inference involves the preliminary training of a topic model, for instance LDA, on an external corpus. Then, once the model is learnt, more specifically the distribution of words over topics and the distribution of topics over the seen documents, inference is used in order to estimate the topic distribution of the new
document. This includes the assignment of the words in the new document to the predefined topics.

Clustering is therefore a lazy-learning technique while topic inference is eager-learning. As such, clustering has the advantage of providing more flexibility for experimentation than topic inference. Therefore, for sake of simplicity and flexibility, at this stage we use clustering algorithms.

5.4.1. Employed Clustering Methods

In order to cluster the text words, we use a reference corpus, and compute co-occurrence statistics of words over it. We have experimented with the Wikipedia full-text corpus because it covers a broad range of domains. As for co-occurrence statistics, we use the positive point-wise mutual information, a very commonly used measure for word co-occurrence. If \( p(w_1, w_2) \) denotes the proportion of documents in the corpus in which both words \( w_1 \) and \( w_2 \) occur and \( p(w_i) \) represents the proportion of documents in which word \( w_i \) occurs, then point-wise mutual information (\( \text{pmi} \)) \([25]\) and positive point-wise mutual information (\( \text{ppmi} \)) are defined as:

\[
\text{pmi}(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}
\]

\[
\text{ppmi}(w_1, w_2) = \begin{cases} 
\text{pmi}(w_1, w_2) & \text{if } \text{pmi}(w_1, w_2) \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]

In this work, we have experimented with two types of clustering algorithms: agglomerative hierarchical clustering, and modularity optimisation. Agglomerative hierarchical clustering produces a hierarchical structure, called dendrogram, by progressively merging clusters until all words belong to the same cluster. Various strategies have been used in literature for choosing the next pair of clusters to merge. We have preliminarily experimented with multiple strategies, and we only report experiments with the complete linkage strategy \([35]\). In order to obtain a flat clustering, the dendrogram is usually “cut” by either fixing the number of clusters or the minimum allowed similarity within the clusters. Therefore, in order to obtain a flat clustering with agglomerative hierarchical clustering, a parameter needs to be set. We tested various threshold settings with the complete linkage strategy and in the reported experiments we set...
it to 0.8. The intuition of this method is that the words are clustered in such a way, that no group contains any pair of words whose ppmi similarity score is lower than 0.8. Our preliminary experiments indicated that this value is low enough to lead to clusters even when few noun-phrases are extracted from the documents, and also high enough to ensure the grouped noun-phrases are related. However, it should not be regarded as a ground truth value, and this threshold should be set in accordance with the combination between the background corpus and length of target documents.

On the other side, modularity optimisation is non-parametric, and mostly used in social network analysis for community finding in graphs. Given a partitioned network, the modularity is a measure that captures the density of links inside partitions as compared to the density of links between partitions. We use the Louvain [19] clustering method that works by optimising modularity. The clustering intuition is to obtain groups of words that are highly similar to each-other within the groups, while they are dissimilar to the words in the other groups.

5.5. Relatedness measures

In this chapter, we introduce the graph-based relatedness measures, that we use for WSD. The need of establishing semantic relatedness between concepts that are part of a graph-based knowledge base is not specific to WSD. Many researched semantic relatedness measures are domain dependent [78, 79, 97, 129], and adaptation to new domains requires manual configuration. With WSD the system does not know in advance what domain each target word is part of, and it must compute the relatedness over all the domains of the senses. For instance, the approach presented by Leal et al. [78] requires specified types of nodes and properties for the targeted domain. The authors exemplify with the music domain, the node types being musical artist, genre, instrument, and the properties being has genre, plays instrument. Different properties have predefined weights that are used to assess the relatedness between entities in the music domain. This type of relatedness measure would require a great effort of manual configurations to cover all domains and all pairwise combinations of domains.

Other related methods emerged for extracting relevant relation paths between pairs of known entities [55, 115]. These measures provide explanations as to why and how the entities are related, and they do not give an overall score of the pair’s relatedness. With respect to disambiguation, all these methods are more suitable for use after the disambiguation, to extend the knowledge about the relations between the disambiguated senses.
On the other side, as shown in Chapter 3, most methods that have been created with word sense disambiguation in mind, assume a hierarchical knowledge base where the depth of the senses in question, as well as properties of the least common subsumers play an important role in assessing sense relatedness. However, although hierarchical structures have been mostly researched, there is no evidence in previous literature that only hierarchical structures are helpful for disambiguation. Moreover, research that compared the use of IS-A hierarchical structures to the use of other types of links, found that best results are achieved when mixed types of relations are used [130]. Concepts in DBpedia are linked based on various semantic relations, and even the categorisation subset of this semantic network is not a hierarchy [46]. Therefore, our need is to research how general graph methods can capture semantic relatedness when used on DBpedia. In the following, we define and formalise the proposed measures.

5.5.1. Inverse Shortest Path [ISP]

The Inverse Shortest Path (ISP) relatedness measure is very basic and its main intuition was introduced in 1989 by Rada et al. [114]. The idea is that the relatedness between concepts in a semantic network is inverse proportional with the length of the shortest path between them, as in Formula (5.4). Rada et al. [114] defines the concept of semantic distance rather than relatedness, and it is computed as the shortest path length between two concepts in a semantic network.

\[ ISP(c_1, c_2) = \frac{1}{1 + dist_{12}} \]  (5.4)

where \( dist_{12} \) denotes the length of the shortest path between concepts \( c_1 \) and \( c_2 \). There is a 1 added to the numerator so that the relatedness between a concept and itself, in which case the shortest path is zero, is equal to 1.

5.5.2. Multiple paths based measures

While the shortest path between two nodes is a common way of measuring relatedness between nodes in a graph, other measures also take into consideration longer paths. The use of longer paths in conjunction with the shortest path is usually more informative about the association between two nodes. Here, we adapt two such measures, one inspired from Stephenson and Zelen [125]’s centrality measure, and one from Katz [68]’s centrality measure. The common ground of the two measures is that they both consider all the paths between the two nodes, but
they give higher weight to shorter paths. The weight of a path is a function of its length. The difference between the two measures lies in the definition of this path weighting function.

Because of the size of DBpedia, the computation of all the paths between two nodes would be very inefficient. Also, longer paths between concepts usually stand as weaker evidence towards their relatedness, while their computational cost is higher. Therefore, we limit the number of paths we consider to a parameter $k$, that we vary throughout the experiments, by only accounting for the top-$k$ shortest paths.

### Path Information Based Relatedness [PathInfo]

Stephenson and Zelen [125] suggest a node centrality graph measure, where a node is central if its paths to the other nodes are short. The authors consider all paths between two nodes, and the weight of a path is the inverse of its length. They define the strength of relation between two nodes as the sum of all the weighted paths between them. Translated to semantic networks, the formalisation of this definition is as in Formula (5.5).

$$
rel_{info}(c_1, c_2) = \sum_{p \in P_{12}} \frac{1}{\text{length}(p)}
$$

(5.5)

where $P_{12}$ is the set of all paths between concepts $c_1$ and $c_2$. The weight of each individual path falls within the $[0, 1]$ range, being 1 for the shortest path between two adjacent nodes. We leave this function undefined for the case of computing relatedness between a node and itself.

In our work, we limit the number of considered paths by counting only the top-$k$ shortest paths with $k > 0$. We also normalise the score by dividing the accumulated weight to $k$. Therefore the formula we propose is:

$$
rel_{info}^{(k)}(c_1, c_2) = \frac{\sum_{p \in SP_{12}^{(k)}} \frac{1}{\text{length}(p)}}{k}
$$

(5.6)

where $SP_{12}^{(k)}$ is the set of the top-$k$ shortest paths between concepts $c_1$ and $c_2$.

### Katz Relatedness [Katz]

The other measure we will tailor for computing relatedness between concepts in a semantic network is inspired from Katz [68]’s centrality measure. As in Stephenson and Zelen [125]’s
work, the centrality of a node in a graph is computed as the sum of its relation strength to all the other nodes in the graph. This pairwise relation strength though is computed differently. Here, the idea is that the effectiveness of a link between two actors in a social network is governed by a known, constant probability, $\alpha$. In case of a path made up of $k$ nodes, the probability of the path is $\alpha^k$.

$$rel_{Katz}(c_1, c_2) = \sum_{p \in P_{12}} \alpha^{length(p)}$$  \hspace{1cm} (5.7)

where $P_{12}$ is the set of all paths between concepts $c_1$ and $c_2$.

We also adapt this measure to fit our setting, by limiting the number of considered paths by counting only the top-$k$ shortest paths, $k > 0$, and then normalising, so that the formula becomes

$$rel_{Katz}^{(k)}(c_1, c_2) = \frac{\sum_{p \in SP_{12}^{(k)}} \alpha^{length(p) - 1}}{k}$$ \hspace{1cm} (5.8)

where $SP_{12}^{(k)}$ is the set of the top-$k$ shortest paths between concepts $c_1$ and $c_2$. We also change the exponent to $length(p) - 1$ so that the weight of the shortest path between two adjacent nodes is 1 and the function is undefined for the case of relatedness between a concept and itself. Since $rel_{Katz}^{(k)}$ and $rel_{info}^{(k)}$ take more paths into consideration, they are both more discriminative than ISP.

The Katz centrality has been adapted to measure the semantic relatedness between concepts previously by Nunes et al. [109], but they limit the number of considered paths by limiting the length of the considered paths. Therefore, the computation adds the weights of all paths shorter than a preset length. In our preliminary experiments though, we observed that with this strategy, the score of a pair is highly affected by nodes with many common neighbours. Consequently, pairs of nodes with high degree will have disproportionate numbers of paths between them.

Therefore, as presented we adapt the multiple based measures $rel_{Katz}^{(k)}$ and $rel_{info}^{(k)}$ so that they limit the number of paths considered.
5.5.3. Personalised PageRank Cosine Similarity [PPR]

We also experiment with the measure used by Agirre et al. [4]. This method combines graph metrics - Personalised PageRank [53] - with traditional information retrieval metrics - cosine similarity. Therefore it is the only graph proximity measure we evaluate that is not path-based.

Personalised PageRank (PPR) is a variation of the traditional PageRank algorithm, that biases the selection of the nodes visited at random jumps, to some nodes of interest. It can be used to compute a vector space representation of one node, by setting the algorithm so that every time a “random” jump is taken, it lands on the target node with probability 1. As in traditional PageRank, the probability of not taking the “random” jump is called the damping factor. Used for semantic relatedness, the idea is to compute, for each sense in the pair, the PPR scores for all other nodes in the semantic network (in our case, the ego-network of the pair). This way, each sense of the pair is assigned a vector of scored neighbouring nodes. The semantic relatedness between the nodes is then computed as the cosine similarity between their corresponding PPR vectors. Agirre et al. [4] used this measure for semantic relatedness with WordNet as a knowledge base. However, it has not been previously used for WSD.

These being the proposed measures, we validate their ability to capture semantic relatedness, before attempting WSD.

5.5.4. Evaluation of the Relatedness Measures

In this section, we evaluate the performance of the aforementioned semantic relatedness measures for capturing relatedness and/or similarity between DBpedia concepts. We perform this evaluation in order to validate the suitability of these measures for assessing relatedness. Besides this validation, we need to understand the influence of parameters and system settings, to apply the measures optimally to the disambiguation task.

In this evaluation, we follow one main objective: to understand if the DBpedia graph-based relatedness measures that we described capture similarity and/or relatedness between concepts.

The methodology consists in computing the pairwise semantic relatedness with our measures on pairs of words whose relatedness/similarity has been assessed by humans. The performance of the measures is then assessed by computing the Spearman correlation between the pairwise scores obtained by the automatic methods, and the human assessed scores. To this end, we use and adapt three ground truth datasets that have been widely used in related literature.
We first describe the three datasets and the adaptations we perform in order to use them with DBpedia. Then, we describe the settings that we use for the application of the measures with DBpedia. Finally, we report the correlation results for all the methods in various system settings.

Datasets

The most common way of evaluating the performance of relatedness measures is to measure the correlation between system’s and humans’ assessment of pairwise relatedness of words or concepts. We experimented with three such datasets, which are the most commonly used in literature:

R&G One of the oldest and most used datasets that contain human assessment of word similarity is that of Rubenstein and Goodenough[122], which we call R&G. It contains 65 pairs of words together with the overall assessment of humans, gathered from 51 subjects. The users were requested to judge the “similarity of meaning” on a scale from 0.0 to 4.0, where a high score means high similarity.

WordSim353 The second dataset we use is that of Finkelstein et al. [40], known as WordSim353. It contains 353 pairs of words assessed on a scale from 0 to 10 by 13 to 16 human users. In their user study, Finkelstein et al. [40] make no distinction between similarity and relatedness, as while they instruct the annotators to assign a numerical similarity score, they define a score of 10 as words are very closely related, and they also request the users to consider antonyms as similar rather than dissimilar. This has been a source of criticism of the suitability of WordSim353 as a ground truth dataset[4]. However, Agirre et al. [4] split the dataset through another user study in two overlapping parts:

WordSim353-Similarity - containing 203 pairs that the users considered suitable for similarity computation;

WordSim353-Relatedness - containing 252 pairs that the users considered suitable for relatedness computation;

R122 The third dataset we use is very recent and was created by Szumlanski et al. [132] specifically for measuring relatedness. It contains 122 pairs of words, scored within a range from 0.0 (completely unrelated) to 4.0 (very strongly related), each pair being evaluated by 14 to 22 annotators out of a total of 92 participants.
We therefore use 4 datasets in total, 2 of which contain similarity assessments and 2 contain relatedness assessments.

**Dataset Adaptation to DBpedia**

The datasets contain pairs of words rather than pairs of explicit senses. In order to apply them to our DBpedia concept relatedness measures, the words must be assigned the corresponding DBpedia concept. However, this task might pose problems in the case of ambiguous words. In related literature, one adopted solution is to measure the relatedness between all the pairs of possible concepts, and assign the pair of words the maximum similarity among the concept pairs [118]:

$$\text{sim}(w_1, w_2) = \max_{c_1, c_2} \text{sim}(c_1, c_2)$$

This approach does an implicit disambiguation of the two words, in the very limited context of the pair itself. As such, it assumes a high risk of a wrong disambiguation and therefore wrong concept relatedness assessment. Its use is therefore best suitable for small dictionaries, and less so for DBpedia. We therefore perform manual linking between the words and the corresponding most suitable DBpedia concepts. However, to insure objectivity, we follow some simple principles:

- For the words that have exactly one corresponding non-ambiguous DBpedia page, for example *fruit* whose DBpedia page is `dbpedia:Fruit`, we directly assign that resource.

- For the words that have a redirect to a non-ambiguous DBpedia resource we assign the target resource of the redirect link. For example, the word *car* has the corresponding DBpedia resource `dbpedia:Car`, which is redirected to `dbpedia:Automobile`. We assign *car* the resource `dbpedia:Automobile`.

- For the words that have a DBpedia disambiguation page, and among the listed possible meanings the common noun meaning is present, we assign the word the resource of the common noun meaning. For example, the word *magician* has the corresponding DBpedia resource `dbpedia:Magician`, which points to 25 possible meanings, all of them being names of novels, games, series or films, except the common meaning `dbpedia:Magician_(fantasy)`\(^2\). We then assign the word *magician* the resource `http://dbpedia.org/page/Magician_(fantasy)`.

---

\(^2\)The DBpedia definition for `dbpedia:Magician_(fantasy)` is "A magician, which may also be known in various regions as a magic-user, mage, magister, archmage, sorcerer/sorceress, shugenja, witch, wizard, warlock, wu jen, enchanter/enchantress, illusionist, diviner, conjurer, or thaumaturge; depending on the broad contextual
For the words that point to a resource that is ambiguous and among the possible meanings many of them are common nouns that might be suitable in the context, we consider that word polysemous and therefore ambiguous and we do not assign it any resource. This decision removes the subjectivity of the selection of the correct meaning, as we acknowledge that such a choice can only be made in the context of a user study. For example the word *group* has the corresponding DBpedia resource *dbpedia:Group*, which points to more than 50 possible disambiguations. Some of them are *dbpedia:Ethnic_group, dbpedia:Functional_group, dbpedia:Group_(mathematics), dbpedia:Social_group*, to name just a few. There is no resource for the general concept of *group* (i.e., *a number of persons or things considered as a collective unit*\(^3\)). For this evaluation, we remove the pairs containing such words.

For the words that have a corresponding resource in DBpedia, which points to multiple possible disambiguations but none of them is the common meaning of the word (for example the words *journey, grin* and *lad*), we remove from the dataset all the pairs containing them.

There were no words in the datasets we experimented with that had no corresponding resource in DBpedia.

This manual assignment left us with 38 pairs of concepts from the R&G dataset, 157 pairs in WordSim353-Similarity, 175 pairs in WordSim353-Relatedness and 95 pairs of concepts in R122.

**Evaluated Measures**

We evaluate the relatedness measures that we introduced in this section, and in the experiments report we call them as follows:

**ISP** - Inverse Shortest Path, as described in Section 5.5.1.

**Katz** - Katz relatedness, as described in Section 5.5.2, that we use with various setting for the parameter \(\alpha\), and various settings of parameter \(k\);

**PathInfo** - Path Information relatedness as described in Section 5.5.2, that we use with various settings for parameter \(k\).

---

\(^3\)definition of *group* in Collins English Dictionary
**PPR** - Personalised PageRank Cosine Similarity as described in Section 5.5.3, that we use with the damping factor 0.85, as suggested by [4].

Numerically, the difference between the **PathInfo** and **Katz** relatedness as defined above for various $\alpha$ values is shown in Figure 5.3. This figure shows for example, that using Katz relatedness with $\alpha = 0.5$ or Path Info, two paths of length 2 account as much as a path of length 1 in concept relatedness. Katz@0.6 gives 2 paths or length 2 more cumulative weight than a path of length 1, and on the opposite side, 2 paths of length 2 as weighted by Katz@0.4 provide less evidence towards relatedness than one path of length 1. It also shows that the weights of the paths as scored with Katz measures decrease very fast, so that paths of length 5 weight less than 0.2, and longer ones are almost negligible. On the other side, the Path Info computation assigns paths of length higher than 4 almost the same weight, and the weights decrease very slowly beyond this point.

**Experiment Settings**

We compute the semantic relatedness scores between DBpedia concepts by extracting their corresponding 2nd degree ego-networks and then merging them to create the ego-network of the pair. The graph based relatedness measures that we previously introduced are then applied on the pair’s ego-network. We now describe three properties of the ego-network extraction that we experiment with.

**DBpedia Stop-URIs** One problem we face in the DBpedia graph is that concepts are often linked with Wikipedia administrative categories (e.g., Category:Pages_containing_
deleted_templates), with categories referring to words’ etymology (e.g., Category: Latin-loanwords), and with very generic LOD concepts (e.g., owl:Thing, owl:Class, skos:core#Concept, etc.). These nodes create a range of shortcuts between concepts that do not reflect relationships we are interested in. For example, if all concepts are considered an instance of skos:core#Concept, then there will be a path of length two between any two concepts.

To overcome this, we automatically created a list of stopURIs that tries to cover this type of nodes. We created that list by navigating the higher levels of the category hierarchy rooted at the node Category:Contents. For instance, the skos:broaderOf categories of this root category are: Category:Articles, Category:Featured_content, Category:Glossaries, Category:Wikipedia_administration, and so on. We manually inspected each of these branches for the depth up to which all child categories are stopURIs themselves. For instance, a child of Category:Wikipedia_categories is Category:All_redirect_categories which we also consider a stopURI. Also, we consider all lists and glossaries as stopURIs. This semi-automatic process identified 865 stopURIs, which we made publicly available as it was generated in May 2012. In our experiments, we analyse the influence of these stopURIs, and report our findings in Section 5.5.4.

**Category and Concept Merge** The second factor that we analyse is that of DBpedia concepts and Wikipedia Categories bearing the same meaning. For example, the concept dbres:United_States and the category dbres:Category:United_States. The two resources have different properties, one referring to the properties of the United States like dbpedia-owl:largestCity, and the other one to the properties of the category (the hierarchical properties like skos:broaderOf). In our experiments, we use two settings: (i) in one case we automatically merge the category and the concept nodes into only one concept node which inherits all the relations from both source nodes as seen in Figure 5.4, and (ii) in the second case we leave the two nodes separate. For the first case, we automatically detect if a DBpedia concept and a Wikipedia Category have the same meaning if the concept has the name `<name>` while the Wikipedia Category has name `Category:<name>`.

**DBpedia Graphs** The third aspect that we vary is concerned with the DBpedia datasets that we extract the concepts and relations from. We experiment with two settings:

---

4http://uimr.deri.ie/sites/StopUris
Types & Categories We extract the 2nd degree ego-networks considering the properties from Wikipedia Categories, the DBpedia Ontology and YAGO. Given a DBpedia concept, its Wikipedia Categories are extracted by using the `dcterms:subject` property. The Wikipedia Categories hierarchy is navigated with the `skos:broader` and `skos:narrower` properties. Given a DBpedia concept, its type in the DBpedia Ontology as well as its type in YAGO are extracted using the `rdf:type` property. Both DBpedia Ontology and YAGO are then traversed using the `rdfs:subClass` and `rdfs:subClassOf` properties.

Types & Categories & Infobox We extract the ego-network considering the properties from Wikipedia Categories, DBpedia Ontology, YAGO and Infobox Properties. Due to the great amount of Infobox Properties, for this setting we created a separate repository containing the aforementioned datasets. Given a DBpedia concept, its ego-network is created by traversing all properties in this repository leading to paths shorter or equal to two hops from the targeted concept.

Results

Concept Pair Connectivity We first analyse how many pairs become connected as the pair ego-networks are extracted from DBpedia. The results are shown in Table 5.1.

The first and most important observation is that the sense ego-network connectivity is strongly dependent on the stop-URIs. Out of the analysed pairs of concepts, only 40%-60% become connected if the stop-URIs are removed. When the stop-URIs are not removed, more than 90% of the pairs are connected. Therefore the stop-URIs are responsible for approximately
### Correlations with Ground Truth Datasets

Now we report the Spearman correlations between the tested measures and the human assessments. These results are shown in Table 5.2.

The first and most important observation is that all measures capture with strong correlation to humans, the semantic similarity between concepts. The actual values reach 0.82 for ISP...
and PPR on the R-G dataset, and are just slightly lower for the other measures. Regarding the other similarity dataset, **WordSim535-Similarity**, the correlations reach 0.75 for ISP and PPR. However, all measures have only a moderate correlation to human judgements of relatedness. ISP and PPR reach 0.50 on the R-122 dataset and 0.44 on the WordSim535-Relatedness dataset.

This difference between ability of capturing similarity and relatedness is quite unexpected, as DBpedia relations have various semantics. As such, it can be argued that our proposed relatedness measures used on the DBpedia graph, are able to capture similarity between concepts, rather than more complex relations. However, this effect is the result of a combination of factors:
• the relations existing in the knowledge base for the tested concepts;

• the used relatedness measures;

The concepts that exist in the used ground-truth dataset refer to common nouns. In DBpedia, concepts that are referred to through common nouns have mostly properties that have to do with their categorisation like `dcterms:subject`, `skos:broader`, `rdf:subClass`. These hierarchic properties are able to capture similarity better than relatedness. The vast majority of properties that refer to various other types of relations than hierarchical such as `dbpedia-owl:capital`, `dbpedia-owl:birthPlace` defined by DBpedia, connect named entities. Intuitively, following paths made up from these types of relations, we could capture relatedness rather than similarity. However, the used datasets contain almost exclusively common nouns, therefore our tests have not traversed these types of properties. This result might also indicate that the properties of concepts that are not named entities are under-represented in DBpedia. Consequently, more research is required in order to decide if the much better performance obtained for similarity datasets is the result of the relatedness measures, is the result of the used ground-truth datasets, or is the result of using the DBpedia knowledge base.

The second important aspect to notice is that the performance of the measures tend to slightly decrease as more paths are considered, and for Katz, as $\alpha$ increases. Katz and PathInfo perform constantly poorer than ISP. As more and therefore longer paths are being considered by incorporating their participation in the Katz and PathInfo, the correlations to human assessment decrease. The same effect is observable in the case of Katz relatedness, where as the value of $\alpha$ increases, giving higher weights to longer paths (see Figure 5.3 on page 107), the correlation values decrease. This is also consistent with the lowest overall scores achieved by PathInfo, whose weights for long paths decrease very slowly, as seen in Figure 5.3 on page 107.

These being the main observations, we now present some other very important insights.

While PPR metric achieves best scores overall, it is closely followed by ISP and Katz $\alpha = 0.4$. The differences between these three methods are not statistically significant as verified with the t-test statistic at a 0.05 significance level.

The decision of considering only the Types and Categories datasets, or also the Infobox properties makes almost no difference. However, our deeper investigation showed this is because the datasets contain mostly common nouns, and in DBpedia, the resources that represent common nouns have very few properties in the Infobox Properties dataset, as their corresponding Wikipedia pages do not have an Infobox. While the results might be different
for relatedness between named entities, this cannot be evaluated with these datasets. For this reason, in Table 5.2, we show the results for the cases when the “Only Types & Categories” is set to “No” only for the first dataset, shaded, and we omit them for the other datasets.

As in the experiments related to pair connectivity, we observe a great impact from the stop-URIs. Their presence in the graphs lowers all methods’ results under all settings, with approximately 0.20. Moreover, whereas the scores of the path-based measures, in the condition when the stop-URIs are removed from the graph decrease very slowly as $k$ increases, they fall abruptly when the stop-URIs are left in the graphs. The stop-URIs have a strong influence on the performance of ISP as well, even if this measure only considers the top shortest path. This is due to the fact that as seen in Table 5.1, stop-URIs “shortcut” the paths between otherwise unconnected pairs, and they also “shortcut” the longer “semantic paths”. Performance of PPR also drops considerably if the stop-URIs are present in the graph. Since this measure depends on all the nodes in the ego-network as opposed to only the nodes on the paths between target concepts, this shows that the stop-URIs actually strongly impact the topology of the ego-networks. The great impact of stop-URIs over these measures suggests that their fully automatic detection is a research direction worth investigating in the future.

As for the influence of merging categories with the corresponding concepts into one node in the analysed graph, it proves to be beneficial and statistically significant ($p < 0.001$ on paired t-test) only when the stop-URIs are removed and only on the relatedness datasets, as it can be observed by comparing the Yes/Yes/Yes to the Yes/No/Yes cases.

To sum up, all methods achieve very strong correlations on the similarity pairs, and moderate correlations on the relatedness pairs. These results prove the measures’ suitability for use in WSD. However, the removal of stop-URIs is crucial. This step ensures that DBpedia can be used as a knowledge base to measure semantic relatedness. Without this step, the performance of previously researched relatedness measures like that of PPR, which has achieved a score of 0.83 on WordNet [4], would fall on DBpedia to as low as 0.53. The removal of stop-URIs recovers the score of this method to up to 0.82.

Because the differences between the methods are very small, we re-evaluate all these relatedness methods in the context of joint word-sense disambiguation with DBpedia in the next section.
5.6. Global Optimisation for Joint Word Sense Disambiguation

We use the previously described semantic relatedness measures for WSD with DBpedia. As shown in the previous chapter, joint disambiguation poses several important advantages over local disambiguation, but it suffers from the computational overhead problem. In the previous section, we defined the used pairwise relatedness measures. Their application scores every pair of senses in a feasible solution. As defined in Section 4.1.1, the objective of the joint disambiguation problem is to optimise a function over the space of all feasible solutions. In this chapter, we use the sum of pairwise relatedness as the objective function. We also describe a branch-and-bound algorithm that finds the global optimum solution, starting from pairwise relatedness scores, without computing all the possible combinations, but by pruning the search space.

5.6.1. Frozen-past wrapper function

As we previously mentioned, joint WSD problem is NP-hard. However, the results we obtained in Chapter 4 clearly show that the performance of joint disambiguation is not significantly lower when disambiguation context contains fewer words. This new insight motivates us to use a frozen-past technique [130] which achieves global approximate optimisation and is the focus of this section.

The idea is to set a threshold on the size of the search space, and only attempt to disambiguate as many words as possible at a time, without overpassing the threshold. The first group of selected words are jointly disambiguated. As the senses of these words are selected, all their other senses are removed. The overall search space becomes smaller, and the next words from the list of remaining words are disambiguated together with the already solved words (but that now have their senses fixed). The disambiguated words are always reused for the disambiguation of the next words in context, to ensure that the overall relatedness is optimised. The process repeats in this manner, until all words in the given context have been disambiguated. We add the words to the disambiguation problem in the ascending order of the number of candidate senses they have. This process is clarified in Algorithm 1.

The size of the search space is the number of feasible solutions and it is computed in line 5-6. If it is larger than the set threshold, then a subset of words is selected for disambiguation, in lines 8-13. These words are jointly disambiguated with any exhaustive search algorithm.
Algorithm 1: FrozenPast()

input : a list target_words of words to be disambiguated and their possible senses
output : a list disambiguated_words or words that were disambiguated and their disambiguated senses

1 disambiguated_words ← []
2 done ← false;
3 repeat
4 problem_size ← 1; problem ← target_words;
5 foreach word w in target_words do
6 problem_size ← problem_size × w.possible_senses.count;
7 if problem_size > threshold then
8 subproblem_size ← 1; subproblem ← [];
9 sorted_words ← sort target_words, ascending by number of senses;
10 foreach word w in sorted_words do
11 subproblem_size ← subproblem_size × w.possible_senses.count;
12 if subproblem_size ≤ threshold then subproblem.add(w);
13 else break;
14 Disambiguate (subproblem);
15 foreach word w in subproblem do
16 w.possible_senses ← w.disambiguated_senses;
17 if w.disambiguated_senses.count = 1 then disambiguated_words.add(w);
18 else
19 Disambiguate (problem);
20 foreach word w in problem do
21 w.possible_senses ← w.disambiguated_senses;
22 if w.disambiguated_senses.count = 1 then disambiguated_words.add(w);
23 done ← true
24 until done = true;
25 return disambiguated_words;

The disambiguated words have their senses “frozen” in line 15. At the next processing of the loop, as these words typically have only one sense, they become part of the next sub-problem, without increasing the search space. The process finishes when the overall problem is small enough and solved in lines 18-23.

It is important to notice that in this implementation, we only consider a word as disambiguated if it has only one sense selected (see lines 17 and 22). This differs from the approach in the evaluation in Chapter 4 because in this setting we only use one sense inventory, DBpedia, so we penalise the extraction of more senses as inability to disambiguate. Still, if more senses are retrieved in a sub-problem, then they are passed to the next sub-problem (see line 16), as the system might be able to better discriminate with the knowledge of this new context.

To decrease the risk of propagating an erroneous early disambiguated word, we sort the words ascending with respect to the number of candidate senses, so that the first sub-problem jointly disambiguates as many words as possible.
In this chapter, we use a branch and bound (B&B) algorithm for achieving an exhaustive search of the global solution of the subproblem. The B&B methodology achieves complete optimisation although it avoids explicitly computing the score for all feasible solutions by pruning the search space. In general it can be used on its own, but in worst case scenario, its performance is still exponential. However, its very good typical performance allows setting a high space threshold, and therefore only perform the frozen past routine on problems that exceed it. As we later detail, the efficiency of B&B is strongly improved if it starts with a \textit{current best} score. To obtain this provisional score, we use simulated annealing. Putting together the simulated annealing and the branch and bound algorithms, in the frozen past wrapper routine, the global optimisation process can be illustrated as in Figure 5.5.

![Simulated Annealing and Branch and Bound](image)

**Figure 5.5.**: The frozen past wrapper routine for disambiguation

In the following we briefly explain our B&B algorithm for exhaustive search.

### 5.6.2. B&B for Joint WSD

Before detailing the B&B approach we use for joint WSD, it is worth defining some notions we work with in this section.

Let us consider we have a set $W$ of $n$ target words that need disambiguation. Each word $w_i \in W$, $i \leq n$ has $m_i$ candidate senses. We introduce here the notion of \textit{relatedness graph}, to denote the graph $G^R(V, E)$ having as nodes all the candidate senses $s_{i,j}$, $j \leq m_i$ of all the words $w_i$. $E$ contains one edges between every two senses $(s_{i,j}, s_{k,h})$, if and only if
they do not correspond to the same word (i.e., $i \neq k$). Every edge in $E$ is weighted by the relatedness between its adjacent senses. Using the toy example we gave earlier in Figure 5.1 of Section 5.2.1, the so-called “condensed” graph on the right side of the figure is what we from now on call relatedness graph. The objective of our optimisation problem is to extract from $G^R$ the maximum edge weight clique.

We also define the concept of a partial solution. We call a partial solution, any combination of senses of a subset of the target words, $W^{PS} \subset W$, that contains exactly one sense for each word in $W^{PS}$. Therefore, given the relatedness graph $G^R(V, E)$, partial solutions of the disambiguation problem correspond to cliques in $G^R$. As already stated, feasible solutions correspond by definition to maximal cliques in $G^R$.

The Branch and Bound Paradigm

The Branch and Bound (B&B) algorithms search the whole space of solutions therefore they find the optimal solution, but they do not explicitly enumerate all feasible solutions. Rather, they prune the search space by approximating an upper/lower bound of a feasible solution, given a partial solution. Then, given a current best score and the approximation, the system decides whether to continue the search for a feasible solution starting from the partial one, or to remove the partial solution and prune out the whole search subspace of the partial solution. The aim of the system is to prune as much as possible and as early as possible.

Branching Mechanism

The main idea is to incrementally build the feasible solutions by branching the partial solutions.

The system starts with an empty partial solution, and in the first step it instantiates a partial solution for each sense of the first considered word. In the next stage, the system considers the second word and creates all partial solutions for the first two words. These new partial solutions are children of the previous one sense partial solutions. This so-called branching process would continue, until one by one, all words have been considered. The result is a tree-shaped structure, whose nodes are partial solutions and leaves are feasible solutions. However, the desideratum is to avoid unfolding the entire search tree. This is achieved by deciding at each step if the partial solution is worth branching, or if it stands no chance of producing a feasible solution with a score better than the current best. This decision is taken based on an upper bound computed for each obtained partial solution, by the bounding function.
The Bounding Function

The bounding process runs for all produced partial solutions. When the branching mechanism creates a new partial solution, a certain function, called the bounding function and denoted by $g$ is called, that computes for the partial solution, the best possibly achievable score starting from it. The bounding function must be defined in such a way that its score approximates that of the objective function from above: $g(PS) \geq f(PS)$. Computing the bounding function $g(PS)$ for any partial solution $PS$, and if the system already knows a feasible solution whose score is $\sigma$, and $\sigma > g(PS)$ holds, then $\sigma > f(PS)$ holds too. Since we are trying to maximise $f$, the feasible solution that produced $\sigma$ is a better solution than any feasible solution formed from $PS$, therefore $PS$ can be discarded, and the feasible solutions in the search space starting from it are discarded without the need of being explicitly enumerated and scored. Otherwise, if $\sigma \leq g(PS)$ holds, $PS$ cannot be discarded so it is added to a queue of live partial solutions that need to be branched.

In our WSD scenario, given a partial solution containing $p$ senses, $p \leq n$, the bounding function estimates a maximum possibly achievable score of a feasible solution that would contain those $p$ senses. This score does not need to correspond to a feasible solution, but to be bigger or equal to the score of any feasible solution containing those $p$ senses. As such we only keep the constraint that the solution must contain $\frac{1}{2}n(n-1)$ edges (the number of edges in a clique of $n$ nodes), including all the edges between the given $p$ senses. This score is easily computed by $g(PS)$ in linear time, as we show in Appendix A.

Finally, putting together the branching mechanism and the bounding function, we obtain the branch and bound algorithm that we now present.

The B&B algorithm for Joint WSD

The order in which we consider words is ascending with respect to the number of their possible meanings, so that each pruned subtree removes as many feasible solutions as possible. Regarding the traversal of the search tree, we are using a best-first search mechanism. This means that we keep a priority queue of live partial solutions, sorted descending by the score of the bounding function, therefore the partial solution with highest chance of producing a highly scoring feasible solution is branched first. The pseudocode of the complete algorithm is presented in Algorithm 2.
Algorithm 2: BranchAndBoundWSD(current_best_solution, current_best_score, SensesPerWord, Pairwise_Sense_Relatedness)

1. \( \text{Live} \leftarrow \text{PriorityQueue}([[]]) \);
2. \( \text{sortedWords} \leftarrow \text{sortWordsByNbOfSenses}() \);
3. \( \text{currentWord} \leftarrow \text{sortedWords}[0] \);
4. \( \text{foreach sense } s \text{ in currentWord.senses do} \)
   \( PS \leftarrow \text{new partial solution}(s) \);
   \( \text{if } g(PS) \geq \text{current_best_score} \text{ then } \text{Live.add}(PS) \);
5. \( \text{while Live} \neq \phi \text{ do} \)
   \( PS \leftarrow \text{Live.pop}() \);
   \( \text{children} \leftarrow \text{branch}(PS, \text{currentWord.next}()) \);
   \( \text{foreach } PS_\text{child in children do} \)
   \( \text{if } PS_\text{child is feasible solution then} \)
   \( \text{score} \leftarrow f(PS_\text{child}) \);
   \( \text{if score} \geq \text{current_best_score} \text{ then} \)
   \( \text{current_best_score} \leftarrow \text{score} ; \)
   \( \text{current_best_solution} \leftarrow PS_\text{child} ; \)
   \( \text{remove from Live all partial solutions with bound smaller than score.} \)
   \( \text{else} \)
   \( \text{bound_child} \leftarrow g(PS_\text{child}) ; \)
   \( \text{if bound_child} \geq \text{current_best_score} \text{ then } \text{Live.add}(PS_\text{child}) ; \)
6. \( \text{return current_best_solution} ; \)

The algorithm receives as parameters an approximate score and a solution, \( \text{current_best_score} \) and \( \text{current_best_solution} \). These values have been obtained by an approximate optimisation search that uses simulated annealing, as described in Section 5.6.3. Lines 4 - 5 create the first partial solutions, each corresponding to a sense of the first processed word. The partial solutions are only added to the \( \text{Live} \) priority queue if their bound is higher than the preliminary best score, in line 6. Therefore, it is crucial that this score is very good, so that line 6 already prunes the search space. Algorithm lines 7 to 19 perform the actual traversal of the search tree. The partial solution with highest bound is branched in line 9. Then, all the obtained solutions are checked if they are feasible solutions in line 11. If this check evaluates to \( \text{true} \), then the score of this feasible solution is computed. If the score is better than the current best, then it becomes the current best and all \( \text{live} \) partial solutions that have a bound smaller than the new current best are pruned. If the check in line 11 evaluates to \( \text{false} \), it means that the child partial solution is not complete and its bound is computed in line 18. If its obtained bound is higher than the current best, then the search subspace rooted at this child partial solution might contain a feasible solution with a score higher than the current best, so it is added to the \( \text{Live} \) priority queue. Otherwise, the algorithm implicitly prunes this partial solution by just passing to the next child in line 10, or to the next partial solution in line 8, without adding the partial solution to the priority queue. When the \( \text{Live} \) priority queue is empty, it means that all search space has
been searched, either explicitly or it has been pruned, and that the current best solution is the
global optimum.

### 5.6.3. Selecting a Preliminary Solution with Simulated Annealing

Simulated annealing is an approximate search algorithm for solving optimisation problems. It has already been used for joint WSD by Cowie et al. [31] andNavigli and Lapata [102]. The idea of the algorithm is to start with a random feasible solution, and than do a random change that leads to another feasible solution. If this move to a random neighbour improves the score, it is always selected. If the random change has a poorer score, it is selected with a certain probability. This probability is function of the score difference, so that very bad moves have less chance of being done. This probability also decreases with time, so that at the early stage of the process, the system is more encouraged to try bad moves than at later stages. The process finishes when the system converges because the probability of change to a worse solution nears 0 and there is no better neighbour. The changes to worse solutions are used in order to prevent the system from ending up in a poor local optimum.

In our joint WSD problem, we create a random feasible solution by selecting a random sense for all target words. Then, a random change to a feasible solution is made by randomly selecting one word, and then randomly selecting for it another sense than the current one. While simulated annealing does not guarantee global optimum, its very good results make it a promising option for producing a good preliminary current best solution for B&B [26].

### 5.7. Sen-Dis: The Semantic Network based WSD Process

In order to evaluate the joint disambiguation with DBpedia, we implemented a system whose disambiguation process is illustrated in Figure 5.6. We are only interested in disambiguating the nouns and noun-phrases of the document. We extract them by using the Stanford CoreNLP toolkit[^5]. After the noun-phrases are extracted, the possible senses of the noun-phrases are retrieved from a Lucene Index[^6] where we have indexed all DBpedia concepts. Appendix B details the format of the index. We perform a preliminary candidate sense filter, also described

[^6]: http://lucene.apache.org/core/
in detail in Appendix C. This candidate filter represents the only stage in the Sen-Dis approach that treats named entities different from common nouns. The idea is that if a phrase is a named entity in the text (as annotated with Stanford CoreNLP), then only candidates that are named entities are considered for disambiguation. If a phrase is not a named entity in text, then all candidates are considered (including named entities), in order to reduce the effect of potential false negatives generated by the named entity annotator.

For the filtered set of candidate senses, we extract their ego-networks as previously described in Section 5.3. At the same time, the noun-phrases are clustered based on their co-occurrence in a reference corpus, as we described in Section 5.4, forming the contexts in which disambiguation will occur. The clusters of noun-phrases, together with the senses and their ego-network representations are then used to compute the pairwise relatedness scores, using the measures we defined in Section 5.5. These scores are then sent to the disambiguation module described in Section 5.6.

5.8. Evaluation and Results

We now present the experiments we carried out in order to evaluate the proposed approach to joint WSD with DBpedia. Our hypothesis is that the graph-based semantic relatedness measures can be used with success for joint WSD with DBpedia. We test this hypothesis by comparing the results of the graph-based disambiguation methods, to the E-WSD approach that we showed to perform better than state-of-the-art gloss-based approaches.
In order to test our hypothesis, we implement the disambiguation system presented in Section 5.7, and whose components have been described throughout this chapter. For comparison, we also tailor E-WSD to work with documents rather than just topics, as we present in Section 5.8.1. Then we ran both E-WSD and Sen-Dis independently over all documents from two ground-truth datasets that have been previously used in related literature. The datasets contain human validated annotations with DBpedia concepts, or Wikipedia articles. We score the disambiguation methods by using the traditional measures of precision, recall and f-measure, and we also use a disambiguation accuracy metric. In the following, we first briefly illustrate the process using E-WSD on top of sense extraction and clustering, and then we describe the used ground-truth datasets and the performance measures we employ. Finally, we report and interpret the results.

5.8.1. E-WSD for Documents

For this evaluation, we are revisiting E-WSD, more exactly ASV+E-WSD described and evaluated in Chapter 4, in the context of topic models. We now evaluate it with clusters of words extracted from a document. As E-WSD is a gloss-based disambiguation method, it has the advantage that besides the senses coherence, their similarity to the entire text can also be captured. We capture it directly in the eigenvalue computation, by adding to the disambiguation context a dummy target word, whose only possible sense is the text itself. An example of the implication of this is shown in Figure 5.7(a). The normal edges show the edges due to the group of target words. The thick edges are edges due to the addition of the full text as a dummy sense. Therefore, the more overlap between the text and the senses in the disambiguation context, the higher the combination’s score. Used with word clustering and DBpedia sense extraction, the whole process is illustrated in 5.7(b).

5.8.2. Evaluated Methods and Settings

We have implemented the system presented in Section 5.7 as well as E-WSD as previously illustrated.

For Sen-Dis we extract the 2nd degree ego-networks of senses by following all the relations from the Types & Categories & Infobox graphs. We also ran the same experiments without the Infobox graph, but there was no significant difference in the results. For all reported
5.8.3. Ground Truth Datasets

In order to evaluate the use of the relatedness measures described in Section 5.5 for WSD, we incorporate them in the branch-and-bound methodology described in the previous section. We are using two datasets of texts that have been manually annotated by humans, the one used by Mendes et al. [89] for the evaluation of DBpedia Spotlight, and the one used by Milne and Witten [95] as made public by Ratinov et al. [116]:

**NYT10** dataset consists of ten excerpts from news articles published by New York Times [89]. Each text has all the meaning bearing phrases annotated with at most one DBpedia resource. This means that in the case of a multiple word phrase, either the whole phrase is annotated with the corresponding DBpedia concept, or some of its words, but not both. For example, the phrase: “annual meetings of the International Monetary Fund” is annotated with `dbres:Annual_Meetings_of_the_International_Monetary_Fund_and_the_World_Bank_Group`, and its subphrase “International Monetary Fund” is not extracted nor annotated separately, although it might arguably be considered of interest. On the other hand, the phrase “Canadian Embassy” is not extracted nor linked, but its tokens “Canadian” and “Embassy” are extracted and linked separately. This means that an algorithm that extracts such arguably relevant phrases, but that are not annotated in this
dataset, is penalised on either precision or recall, or both. However, this dataset is, to the best of our knowledge, currently the best effort towards all-words-manual-annotation of DBpedia/Wikipedia ground truth. Each text has between 20 and 40 annotations, summing up to 270 annotated phrases.

**Aquaint50** dataset contains 50 documents from the AQUAINT corpus, that were used by Milne and Witten [95]. They have been linked and disambiguated to Wikipedia articles by their system, and the results were evaluated using Amazon Mechanical Turk\(^7\). The workers were instructed to assess each Wikipedia article extracted by the system based on its correctness, relevance and helpfulness. Only the ones extracted by their system and are correct, relevant and helpful are part of the ground truth dataset. Thus, this dataset does not try to link all meaning-bearing phrases, but to link the documents in a fashion similar to how Wikipedia pages are linked. Since the annotations refer to Wikipedia articles rather than DBpedia, and the dataset was produced in 2008, we manually updated each link, so that it points to the corresponding DBpedia page.

### 5.8.4. Performance Measures

In order to understand the performance of the relatedness measures and the joint disambiguation algorithm, we evaluate them under two tests.

**Document Annotation Test**  The first test is an “annotation test” and consists of the traditional information retrieval evaluation measures, precision and recall. In this test, we are interested to verify how many of the extracted DBpedia concepts for a text, are correct. We treat each text as a set of disambiguated DBpedia concepts, and we compare the ground truth set of concepts to the disambiguated group of concepts. More specifically:

\[
Precision = \frac{|Sen - Dis \cap \text{Ground Truth}|}{|Sen - Dis|} \\
Recall = \frac{|Sen - Dis \cap \text{Ground Truth}|}{|\text{Ground Truth}|}
\]

where \(Sen - Dis\) is the set of linked and disambiguated concepts by our system, and \(\text{Ground Truth}\) is the set of all the manually annotated DBpedia concepts for the documents.

\(^7\)https://www.mturk.com
We can only apply this test to the NYT10 dataset, because this dataset attempts to link every meaning-bearing phrase.

Besides the correctness of disambiguation, the results of this test are highly influenced by the implicit noun-phrase extraction. For example, the phrase “annual meetings of the International Monetary Fund” is annotated in one document of the ground-truth dataset with dbres:Annual_Meetings_of_the_International_Monetary_Fund_and_the_World_Bank_Group, but its subphrase “International Monetary Fund” is not extracted nor annotated separately. Our system fails to extract the “annual meetings of the International Monetary Fund” phrase, but it extracts its subphrase and links it to dbres:International_Monetary_Fund. The “annotation test” losses recall for the missed dbres:Annual_Meetings_of_the_International_Monetary_Fund_and_the_World_Bank_Group concept, and also a point of precision because of the extra concept dbres:International_Monetary_Fund. A similar situation is the phrase “Canadian Embassy”, which is not extracted nor linked in the ground truth dataset, but its tokens “Canadian” and “Embassy” are extracted and linked separately. This might be considered an issue of noun-phrase relevance to the document, and it is complementary to our work.

In order to eliminate the effect of keyphrase extraction from the results, we also devised a second test that we now describe.

The Noun-Phrase Disambiguation Test  The second test is a “disambiguation test” as it verifies to what extent a noun-phrase whose corresponding DBpedia concept is set by humans, is linked and disambiguated by our system to the same DBpedia concept. The difference from the previous test is that here, the ground truth dataset behaves as a sample of noun phrases that have the correct corresponding DBpedia concept set by humans, rather than as the complete golden standard. As such, as opposed to the “annotation test”, the results obtained at the “disambiguation test” are independent from the part-of-speech tagging for noun phrase extraction. To further clarify the distinction between this test and the “annotation test”, in the case of the “annual meetings of the International Monetary Fund” phrase, the “disambiguation” test does not penalise the system, unless it links this exact phrase to a concept other than the ground truth dbres:Annual_Meetings_of_the_International_Monetary_Fund_and_the_World_Bank_Group.
where $\{\}$ represents the identity function, which evaluates to 1 when the expression inside is true and to 0 when the expression is false. $NP_{\text{GroundTruth}}$ represents the noun phrases that are annotated in the ground truth dataset, and $NP_{\text{Sen-Dis}}$ represents the noun phrases that are annotated by Sen - Dis. $\text{GroundTruth}(NP)$ is the DBpedia concept that the ground truth dataset has assigned to the noun phrase $NP$, and similarly $\text{Sen - Dis}(NP)$ is the DBpedia concept that our system Sen - Dis assigned the same noun phrase, $NP$. As shown in Formula (5.9), this test measures the proportion of noun phrases that have been disambiguated correctly, out of the noun phrases annotated by both the human annotators and the DB_WSD system. This test can be applied to both NYT10 and Aquaint50.

These tests are applied to the groups of words created through the clustering methods presented in Section 5.4. The clustering algorithms produce the number of clusters per document and phrases per cluster as shown in Table 5.3. We also evaluate disambiguation without word clustering, so that all noun phrases extracted from the text are simultaneously disambiguated. The “No Clustering” setting leads to a purely DBpedia knowledge based disambiguation, with complete independence from any other knowledge base.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Louvain Clusters per document</th>
<th>Louvain Words per cluster</th>
<th>Agglomerative Clusters per document</th>
<th>Agglomerative Words per cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYT10</td>
<td>4.4</td>
<td>6.1</td>
<td>11.7</td>
<td>2.3</td>
</tr>
<tr>
<td>Aquaint50</td>
<td>5.4</td>
<td>6.4</td>
<td>13.3</td>
<td>2.62</td>
</tr>
</tbody>
</table>

*Table 5.3:* Number of clusters per document and words per cluster for the two clustering algorithms

### 5.8.5. Results

Table 5.4 shows the results on the NYT10 dataset, for all the evaluated relatedness measures, including E-WSD which was introduced in Chapter 4, for the case when only the hierarchical DBpedia links are considered.
Figure 5.8 visualises the relations between precision and recall for the evaluated methods. It brings to evidence the outliers as well as the skyline methods. In this picture, the effect of the clustering methods is also visible due to the grouping of the values for each clustering method.

![Figure 5.8: Precision vs Recall on the NYT10 dataset](image)

Table 5.4.: Annotation test results for NYT10 dataset, using only the hierarchical links; P - precision; R - recall; F - F-measure

<table>
<thead>
<tr>
<th></th>
<th>Hierarchical</th>
<th>Louvain</th>
<th>No Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>EWSD</td>
<td>61.10%</td>
<td>56.27%</td>
<td>58.11%</td>
</tr>
<tr>
<td>ISP</td>
<td>64.66%</td>
<td>55.50%</td>
<td>59.39%</td>
</tr>
<tr>
<td>PPR</td>
<td>63.16%</td>
<td>55.64%</td>
<td>58.77%</td>
</tr>
<tr>
<td>Top 5</td>
<td>64.36%</td>
<td>55.50%</td>
<td>59.30%</td>
</tr>
<tr>
<td>Top 10</td>
<td>64.69%</td>
<td><strong>56.30%</strong></td>
<td>59.85%</td>
</tr>
<tr>
<td>Top 15</td>
<td>63.98%</td>
<td>55.88%</td>
<td>59.29%</td>
</tr>
<tr>
<td>Top 20</td>
<td>63.90%</td>
<td>55.55%</td>
<td>59.08%</td>
</tr>
<tr>
<td>Top 5 Path Info</td>
<td>64.66%</td>
<td>55.50%</td>
<td>59.43%</td>
</tr>
<tr>
<td>Top 10</td>
<td>64.69%</td>
<td><strong>56.30%</strong></td>
<td>59.85%</td>
</tr>
<tr>
<td>Top 15</td>
<td>64.69%</td>
<td><strong>56.30%</strong></td>
<td>59.85%</td>
</tr>
<tr>
<td>Top 20</td>
<td>64.69%</td>
<td><strong>56.30%</strong></td>
<td>59.85%</td>
</tr>
<tr>
<td>Katz and 1</td>
<td>64.66%</td>
<td>55.50%</td>
<td>59.43%</td>
</tr>
<tr>
<td>Katz and 5</td>
<td>64.99%</td>
<td><strong>56.30%</strong></td>
<td>59.98%</td>
</tr>
<tr>
<td>Katz and 10</td>
<td>64.99%</td>
<td><strong>56.30%</strong></td>
<td>59.98%</td>
</tr>
<tr>
<td>Katz and 15</td>
<td>64.99%</td>
<td><strong>56.30%</strong></td>
<td>59.98%</td>
</tr>
<tr>
<td>Katz and 20</td>
<td>64.66%</td>
<td>55.50%</td>
<td>59.43%</td>
</tr>
<tr>
<td>Katz and 50</td>
<td>64.41%</td>
<td><strong>56.30%</strong></td>
<td>59.72%</td>
</tr>
<tr>
<td>Katz and 100</td>
<td>64.69%</td>
<td><strong>56.30%</strong></td>
<td>59.85%</td>
</tr>
<tr>
<td>Katz and 500</td>
<td>64.41%</td>
<td><strong>56.30%</strong></td>
<td>59.72%</td>
</tr>
</tbody>
</table>
Regarding the “Disambiguation test”, the results of the accuracy measure are shown in Table 5.5.

The results we obtained clearly support our hypothesis. It is mostly supported by the results all the measures obtained at the “Disambiguation Test”, and shown in Table 5.5. These scores are very promising given that supervised systems that learn to disambiguate to Wikipedia can achieve up to 96% accuracy [95].

The DBpedia measures obtain higher results than E-WSD on all tests except for the “Disambiguation test” on the “No Clustering” setting, as seen in Table 5.5. This means that the graph-based measures are more vulnerable to the noise in unrelated senses, than E-WSD. The advantage of E-WSD when no clusters are used might be caused by its adaptation to documents shown in Section 5.8.1.

**Effect of word clustering** With respect to word clustering, our results are strongly in its favour. For all methods and all tests, the performances when disambiguating words grouped by clustering are higher than when disambiguating all words. Actually, as seen in Table 5.5,
clustering improves disambiguation accuracy of the DBpedia graph based methods by more than 15%. Figure 5.8 also illustrates the differences in precision and recall due to the various clustering settings. Regarding the two clustering algorithms, disambiguation with hierarchical clustering achieves overall lower F-measure scores due to poor recall even if it achieves the higher precision and accuracy. Figure 5.9 shows that for all methods, the hierarchical clustering links the smallest number of noun phrases. Deeper investigation revealed that hierarchical clustering, used on this data with the 0.8 dendrogram cutting threshold produces more clusters that contain only one word, than Louvain does. In such clusters, if the word needs disambiguation, our method does not attempt to disambiguate it in the lack of context, leading to a poorer recall.

The Louvain clustering method creates bigger clusters, therefore it achieves a higher recall overall than the hierarchical clustering. While its precision drops a bit, its F-measure is still significantly higher than that obtained with hierarchical clustering. The noise caused by the bigger clusters of Louvain affects its accuracy, which is lower for this clustering method. The negative effect of noise due to the ego-graphs of unrelated concepts is mostly evidenced by the poorest accuracy under the “No Clustering” setting.

![Graph showing number of noun phrases per document](image)

**Figure 5.9:** Number of noun phrases per document extracted by Sen-Dis and annotated by humans, on the NYT10 dataset

These results show that the decision between small but accurate clusters, or big clusters with higher recall should be taken based on the application requirements. An important difference between the clustering methods is the fact that Louvain is non-parametric as opposed to the
hierarchical clustering method that requires a cutoff level. In this evaluation, we report results after we set the level to 0.8. Higher values of this parameter will produce bigger clusters. However, for bigger clusters we prefer to use Louvain for its non-parametric advantage.

**Comparison between the Relatedness Measures** One of the first important things to notice is that all the graph-based methods achieve very similar accuracy results, while the other scores vary slightly. Still a couple of trends can be isolated. First is that the ISP method is most sensitive to clustering, a characteristic visible in Figure 5.8. This reinforces ISP’s sensitivity to noise that was also noticeable in its high score variation under the influence of stop-URIs in Section 5.5.4 on page 103.

Second, the methods that consider more paths tend to peak in performance somewhere around 10-15 paths, a trend visible in both “Annotation test” and “Disambiguation test”. For the Katz method, there is no evidence to support the choice of a value of $\alpha$ over another, among the three tested values. PathInfo and PPR perform slightly, but not significantly worse than Katz. PPR achieves the poorest results on the Annotation test, but the difference is not statistically significant at significance level $\alpha=0.05$.

E-WSD’s accuracy results are very close to the ones we obtained in the previous experiments with E-WSD, presented in the Chapter 4. This shows that this method is very robust and can be used with success when little other knowledge about senses is available. However, except for the “No Clustering” case, E-WSD obtains the worst overall results in both annotation and disambiguation tests. This comes to reinforce that the existence and exploitation of a unifying semantic network greatly improves the chance of good disambiguation, as compared to gloss only based methods.

**Comparison to Related Work**

DBpedia Spotlight achieved an F-score of 56.0% on the NYT10 dataset under the best configuration and 45.2% without any configuration[89]. The authors also report results obtained by other systems on the same dataset, including proprietary software like Zemanta8, OpenCalais9, Alchemy10 and The Wiki Machine11. The last one obtained the highest F-score of 59.5%, with a recall of approximately 65% and precision of 55%[89]. The other ones obtained much lower

---

10 [http://www.alchemyapi.com/](http://www.alchemyapi.com/)
F-scores: Zemanta - 39.1%, OpenCalais - 16.7% and Alchemy 14.7%. These values are low due to poor recall, constantly under 25%, with a higher precision for Zemanta which achieved more than 80%. Therefore these proprietary systems are characterised by a very conservative approach, aiming for high precision at the cost of very low recall.

Regarding the accuracy, Mendes et al. [89] report 73.39% for the unsupervised version of DBpedia Spotlight, with Wikipedia article text used as ground truth. The semi-supervised version achieved 80% accuracy. This is very close to the accuracy obtained by Sen-Dis in the “No Clustering” setting, which is based purely on DBpedia graph structure. E-WSD, when used with the “No Clustering” setting, achieves 80% accuracy on Aquaint50 and 85% accuracy on NYT10, significantly more than the unsupervised DBpedia Spotlight.

Furthermore, for comparison, it is worth mentioning that all the semantic relatedness measures we evaluated under the B&B disambiguation methodology obtained at least two percent higher F-score than the best DBpedia Spotlight configuration. Also, for the Louvain clustering and the No clustering settings, except for E-WSD, all measures achieved higher F-measure than the best previous work - The Wiki Machine, as reported by Mendes et al. [89]. The Wiki Machine is very little documented, and a recent scientific paper about it [134] describe it as a fully supervised system for linking to Wikipedia.

The results of Sen-Dis are better than those achieved by the related state-of-the-art. Even very basic measures like the ISP achieve very good results. We stress that the disambiguation stage of the process uses only the DBpedia graph knowledge about concepts, as opposed to related approaches which exploit the link structure and vector space representation of the whole Wikipedia.

5.9. Conclusion

To sum up, in this chapter we investigated how graph-based measures perform on DBpedia for WSD. We adapted one of the oldest such measures, Rada et al. [114]’s, we adapted Katz [68] centrality measure in a way close to Nunes et al. [109], and also Stephenson and Zelen [125]’s information centrality. We also reimplemented Agirre and Soroa [3]’s personalised PageRank approach, that was initially created with WordNet in mind. We then incorporated these measures in the branch-and-bound methodology and evaluated their performances. The results strongly indicate that the structure of the DBpedia semantic network can be used with success for WSD. The analysis of the graph properties like the length of paths between concepts
reveals knowledge that, when used for WSD, helps systems perform better than those that use more complex knowledge sources. Clustering of words based on statistical co-occurrence greatly improves the results, by focusing the context of disambiguation and reducing the influence of noise.

In this chapter, we revisited E-WSD, the method we proposed in Chapter 4 for gloss-based joint disambiguation. The results we obtained reinforce that this method can compete with much more elaborate approaches, obtaining similar or even superior results to related work. Thus although it performs generally worse than the semantic-network based methods, it provides a good solution in scenarios where there is scarce knowledge about the senses apart from their glosses.

We also investigated the performance of graph-based semantic relatedness measures on the DBpedia knowledge base, on ground truth datasets for both relatedness and similarity. Interestingly, we found a significantly higher correlation between the scores of humans and our measures, on the similarity pairs. This comes to show that DBpedia graph still falls short from expressing the connections that humans recognise between concepts. We showed that the existence of stop-URIs in DBpedia, as “inheritance” from Wikipedia, perturbs the local properties of the semantic network and their removal greatly improves the performance of all tested measures. Actually, their tolerance in the semantic network sometimes renders the path-based semantic relatedness measures unusable.

The joint word-sense disambiguation has the advantage of maximising the global relatedness between the targeted concepts, but it suffers from being computationally expensive. In this chapter, we addressed this issue by providing a detailed branch-and-bound methodology, a popular solution to solving NP-hard optimisation problems. We also use a frozen-past wrapper function that further improves the computational fingerprint of the joint disambiguation, by achieving an approximate optimisation. However, the very good results obtained suggest that this technique has very little negative impact on the disambiguation results.

A very interesting future direction is to research ways of simultaneously performing the clustering and the joint disambiguation. Currently the disambiguation takes place after the clusters have been decided by the clustering algorithm. Therefore, if clustering misplaces a word in the wrong context due to lack of correlation between the reference text corpus and the targeted text, the word stands very little chance of being disambiguated correctly. Barzilay and Elhadad [11]’s algorithm for disambiguation in the context of lexical chains achieves both steps simultaneously, but theoretical analysis shows that a reimplementation of their algorithm would be very inefficient when dealing with cliques. However, the prospective of a process
that optimises clustering and disambiguation at the same time, seems feasible and desirable for eliminating any inconsistencies between the two steps.
Chapter 6.

Using the DBpedia Graph for Unsupervised Topic Labelling

In the previous chapter, we showed that connectivity between concepts in the DBpedia semantic network can be used to assess the relatedness between them. In this chapter, we further investigate the correspondence between structural properties of concepts in semantic networks, and their semantic properties. We adapt measures of node centrality from social network analysis to our semantic topic networks, in order to find the concepts that are most suitable for labelling a group of related concepts. The intuition is that a concept that denotes the theme or the domain of a topic model is strongly related in the semantic network to all the topic’s disambiguated concepts. We build on top of the work in previous chapters, and integrate our previously proposed word-sense disambiguation methods with the approach to topic labelling that we introduce here. We evaluate our proposed centrality measures for topic labelling as well as for document labelling. Our experiments show that the proposed measures have a great potential and perform much better than text-based state-of-the-art approaches to topic labelling.

6.1. Introduction

Linking text to external semantic networks as shown in the previous chapter, brings additional benefits than just the provision of definitions for the words in text. Text indexing for information retrieval [49, 87], document similarity and document clustering [61] are some of the most common domains that benefit from linking and disambiguation. In this chapter, we look into another direction, that is mostly related to helping humans interpret and make sense of topic

\footnote{parts of the research presented in this chapter have been published as [64]}
models. We rely on the insights gained in the work presented in the previous chapter, that the way concepts are connected in DBpedia’s semantic network reveals strength of association between them that is not necessarily explicitly stated in the network.

In texts, as well as in topic models, the domain or theme is often not explicitly stated but it is implicit and requires human interpretation. However, semantic networks often contain the concept that names the implicit theme / domain of the topic. Our intuition is that this concept, that we call label, is characterised by a strong structural relatedness to the concepts in the targeted topic or document. The idea of finding nodes in networks that are strongly connected to the other nodes has been widely studied and perfected in the social networking domain. The property of a node of having strong connectivity to the other nodes in the network is commonly known as centrality. In this chapter, we inherit and adapt centrality measures used in social network analysis, in order to identify the concepts from the background semantic network that are suitable for labelling topic models.

One of the most popular approaches for identifying the subject matter of a collection of documents is to determine its inherently addressed topics, through methods like latent semantic analysis, Latend Dirichlet Allocation, and others that we presented in Section 2.3 of Chapter 2. Broadly speaking, such topics consist of groups of co-occurring words, ranked by their relevance. Such models are largely used in the domain of text analysis for summarising big document corpora.

Typically users have then to interpret these sets of words in order to label the underlying concepts for further processing and classification. Labelling in this context refers to finding one or a few single phrases, or better concepts, that sufficiently describe the topic in question. This can become a cumbersome task when a corpus is summarised by some hundreds of topics. In this light, automatic topic labelling becomes an important problem to solve in order to support users in their task to efficiently and conveniently analyse, understand and explore document collections. Besides that, it further promises benefits for web search engines, as it allows clustering groups of words under the same umbrella term.

This chapter describes an approach to automatically extract topic labels by linking the inherent topics of a text to concepts found in DBpedia and mining the resulting semantic topic graphs. Our aim is not only to find a good label itself, but also to integrate the topic into a knowledge base to support subsequent exploitation and navigation of related concepts. An important aspect of our work is therefore to relate a topic label with a URI identifying a concept, which opens the way for facilitating knowledge exploration in DBpedia – and far beyond, based on its rich linkage within Linked Open Data project.
We argue that current approaches for topic labelling based on content analysis capture the essence of a topic only to a limited extent. Mainly, because they do not focus on the structure behind the concepts, nor on the navigation and exploration of these topics. We hypothesise that concepts co-occurring in the text are also closely related in the DBpedia graph. Using graph centrality measures, we are able to identify the concepts that are most likely to represent the topics and are therefore suited to label them. Our contribution can be summarised as follows:

1. We propose a novel approach for topic labelling that relies only on structured data. The method does not require any pre-processing and can thus be run directly on-line against queryable knowledge bases like DBpedia.

2. The approach is suited for finding a good label and for integrating the topic into a knowledge base to support subsequent exploitation and navigation of related concepts.

3. We show that graph-based algorithms can be used with success to label topics and that they provide richer knowledge than purely text-based methods.

4. We present a thorough comparative evaluation, based on human judgements about the quality of labels, collected through a user study.

In Section 6.2 we briefly overview the overall framework that comprises the topic modelling proposed in this work. Based on a motivating example, we formalise the problem statement and introduce the general principle of our solution. We present our approach for graph-based topic labelling in Section 6.2.3. We examine particular important aspects of our approach and compare it to the standard text-based approach in terms of precision and topic coverage in Section 6.6. This evaluation is based on the results from a user study involving texts from three different document corpora. Finally, we conclude in Section 6.7.

6.2. Overview

In this section, we motivate our graph-based approach, formally define the problem that this work focuses on, and also define the main notions used throughout this chapter.

6.2.1. Motivating Example

An intuitive approach for labelling topics represented by a set of concepts $C^g$ is to determine a minimum spanning tree encompassing all $C \in C^g$ from a hierarchical knowledge base.
The least common subsumer in such a tree could then be chosen as a label. However, most topics, obtained either through probabilistic topic modelling, or co-occurrence based clustering, contain related words that do not necessarily refer to concepts of the same type. As such, the least common subsumer of these concepts would most often produce very generic terms, close to the root of the overall hierarchy of concepts. Consider a topic \( \theta \) described by [patient, drug, hospital, health, professional ...]. All top five words come from very different branches of a standard knowledge base like WordNet. In this case, the least common subsumer is the very root of the WordNet hierarchy: Entity. Similarly, in DBpedia’s structure of categories, the least common subsumer is Life. However, a considerably good label would be Health itself, Healthcare or even Medicine. Healthcare is the least common subsumer of patient, drug and hospital, and a child of Health. Medicine, however, is only subsuming drug.

In order to identify good labels we can thus not rely on the simple least common subsumer. This motivates us to exploit graph specific methods on graph-based knowledge repositories, in our case DBpedia. The example also illustrates the main challenges of this approach, which we formalise next.

### 6.2.2. Problem Statement

In this chapter, we consider the task of topic labelling independent of the way the topics have been linked and disambiguated to DBpedia concepts. We formulate the problem as follows: Let \( C^\theta \) be a set of \( n \) DBpedia concepts \( C_i, i = 1, \ldots, n \), that correspond to a subset of the top-\( k \) words representing one topic \( \theta \). The problem is to identify the concept \( C^* \) from all available concepts in DBpedia, such that the relation \( r(C^\theta, C^*) \) is optimised. Thus, the main challenges are:

1. to extract an appropriate set of concepts from DBpedia as candidates for \( C^* \), and
2. to define \( r \), which quantifies the strength of the relation between the concepts \( C_i \in C^\theta \) and \( C^* \), in a way resulting in topic labels that are meaningful for humans.

We propose to extract a good candidate set by extracting a topic graph \( G \) from DBpedia consisting of the close neighbours of concepts \( C_i \) and the links between them. Then, we investigate how to define the relation \( r \) by analysing the conceptual graph of DBpedia underlying \( G \). We adopt principles from social network analysis to identify in \( G \) the concepts that most accurately label a topic \( \theta \).
6.2.3. Approach Overview

Conceptually, our approach to topic labelling, instantiates the Kanopy process (see page 12). The first step is the linking and disambiguation of the topic words. The resulted concepts are used in order to extract the topic graph. Novel centrality measures are applied on the topic graph in order to extract suitable topic labels. Figure 6.1 illustrates these stages.

As we dealt with the DBpedia linking and disambiguation in the previous chapters, in the following we focus on the Topic graph extraction and Focused centrality computation stages.

6.3. Topic Graph Extraction

The topic graph $G$ is a union of the senses’ ego-networks of one topic. Topic graph $G$ is passed as the result of the graph extraction phase to the actual graph-based labelling step. If we use Sen-Dis presented in Chapter 5 for disambiguation, the topic graph is the graph produced by the optimal disambiguation solution. However, for sake of generality we do not consider the two approaches dependent, and the topic graph can be created by extracting different semantic relations from DBpedia rather than the ones extracted for disambiguation.

**Definition 5.** Let $C^\theta = \{C_1, \ldots, C_n\}$ be the set of DBpedia concepts corresponding to the disambiguated senses of the words in the latent topic $\theta$. Let $R^\theta$ be the set of interest types of semantic relations defined in DBpedia. Let $C_i^k = (V_i, E_i, C_i, R_i^\theta)$ be the sense ego-graph of degree $k$ corresponding to $C_i$, $\forall i \in 1, \ldots, n$. Then, $G = (V, E, C^\theta, R^\theta)$ is the topic graph or topic network of $\theta$, if $V = \bigcup V_i$, and $E$ is the set of undirected edges between nodes in $V$, that correspond to the DBpedia relations $R^\theta$. The concepts $C_i \in C^\theta$ are called the seed concepts or seed nodes of $G$. 
Definition 6. Let $G = (V, E, C^\theta, R^\theta)$ be the topic graph of $\theta$. Let us consider $G$ contains $c$ connected components, denoted $G_j$, so that $G_j = (V_j, E_j, C_j^\theta, R_j^\theta)$, $j \in [1, c]$. It follows that $\bigcup C_j^\theta = C^\theta$, and $\bigcap C_j^\theta = \emptyset$. Then we define a core topic graph any connected component $G_j$ for which $|C_j^\theta| > 1$. The seed concepts that belong to a core topic graph are called core concepts.

By Definition 6, we restrict our analysis only to the topic concepts whose ego-networks become connected to other topic concepts’ ego-networks. The concepts whose ego networks do not merge with any other concept’s ego network are not considered further for labelling. For example, let us consider the topic [energy, atom, electron, quantum, classic], whose words have been linked and disambiguated to DBpedia concepts: \{dbpedia:Energy, dbpedia:Atom, dbpedia:Electron, dbpedia:Quantum, dbpedia:Classical_studies\}. In this example, most likely, the disambiguation of the word classic is wrong. As shown in Section 6.6.1 of Chapter 5, concepts that are related to each other are much more likely to become connected in the DBpedia graph after 2 hops, than concepts that are not related. This means that concepts that remain isolated and do not become connected to the other seed concepts in the core topic graph are less related to the other seed concepts, and probably they have resulted from wrong disambiguation. By using only the core topic graph, we reduce the influence of these concepts. Figure 6.2 illustrates the sense ego-networks in the previous example, their obtained topic graph and the core topic graph.

When a topic graph has multiple core components, we simply consider them as distinct topics. In our experience, usually the topic graphs have only one core component, and the concepts that are not connected to it are isolated. Therefore, for simplicity, in the remainder of this chapter we assume only one core component, unless specified otherwise.

Given these definitions, we address the first challenge listed in Section 6.2.2. We consider the set of label candidates for a topic as the set of all concepts in the core topic graph. The second problem, to define a measure for assessing the goodness of all candidate topics, is solved by applying adapted graph centrality measures on the core topic graph. This is based on the assumption that a good topic label should be a central node in the topic graph, with respect to the seed nodes. We discuss the benefits of different centrality measures, the reasons for adapting and the resulting formulae for computing them in the next section.
6.4. Topic Labelling with Graph Centrality Measures

As our graph is a semantic graph, we hypothesise that nodes that play important structural role in the graph also have an important semantic relationship to the seed concepts. We select our candidate labels from these nodes. As previously discussed, relying upon subsumption relationships tends to produce very generic topic labels. Therefore, we analyse centrality measures that might capture the properties of good topic labels.

Centrality measures are a well-known concept in (social) network science. They are used to identify nodes (or actors) that are most important (and thus, central) for the network, an objective in line with our own requirements. Different criteria for importance, suitable for different purposes and scenarios, have led to a range of centrality measures proposed in the literature [107]. Two of the most popular ones are the closeness centrality and betweenness centrality.
centrality. These two centrality measures rely on shortest paths only. The assumption that the spread of information is best modelled by the use of shortest paths has been questioned [106, 125]. Some alternatives have been suggested, which consider all paths in a network rather than just the shortest paths. Two such measures are information centrality [125] and random walk betweenness centrality [106].

Closeness centrality and information centrality rely on a common intuition that nodes are important in a network if they lie close to all other concepts. Betweenness centrality and random walk-between centrality rely on another common intuition, that nodes are important in a network if they facilitate the flow between the nodes of the network.

In general, all centrality measures compute the importance of a node with respect to all other nodes. This means that given a target node, all network nodes contribute with the same weight to its score. However, in the case of topic labelling, we are particularly interested in the seed concepts, as it is their combination that defines the topic. All our preliminary experiments confirmed that centrality measures that consider all graph nodes equally important, fail to capture the essence of the topic. Consequently they result in top central concepts that are too generic and often just vaguely related to the seed concepts.

We therefore propose to adapt the centrality measures so that they focus on the seed nodes, rather than on all the nodes of the graph. We call these adapted measures focused centralities. This focus on seed nodes reduces the impact that broad concept nodes have due to their high degree and that of dense local clusters due to their sheer size. Table 6.1 shows the top-5 most central nodes as computed with various centrality measures, including the node degree as a trivial baseline. It shows that the focused centralities determine, in comparison to their non-focused counterparts and node degree, concepts that are more related to the seed concepts.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Degree</th>
<th>Closeness Centrality on all graph (CC)</th>
<th>Focused Closeness Centrality (fCC)</th>
<th>Focused Information Centrality (fIC)</th>
<th>Betweenness Centrality on all graph (BC)</th>
<th>Focused Betweenness Centrality (fBC)</th>
<th>Focused Random Walk Betweenness (fRWB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Government</td>
<td>Economic problems</td>
<td>Monetary policy</td>
<td>Economic problems</td>
<td>Government</td>
<td>Economic problems</td>
<td>Economic problems</td>
</tr>
<tr>
<td>2</td>
<td>Politics by issue</td>
<td>Government</td>
<td>Money</td>
<td>Economics</td>
<td>Economic problems</td>
<td>Public economics</td>
<td>Debt</td>
</tr>
<tr>
<td>3</td>
<td>Economic Problems</td>
<td>Debt</td>
<td>Economic problems</td>
<td>Debt</td>
<td>Decision theory</td>
<td>Public economics</td>
<td>Debt</td>
</tr>
<tr>
<td>4</td>
<td>Economics</td>
<td>Policy</td>
<td>Inflation</td>
<td>Monetary economics</td>
<td>Politics by issue</td>
<td>Monetary policy</td>
<td>Mathematical finance</td>
</tr>
<tr>
<td>5</td>
<td>Scientific modeling</td>
<td>Politics by issue</td>
<td>Economics</td>
<td>Money</td>
<td>Policy</td>
<td>Mathematical finance</td>
<td>Mathematical finance</td>
</tr>
</tbody>
</table>

Table 6.1: Top-5 most central concepts in the ego-network of topic with DBpedia concepts: [Government, Economy, Policy, Money, Unemployment, Income, Inflation, Interest, Interest rate, Demand (economics), Fiscal policy]

In the remainder of this section, we introduce our four proposed focused centralities: focused closeness centrality (fCC), focused information centrality (fIC), focused betweenness
centrality (fBC), and focused random walk betweenness (fRWB). We use the example in Table 6.1 throughout this section to illustrate our argumentation.

### 6.4.1. Focused Closeness Centrality

The *focused closeness centrality* is an adaptation of the traditional closeness centrality. The idea behind closeness centrality is that a node is important if it lies close to all of the other nodes in the network. In the context of topics, nodes with high closeness centrality indicate concepts that have short paths to all other concepts of the topic graph.

The average shortest distance $l_i$ from a node $i$ to all other nodes in a graph $G = (V, E)$ is computed as

$$ l_i = \frac{1}{|V|} \sum_{v_j \in V} \text{dist}_{ij} $$  \hspace{1cm} (6.1)

where $\text{dist}_{ij}$ is the length of the shortest path between nodes $v_i$ and $v_j$. The closeness centrality $CC$ is calculated as the inverse of this average:

$$ CC_i = \frac{1}{l_i} = \frac{|V|}{\sum_{v_j \in V} \text{dist}_{ij}} $$  \hspace{1cm} (6.2)

In our labelling problem, the concepts that would intuitively make good labels, need to have short paths to the seed nodes. Therefore, we adapt the closeness centrality and propose the *focused closeness centrality* $fCC$ that we compute as:

$$ fCC_i' = \begin{cases} \frac{\sum_{c_j \in C^\theta} n \text{dist}_{ij}}{n-1} & v_i \notin C^\theta; \\ \frac{\sum_{c_j \in C^\theta} n-1 \text{dist}_{ij}}{n-1} & v_i \in C^\theta; \end{cases} $$  \hspace{1cm} (6.3)

where $n$ is the number of seed nodes in $G$. Note that if $v_i \in C^\theta$, there are only $n-1$ other seed nodes in $C^\theta$ and thus we use $n-1$ as numerator.

The effect of the focusing on the seed nodes, can be seen in Table 6.1, where the top-5 most central concepts as scored by $fCC$ are more specific than the top-5 concepts scored by $CC$ on the whole graph. The $fCC$ top-5 concepts are also more related to the topic. Another property
of this measure is visible, that is that $fCC$ gives high scores to the actual concepts of the topic. In the example, two out of the top-5 labels are seed concepts of the topic: *Money* and *Inflation*.

### 6.4.2. Focused Information Centrality

The core idea behind information centrality is that the *information* of a path $p$ is inverse proportional to its length, $\text{length}(p)$.

$$I_p = \frac{1}{\text{length}(p)}$$

In Chapter 5, this centrality inspired the *PathInfo* measure for semantic relatedness. The same idea holds, that the information between two nodes can be computed as the sum of the path informations of all paths connecting them. For sake of completeness, we restate here the formula, similar to Formula (5.5) on page 101:

$$I_{ij} = \sum_{p \in P_{12}} \frac{1}{\text{length}(p)}.$$

(6.4)

Stephenson and Zelen [125] use this definition of graph information transfer between two nodes, in order to measure a node’s graph centrality. They therefore define information centrality as the harmonic mean of the information between the target node and all the other nodes in the graph. Therefore, the information centrality $fIC_i$ of node $v_i$ is:

$$IC_i = \frac{|V|}{\sum_{j=1}^{n} \frac{1}{I_{ij}}}.$$

(6.5)

As we limit the nodes of interest to the seed nodes, we define the focused information centrality $fIC$ as:

$$fIC_i = \begin{cases} \frac{\sum_{c_j \in C^0}^{n} 1/I_{ij}}{\sum_{c_j \in C_{i} \setminus C^0}^{n-1} 1/I_{ij}} & C_i \notin C^0; \\ \sum_{c_j \in C^0 \setminus C_i}^{n} 1/I_{ij} & C_i \in C^0; \end{cases}$$

(6.6)
In comparison with $fCC$, $fIC$ measure gives high scores to concepts that do not necessarily have very short shortest paths to the seed concepts, but also many longer paths. Usually, nodes that are connected to the target nodes by *many* longer paths, are concepts that have many neighbours, therefore high degree. In Table 6.1, we notice that high degree nodes like *Economic problems* and *Economics* are the highest scored concepts, by $fIC$. In Figure 6.3, we also notice that $fIC$ gives high weights to the seed nodes, comparative to the other illustrated methods. Moreover, it is the only measure that scores high the concepts *Monetary economics* and *Macroeconomics and monetary economics* (see Figure 6.3(d)), whose scores are negligible by the other measures.

### 6.4.3. Focused Betweenness Centrality

The idea behind betweenness centrality is that a node is important if it facilitates the flow of information between other nodes in the graph. In a semantic network, nodes with high betweenness centrality are the nodes that establish short connections between the other nodes in the graph.

These properties intuitively recommend themselves for identifying labels. However, betweenness centrality is biased towards nodes with high degree, or nodes that are central in large local groups of nodes. This bias is visible in the example of Table 6.1, where two of the top-5 concepts scored with betweenness centrality ($BC$) over the whole graph are also in the top-5 highest degree nodes.

Betweenness centrality $BC$ of a node $v_i$ in a graph $G = (V, E)$ is computed as:

$$BC_i = \frac{\sum_{v_s, v_t \in V \land s < t} x_{st}^i}{g_{st} (|V| - 1)(|V| - 2)/2}$$  \hspace{1cm} (6.7)

where $x_{st}^i$ is the number of shortest paths between $v_s$ and $v_t$ that pass through node $v_i$. $g_{st}$ is the total number of shortest paths between $v_s$ and $v_t$. $|V| - 1)(|V| - 2)/2$ is the total number of pairs of nodes that exist in G, excluding $v_i$. The adapted, focused betweenness centrality $fBC$ we propose is computed as:
\[ fBC_i = \begin{cases} 
\frac{c_s c_t \sum_{C \in C^\theta} \frac{x_{st}}{n(n-1)/2}}{C_i \notin C\theta} \\
\frac{\sum_{C \in C^\theta} \frac{x_{st}}{(n-1)(n-2)/2}}{C_i \in C\theta}
\end{cases} \tag{6.8}
\]

The denominator changes depending if the target node is a seed node or not. Therefore, having \( n \) seed nodes, if the node \( C_i \) is one of them, then there are \( n(n-1)/2 \) pairs of nodes to be considered. Otherwise, we exclude the pairs of \( C_i \) so we only add up \( (n-1)(n-2)/2 \) pairs.

The effect of this focusing is exemplified in Table 6.1, and also illustrated from a graph visualisation perspective in Figure 6.3. In this example, the set of top-5 concepts scored by \( BC \) and the set of top-5 concepts scored by \( fBC \) have only one concept in common, Economic problems. By focusing, concepts that are more specific and related to the target topic occur among the top-5, such as Debt, Public economics and Monetary policy. Figure 6.3 better illustrates the score changes that occur due to focusing, between the \( BC \) graph in Figure 6.3(a) and the \( fBC \) graph in Figure 6.3(b). Nodes Public economics and Monetary policy belong to the top-5 in Figure 6.3(b), while their centralities are negligible in Figure 6.3(a). Also, Decision theory and Politics by issue which score high by \( BC \) (Figure 6.3(a)), have their weights dramatically reduced by \( fBC \) (Figure 6.3(b)).

### 6.4.4. Focused Random Walk Betweenness Centrality

As its name suggests, random-walk betweenness [106] is a variation of betweenness centrality. It roughly measures how often a node is traversed by a random walker going from any node \( s \) to another node \( t \), averaged over all pairs of nodes in the network.

It is computed by the following steps:

1. \( L = D - A \), where \( D \) is a diagonal matrix containing the degrees of the nodes and \( A \) is the adjacency matrix of \( G \). The matrix \( L \) is called the Laplacian matrix.

2. \( T_r = L_r^{-1} \), where \( L_r \) is called the reduced Laplacian. It is obtained from \( L \) by removing any single row \( r \) and the corresponding column. \( T_r \) is the reduced Laplacian’s inverse.

3. The matrix \( T \) is obtained from \( T_r \) by adding a row of zeros and a column of zeros on position \( r \).
(a) Graph Betweenness Centrality

(b) Focused Betweenness Centrality

(c) Focused Random Walk Betweenness Centrality

(d) Focused Information Centrality

Figure 6.3: Node centralities in the ego-network of topic with DBpedia concepts: [Government, Economy, Policy, Money, Unemployment, Income, Inflation, Interest, Interest rate, Demand (economics), Fiscal policy, Output]. The seed nodes are black, and the size of the nodes increases linearly with their centrality illustrated in each graph. For comparison, the nodes in the four graphs change only their sizes and not their positions.
4. Random walk betweenness of node \( v_i \), \( (RWB_i) \) is then computed as:

\[
RWB_i = \sum_{v_s, v_t \in V \land s < t} \frac{I_i^{(st)}}{(1/2)|V|(|V| - 1)}
\]  

(6.9)

where \( I_i^{(st)} \) is the so-called intensity, from this measure’s association to the current flowing through an electrical circuit [106].

\[
I_i^{(st)} = \frac{1}{2} \sum_{v_j \in V} A_{ij} |T_{ij} - T_{it} - T_{js} + T_{jt}|
\]  

(6.10)

The averaging factor \((1/2)|V|(|V| - 1)\) again is the number of all pairs of nodes in the graph.

For the focused random walk betweenness \( fRWB \), we limit the computation to all paths between all pairs of seed nodes, therefore we want to determine how often a node is traversed by a random walker that starts in any seed node, and ends in another seed node, averaged over all pairs of seed nodes. The computation steps are:

\[
fRWB_i = \begin{cases} 
\sum_{C_s, C_t \in \Theta \land s < t} \frac{I_i^{(st)}}{(1/2)n(n-1)} & \text{if } C_i \notin \Theta; \\
\sum_{C_s, C_t \in \Theta \land s < t} \frac{I_i^{(st)}}{(1/2)(n-1)(n-2)} & \text{if } C_i \in \Theta;
\end{cases}
\]  

(6.11)

Applying \( fRWB \) rather than \( fBC \), will also tend to favour high degree nodes, because a random walker is likely to pass through high degree nodes. This is noticeable in Figure 6.3 when analysing the Economics node scores of \( fBC \) and \( fRWB \).

The above measures \( fCC, fIC, fBC \) and \( fRWB \) are the ones that we experimented with for defining the target function \( r \), which quantifies the strength of the relation between each candidate concept and all other concepts in the topic graph \( G \). The graph-based labelling ranks all nodes of \( G \) by the chosen centrality measure and presents the top ones to the user as topic-label candidates.

These being the measures we use for extracting the topic labels from the topic graphs, we now move on to presenting how we integrate them with the word sense disambiguation approaches we introduced in Chapters 4 and 5. Afterwards, in Section 6.6, we present the
experiments we carried out to evaluate the effectiveness of these measures for labelling topics and documents.

6.5. Kanopy: Topic and Document Labelling on top of WSD

In this section, we propose a processing pipeline that uses our proposed labelling approach, on top of a WSD system, and that we call Kanopy. Used in combination with the WSD approaches that we previously proposed in this thesis, E-WSD and Sen-Dis, the resulting Kanopy implementation is illustrated in Figure 6.4. As seen in the figure, when used with Sen-Dis the labelling process reuses the ego-network extracted for disambiguation. When used with E-WSD an extra step is needed before topic labelling, for extracting the topic network.

![Figure 6.4: Kanopy: Topic Labelling on top of WSD with E-WSD or Sen-Dis](image)

The labels are extracted from the topic’s ego-network, therefore they might be related, and are not mutually exclusive. For instance a topic might have the top-1 label *Humanities*, the top-2 label *History* and the top-3 label *Ancient Rome* and so on.

6.5.1. From Topic Labelling to Document Labelling

The topic labelling methods based on the focused centrality measures we proposed are general enough to be theoretically applicable to any group of related words. To test this hypothesis, we rely on the assumption that most text documents are multi-themed and we try to capture these themes by word clustering, using the same techniques for word clustering as in Chapter 5. Then for each word cluster we obtain, we run the process shown in Figure 6.4. By processing text
documents through this pipeline of word clustering, disambiguation and labelling, we represent a document by a bag of words, a bag of concepts and a bag of labels, as illustrated in Figure 6.5. The labels are ranked based on their focused centrality scores, so that they can be limited to one or top-k per document topic.

![Figure 6.5: Document labelling with Kanopy](image)

In the following, we present the evaluation we carried out in order to assess the effectiveness of Kanopy, both in the context of topic models and documents.

## 6.6. Labelling Evaluation

Having introduced the proposed centrality measures for topic labelling, we now present the experiments we carried out in order to assess their performance. We performed the evaluation of these methods in two scenarios: topic labelling, evaluated in Section 6.6.1 and document labelling, evaluated in Section 6.6.2. In the first scenario we evaluated the methods based on a user study, where annotators manually assessed the labels proposed by the methods. For comparison, we also implemented a state-of-the-art text-based labelling method [88].

In the second scenario, we set ground truth labels for document clusters, and we judge the performance of the methods for labelling documents based on the frequency with which they label the documents with the corresponding cluster ground truth label. While this evaluation is more restrictive, it reveals the great potential as well as the limitations of our methods, and helps identifying the gaps that we plan to address in the future work.
In both evaluation scenarios, we extracted the DBpedia 2nd degree ego-networks from the Types & Categories graph (see page 108).

**6.6.1. Experiments and Evaluation of Topic Labelling**

In this section, we describe our experiments and the results we gained on the basis of a user study. Our main objective is to assess the suitability of the centrality measures we propose in Section 6.2.3 for labelling topics. The second objective is to understand the level of abstraction of the produced labels. To this end, we have extracted LDA topic models from three corpora with different text style: scientific essays, news articles and posts from question-answering forums. From these sources we selected some of the most coherent topics for evaluation. We used E-WSD presented in Chapter 4 for the disambiguation of the words in the selected topics. Then we extracted for each topic the topic graph and then applied the labelling methods. More specifically, we used the process illustrated in Figure 6.4, following the E-WSD disambiguation branch. For comparison, we also implemented a state-of-the-art text-based labelling approach from related work.

In order to evaluate the quality of the labels, we devised a user study where participants assessed the suitability of the system proposed labels. We now detail the experiment setup, the datasets, the results we obtained and the gained insights.

**Evaluated Methods**

<table>
<thead>
<tr>
<th>Pearson Correlations</th>
<th>$f_{CC}$</th>
<th>$f_{BC}$</th>
<th>$f_{IC}$</th>
<th>$f_{RWB}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>0.3365</td>
<td>0.4889</td>
<td>0.5072</td>
<td>0.6620</td>
</tr>
<tr>
<td>$f_{CC}$</td>
<td>1</td>
<td>0.4432</td>
<td>0.7967</td>
<td>0.5118</td>
</tr>
<tr>
<td>$f_{BC}$</td>
<td>1</td>
<td>0.4967</td>
<td>0.8923</td>
<td></td>
</tr>
<tr>
<td>$f_{IC}$</td>
<td>1</td>
<td>0.4967</td>
<td>1</td>
<td>0.6436</td>
</tr>
<tr>
<td>$f_{RWB}$</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

*Table 6.2.: Correlation of the focused centrality measures*

To keep the requirements of the user study in meaningful limits, we decided to ask the users to only evaluate $f_{IC}$ and $f_{RWB}$. First, each is strongly correlated with one of the measures not evaluated and they are less correlated with each other. Secondly, $f_{IC}$ and $f_{RWB}$ consider all paths of the topic graphs, and thus take more information about the network topology into account than their shortest path relatives. We show the corresponding Pearson correlation coefficients in Table 6.2.
An important aspect is to compare our methods that use only structured data from DBpedia to approaches that use only content data extracted from the documents to extract labels. We thus compare our algorithm to the state-of-the-art text-based approach \(TB\) as described in [88]. We implemented the first order relevance approach which was reported to have produced the best results. We described this approach in Section 3.2.1. The label score computation is also detailed in Formula (3.7) on page 51.

Therefore in the experiments that we present in this section, we evaluate in detail the following three labelling methods:

- \(fIC\) - focused information centrality;
- \(fRWB\) - focused random walk betweenness;
- \(TB\) - Mei et al.’s “first order relevance” text based method [88]

**Data**

For evaluating our approach and the different centrality measures, we require topics extracted and linked to DBpedia. To generate this, we ran LDA [86] on three corpora:

- The British Academic Written English Corpus (BAWE) [105] consists of 2,761 documents of proficient assessed student writing, ranging in length from about 500-5,000 words. The documents are fairly evenly distributed across four broad disciplinary areas (Arts and Humanities, Social Sciences, Life Sciences and Physical Sciences) covering 35 concrete disciplines.

- The BBC [45] corpus consists of 2,225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005: business, entertainment, politics, sport, and technology.

- The StackExchange \(^2\) dataset consists of all discussion threads from nine forums of the StackExchange website. We chose forums that matched the general knowledge of the users participating in the user study: wordpress, webmasters, web applications, photography, gaming, game development, android, cooking and bicycles. We merged all posts of a single thread in one document and the final dataset consists of 3,709 documents, roughly 400 documents per domain on average.

\(^2\)http://stackexchange.com/
We chose these three corpora because of the different text style they exhibit. We expect that the graph-based methods will be less sensitive to the text style than the text-based labelling method.

With respect to the user study, we aimed for evaluating 200 topics. Apart from the size, a user study also provides constraints by the actual user base and their background knowledge. First, topics should be understandable and coherent. To measure a topic’s coherence, we used the measure published in [96] and computed as:

$$\text{coherence}(\theta; w^{(\theta)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(w^\theta_m, w^\theta_l)}{D(w^\theta_l)} + 1$$ (6.12)

where $$w^\theta = \{w^\theta_1, \ldots, w^\theta_M\}$$ represents the set of top-$M$ most probable words of the topic $\theta$, $D(w)$ represents the number of documents containing the word $w$ at least once, and $D(w, w')$ represents the number of documents containing words $w$ and $w'$, at least once each.

We extracted 150 topics from BAWE, 50 topics from BBC and 50 topics from StackExchange ranging from medium to high coherence. We investigated the 250 topics and removed the ones that we still considered hard to grasp, for example [add, footer, class, title, register, instance, widgets, code, sidebar, id, area widget, php, li], some topics requiring domain knowledge, for example [chromatography, solvent, mobile, column, phase, mixture, mass, sample, gas, liquid, extraction, fraction, compound, stationary], as well as topics about humans that require familiarity with public persons, for example from sport: [morgan, davies, england, thomas, france, cardiff, welsh, nations, wales, jones].

For all the following experiments we therefore made sure not to extract the graph beyond the stop-URIs, and analysed only the core connected component of each topic graph. Figure 6.6 shows the histogram of the proportion of seed concepts that are connected in the core component. We note that more than a quarter of the topics have more than 90% of the seed concepts connected. The median is 0.77, meaning that for half of the topics more than 77% of seed concepts are core concepts. Also, more than 90% of the topics have more than half of their seed concepts belonging to the core topic graph. This strongly reinforces our conclusions from the previous chapter, that concepts of topics lie in proximity to one another in the DBpedia graph.
User Study

In order to comparatively evaluate the three methods, we created a web interface to gather input from human annotators. Before starting the task, the annotators were shown an example of a topic and suitable annotations for each label, so that they understand the requirements. The example is shown in Figure 6.7.

During the task execution, the annotators were given randomly chosen topics. For each topic, the annotators were given the top 5 labels produced by the three evaluated methods: $TB$, $fRWB$ and $fIC$. The labels were listed in a randomised order. The first letter of each label was capitalised so that this could not influence the users perception on the label. For each label, the annotators had to choose between: “Good Fit”, “Too Broad”, “Related but not a good label” and “Unrelated”. The user interface of the web application we developed for gathering users’ input is shown in Figure 6.8.

There was no restriction on how many “Good Fit” labels a topic could have, so users could choose none or several. In the final data set for evaluation, each label has been annotated by exactly three different annotators. There were 54 annotators in total, most of which being PhD students and post-doctoral researchers who voluntarily participated in our experiment.
We computed the Fleiss Kappa for the inter-annotator agreement in two cases: (i) on all four classes, and (ii) on two classes obtained by collapsing “Good Fit” and “Too Broad” as well as combining “Related but not a good label” and “Unrelated”. For the first case we obtained a value of 0.27, and 0.38 for the second case. These values are very much in line with the
agreement obtained by [87] for the task of topic indexing. As these values correspond to the level of fair to moderate agreement, this shows that, although topic labelling is a subjective task, a certain trend in users’ preferences can be observed.

### Performance Measures

![Graph showing performance measures](image)

**Figure 6.9:** Precision and Coverage (y axis) @top-k (x axis) for combined corpora.

We evaluated our methods in two types of tests. The first one, which we call *Good Fit*, counts a *Hit* for a method if the recommended label was annotated as “Good Fit” by at least 2 annotators. The second type of test, called *Good-Fit-or-Broader*, counts a Hit for a method if the recommended label was annotated as “Good Fit” or as “Too Broad” by at least two annotators. This second type is aiming at a scenario of (hierarchical) classification. We expect the relation between specialised terms and general vocabulary is hard to be captured using only text, but easier using structured data.

We compare the three chosen methods based on *Precision* and *Coverage*, taking the top-1 to top-5 suggested labels into account. Precision for a topic at top-k is computed as:

\[
Precision_{@k} = \frac{\# Hits with rank \leq k}{k} \quad (6.13)
\]
Then, we compute the average precision over all topics. As we cannot compute recall, due to the lack of ground truth, we define Coverage as the proportion of topics for which a method has found at least one Hit:

\[
Coverage@k = \frac{\#\text{topics with at least one Hit at rank } \leq k}{\#\text{topics}}
\]  

(6.14)

Results

Figure 6.9 shows the results for all topics from all three corpora. Figure 6.10 shows the results for each individual corpus.
The results indicate two advantages of our graph-based methods over the text-based one: a better coverage over all topics and a much higher ability to identify broader concepts. For the case of Good Fit, the precision values for all methods are comparable. An important difference can be seen for the precision@1 which is 31% for $f_{RWB}$ while the text-based method achieves 17%. Regarding coverage, $f_{RWB}$ has a Good Fit label among the top-5 in 61% of the cases, $f_{IC}$ in 57% and the $TB$ in 46%.

The graph-based methods achieve significantly better results than the text-based one, in the Good-Fit-or-Broader test. In 72% of the cases the top-1 label retrieved by $f_{RWB}$ was either a Good Fit or a Too Broad label while this was the case in 66% of topics labelled with $f_{IC}$ and 33.5% for $TB$. This shows that our approach is better suited for a classification scenario. This also confirms the intuition that the text-based labelling methods encounter problems generalising to broader terms. As for coverage on all corpora, $f_{RWB}$ achieves 96% in top-5, while $f_{IC}$ covers 94% and $TB$ 68%.

The analysis of the different corpora provides interesting insights also. Particularly the StackExchange fora corpus highlights differences. All three methods have their worst precision on the Good Fit test on this corpus, being almost constantly under 20%. As expected, this corpus poses problems especially for the text-based method, whose coverage@5 in the Good Fit test is 0.35, with $f_{RWB}$ scoring 0.6. On the same corpus, in the Good-Fit-or-Broader test, $TB$ has a coverage@5 of 0.45 whereas the $f_{RWB}$ scores 0.93 and $f_{IC}$ 0.90. Regarding the Good-Fit-or-Broader test on each corpus, the coverage@5 of $f_{RWB}$ and $f_{IC}$ reaches more than 0.9. More variation is seen in the coverage@5 of the $TB$ method, which is 0.78 on the BBC corpus, slightly lower on the BAWE corpus, while on StackExchange it results in its worst coverage@5 of less than 0.5.

These results show that the graph-based methods on DBpedia can achieve better results than the standard text-based methods. The text-based method is also more sensitive to the type of text. The graph-based methods are able to retrieve better labels without a high drop in quality for forum text. The biggest difference is observed in their bias towards broader labels as compared to the text-based method. More experiments are needed with other knowledge bases than only DBpedia in order to conclude if the bias towards broader labels is due to the nature of graph-based measures or due to the nature of concepts in DBpedia. However, the results indicate that the graph-based labelling is more suited for recommendation scenarios where a good coverage is more important than a good precision.

Regarding text-based labelling, our results show that the phrases that co-occur with the topic words are not necessarily suitable for labelling. Furthermore, in Section 6.6.2, we compute
in how many documents the ground-truth label is a concept that the document mentions, and show that in most documents, the suitable label does not occur. We will also show that at the same time the ideal label belongs to the topic graph in most cases. This further reinforces the need for topic labelling with external knowledge.

**Stability of Graph Measures**

Topic labelling using external knowledge strongly depends on the quality of the linking of topic words. In our experiments, the disambiguation algorithm received the top 15 words of each topic. Usually, there are topic terms that cannot be linked either because they do not have a corresponding DBpedia concept, or because they are not core concepts in the topic graph. Moreover, we also want to support cases when the input topics are not necessarily probabilistic latent topics, for instance if they are extracted from a sentence, and contain very few words. Therefore, we analyse the impact of the number of disambiguated concepts. We achieve this by inspecting the number of concepts in the core connected component of the topic graph. We want to check if topics with few core concepts are less likely to return accurate labels than concepts with many core concepts.

![Figure 6.11. Influence of number of seed nodes](image-url)
We selected the topics for which the graph-based methods did not find any label annotated with Good Fit by at least two annotators. Then, we statistically checked if the number of core concepts in these cases is biased in comparison to all topics. Figure 6.11 shows the distributions. For each method, we computed the Chi Square Goodness of Fit statistic with respect to the distribution of all topics. In both cases, there was no significant difference between the mistaken topics distribution and the distribution of all topics. For \( f_{RWB} \) we obtained \( \chi^2(13, n = 77) = 7.10, p > 0.10 \), and for \( f_{IC} \) we obtained \( \chi^2(13, n = 85) = 7.44, p > 0.10 \).

This result has an important practical significance, as it shows that even with less than 5 core concepts the labelling can be as successful as with more than 5 or even more than 10 core concepts.

**Convergence to top labels**

Another test we devised in order to understand the performance of the labelling measures, verifies how the set of the top-5 labels changes, as new seed concepts are considered. Therefore, we are interested in how the rankings of the labels change as words are added to the topic, one by one.

For this, we reran the experiments, seeding the graph extraction only with the top-5, top-6, up to top-11 seeds in the original order of the corresponding words in the topic. As seen in Figure 6.11 there are very few topics with 12, 13 and 14 core concepts, and none with 15 core concepts, so we did not analyse these cases, since no statistically significant conclusion can be drawn.

Figure 6.12 shows the proportions of top-5 labels at each stage that are also in the final set for topics with at most 11 core concepts. An important thing to notice here is that for all four measures, after 5 seeds, most of the topics methods converge to two or three of the final top-5. \( f_{CC} \) is the first one to converge, often even with only 6 seed concepts. This shows that \( f_{CC} \) is not very sensitive to new concepts. Once it identified concepts very close to its seed concepts, more seeds have no strong impact. \( f_{IC} \) also presents stronger convergence as after 8 seeds, all topics have at least 3 top labels settled. The most sensitive measures in this context are the betweenness-based measures, \( f_{BC} \) and \( f_{RWB} \) which span almost the whole interval even after 9 seed concepts. This indicates that betweenness methods might be the most vulnerable to noise.

After analysing the results of this evaluation with human judgements about the quality of labels, we conclude that graph centrality measures can be used with success to label topic
models, with the added benefit of integrating the topics in an external knowledge structure. In the following section, we analyse the same labelling methods, but from the perspective of labelling documents. We use names of clusters as the ground truth labels of the contained documents. As this evaluation strategy does not require human annotations, we can gain more insights into the performances of all measures, including $fCC$ and $fBC$.

### 6.6.2. Experiments and Evaluation of Document Labelling

In the previous section, we showed that the graph-based labelling methods can be used with success in the scenario of labelling topics. We now extend the scope of the evaluation, by also experimenting with document labelling. We identify three potential sources of labels:
1st Source: **Document words** the set of words and keyphrases of the document;

2nd Source: **Topic seed concepts** the set of concepts that the words and keyphrases of the document have been linked and disambiguated to (the topic graph seeds);

3rd Source: **Topic graph nodes** the set of DBpedia concepts that are related in the DBpedia graph to the seed concepts;

Our labelling methods extract the labels from the third source, by weighting all the graph nodes with respect to their focused centralities. We illustrate the three potential label sources in Figure 6.13.

**Figure 6.13:** Document labelling process, and the three potential sources of labels: (1) the document words (or topic words), (2) their DBpedia concepts counterpart obtained through linking and disambiguation (seeds), and (3) the DBpedia concepts from the seeds’ ego-networks;

In these experiments, we evaluate all four focused centrality measures: \( fCC \), \( fIC \), \( fBC \) and \( fRWB \). In order to interpret their performances, we also compute:
• the upper bound for the scores attainable by extracting the label from the seed concepts of the document; this bound is computed as the amount of documents whose seed concepts contain the ground truth label;

• the upper bound for the scores attainable by extracting the label from the set of all DBpedia concepts in the document’s topic graphs; this bound is computed as the amount of documents whose seeds’ ego-networks contain the ground truth labels.

In order to define ground-truth labels, we take advantage of the fact that the documents in BAWE corpus have metadata attached that contains, for each document the name of the discipline the document is part of. We use this data to group the BAWE documents in clusters based on the discipline. The StackExchange documents are naturally grouped based on the fora they are part of, and the BBC news-text documents are naturally grouped based on the five domains [45] they belong to. All these clusters and their labels are listed in Table 6.3.

We use the names of the document clusters from the three corpora as ground-truth labels. Our objective is to see if document processing as shown in Figure 6.13 produces, for each document, the label of the document cluster the document is part of. For this evaluation, we disambiguate the words of the document using Sen-Dis, described in detail in Chapter 5. The settings we use are: Louvain clustering for producing word clusters, and the Katz relatedness measure (see Section 5.5.2) with $\alpha = 0.5$ and considering the top-10 shortest paths.

Data

For this evaluation, we use the same document collections that we used in the previous labelling experiment. However, we filter out documents that contain more than 3000 words or 20kb. Longer documents are computationally too demanding for our system to process with the same settings we use for short documents. Additional settings like filtering of noun-phrases, and clustering that also considers text proximity of words, should be implemented to support long document labelling in a meaningful way. This filter left us with 2053 (out of 2761) BAWE documents, 3669 (out of 3709) StackExchange documents and the complete set of BBC documents (2225).

BAWE corpus contains metadata for each document, where the document’s discipline is stated. We use the name of the discipline, or the names of the closest DBpedia resource as the ground truth label for the document. Regarding StackExchange, we use the closest DBpedia resource to the title of the forum, and for BBC we use the names of the five domains. Table 6.3 shows the exact labels we look for.
In order to avoid the confusion between word or concept clusters and document clusters, in the remainder of this section we refer to word and concept clusters as *topics* and only use the term *cluster* to denote document clusters.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Original cluster name or Discipline</th>
<th>Number of Documents</th>
<th>DBpedia target labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAWE</td>
<td>Agriculture</td>
<td>120</td>
<td>Agriculture</td>
</tr>
<tr>
<td>BAWE</td>
<td>Anthropology</td>
<td>37</td>
<td>Anthropology</td>
</tr>
<tr>
<td>BAWE</td>
<td>Archaeology</td>
<td>43</td>
<td>Archaeology</td>
</tr>
<tr>
<td>BAWE</td>
<td>Architecture</td>
<td>8</td>
<td>Architecture</td>
</tr>
<tr>
<td>BAWE</td>
<td>Biological sciences</td>
<td>116</td>
<td>Biology</td>
</tr>
<tr>
<td>BAWE</td>
<td>Business</td>
<td>98</td>
<td>Business</td>
</tr>
<tr>
<td>BAWE</td>
<td>Chemistry</td>
<td>86</td>
<td>Chemistry</td>
</tr>
<tr>
<td>BAWE</td>
<td>Classics</td>
<td>70</td>
<td>Classics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Computer Science</td>
<td>68</td>
<td>Computer science</td>
</tr>
<tr>
<td>BAWE</td>
<td>Cybernetics &amp; Electronic Engineering</td>
<td>20</td>
<td>Cybernetics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Economics</td>
<td>71</td>
<td>Economics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Engineering</td>
<td>168</td>
<td>Engineering</td>
</tr>
<tr>
<td>BAWE</td>
<td>English</td>
<td>68</td>
<td>English</td>
</tr>
<tr>
<td>BAWE</td>
<td>Food Sciences</td>
<td>119</td>
<td>Food science</td>
</tr>
<tr>
<td>BAWE</td>
<td>Health</td>
<td>56</td>
<td>Health</td>
</tr>
<tr>
<td>BAWE</td>
<td>History</td>
<td>73</td>
<td>History</td>
</tr>
<tr>
<td>BAWE</td>
<td>Hospitality, Leisure &amp; Tourism Management</td>
<td>53</td>
<td>Hospitality industry</td>
</tr>
<tr>
<td>BAWE</td>
<td>Law</td>
<td>98</td>
<td>Law</td>
</tr>
<tr>
<td>BAWE</td>
<td>Linguistics</td>
<td>93</td>
<td>Linguistics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Mathematics</td>
<td>25</td>
<td>Mathematics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Medicine</td>
<td>56</td>
<td>Medicine</td>
</tr>
<tr>
<td>BAWE</td>
<td>Meteorology</td>
<td>28</td>
<td>Meteorology</td>
</tr>
<tr>
<td>BAWE</td>
<td>Philosophy</td>
<td>92</td>
<td>Philosophy</td>
</tr>
<tr>
<td>BAWE</td>
<td>Physics</td>
<td>42</td>
<td>Physics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Planning</td>
<td>13</td>
<td>Planning</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>----</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Urban planning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Urban studies and planning</td>
</tr>
<tr>
<td>BAWE</td>
<td>Politics</td>
<td>73</td>
<td>Politics</td>
</tr>
<tr>
<td>BAWE</td>
<td>Psychology</td>
<td>90</td>
<td>Psychology</td>
</tr>
<tr>
<td>BAWE</td>
<td>Publishing</td>
<td>22</td>
<td>Publishing</td>
</tr>
<tr>
<td>BAWE</td>
<td>Sociology</td>
<td>76</td>
<td>Sociology</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Android Enthusiasts</td>
<td>530</td>
<td>Android (operating system)</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Bicycles</td>
<td>353</td>
<td>Bicycle</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Bicycles</td>
<td></td>
<td>Bicycles</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Bicycles</td>
<td></td>
<td>Cycling</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Cooking</td>
<td>324</td>
<td>Cooking</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Game Development</td>
<td>304</td>
<td>Video Game Development</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Gaming</td>
<td>564</td>
<td>Gaming</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Photography</td>
<td>307</td>
<td>Photography</td>
</tr>
<tr>
<td>StackExchange</td>
<td>Web Applications</td>
<td>429</td>
<td>Web Application</td>
</tr>
<tr>
<td>StackExchange</td>
<td>WordPress</td>
<td>468</td>
<td>WordPress</td>
</tr>
<tr>
<td>BBC</td>
<td>Business</td>
<td>510</td>
<td>Business</td>
</tr>
<tr>
<td>BBC</td>
<td>Entertainment</td>
<td>386</td>
<td>Entertainment</td>
</tr>
<tr>
<td>BBC</td>
<td>Politics</td>
<td>417</td>
<td>Politics</td>
</tr>
<tr>
<td>BBC</td>
<td>Sport</td>
<td>511</td>
<td>Sport</td>
</tr>
<tr>
<td>BBC</td>
<td>Technology</td>
<td>401</td>
<td>Technology</td>
</tr>
</tbody>
</table>

Table 6.3.: Ground truth labels for document clusters

3http://android.stackexchange.com/
4http://bicycles.stackexchange.com/
5not online anymore
6http://gamedev.stackexchange.com/
7http://gaming.stackexchange.com/
8http://photo.stackexchange.com/
9http://webapps.stackexchange.com/
10http://wordpress.stackexchange.com/
Objectives

This evaluation follows several objectives:

- Our first objective is to understand if the labelling methods generalise over multiple domains and text styles (forum, news, scientific essay).
- The second objective is to understand the performance of the labelling methods with respect to the specificity of the ground truth label.
- The third objective is to understand the difference between selecting the labels from the document concepts (2nd source), and graph concepts (3rd source);

Performance Measures

Purity of Labelling with Focused Centralities

The measure we use in order to meet our objectives and assess the effectiveness of our labelling methods, is the cluster purity with respect to the ground truth label. It is computed as the ground-truth label frequency in the document cluster.

To compute it, we define the following variables:

- $D$ the set of documents in a cluster;
- $L^*$ the set of ground truth labels of a cluster, as defined in Table 6.3;
- $L^d_k$ the set of top-$k$ labels of document $d$; This is equal to the union of the top-$k$ labels of all topics of the document;

$$purity@k = \frac{|\{d \in D|L^* \cap L^d_k \neq \emptyset\}|}{|D|} \quad (6.15)$$

Upper Margin for the Purity of Labelling with Topic Seeds

For comparison, we also compute as shown in Formula (6.16), how frequent the ground-truth labels are among the concepts extracted from the document through disambiguation. These concepts are the seeds of the topic graphs. We then denote by $S^d$ the set of seed concepts of document $d$. 
Using the DBpedia Graph for Unsupervised Topic Labelling

\[
max_{\text{purity}}^{\text{Seeds}} = \frac{|\{d \in D| \mathcal{L}^* \cap S^d \neq \emptyset\}|}{|D|}
\]  
(6.16)

\[
max_{\text{purity}}^{\text{Seeds}} \text{ therefore acts as an upper margin for the maximum score achievable by approaches such as those of Medelyan et al. [87] and Grineva et al. [49], that extract the document label out of the senses of the disambiguated words of the document.}

**Upper Margin for the Purity of Labelling with Related Graph Concepts**

We also report throughout this evaluation, the frequency of the ground truth labels in the topic graphs of the documents. The nodes of the 2\textsuperscript{nd} degree topic graphs are the 2-step neighbours of the seed concepts. We denote by \( V^d \) the union of all vertices of all 2\textsuperscript{nd} degree topic graphs of document \( d \). We note that \( L_k^d \subseteq V^d \), for \( k \leq |V^d| \), and in general, \( S^d \subset V^d \). Then we compute the frequency of the ground truth labels in the topic graphs of the documents \( D \) as:

\[
max_{\text{purity}}^{\text{graph}} = \frac{|\{d \in D| \mathcal{L}^* \cap V^d \neq \emptyset\}|}{|D|}
\]  
(6.17)

Basically, this value computes the ratio of documents in which at least one ground truth label is extracted in the 2-hop topic graph of at least one topic. Therefore this value acts as an upper margin for the maximum score achievable by graph-based approaches that look for the document label in the DBpedia 2-hop neighbourhood graphs of the document concepts.

**The Gain Measure**

Our approach can retrieve the ground truth label whether or not the label belongs to the documents seed concepts. This can be checked by verifying if \( purity@k \geq max_{\text{purity}}^{\text{Seeds}} \) holds. The ground truth label can only be extracted from the topic graphs, therefore a maximum limit for \( purity@k \) is \( max_{\text{purity}}^{\text{graph}} \). We then define the target interval, target range and gain as follows:

\[
\text{target interval} = [max_{\text{purity}}^{\text{Seeds}}, max_{\text{purity}}^{\text{graph}}].
\]

\[
\text{target range} = max_{\text{purity}}^{\text{graph}} - max_{\text{purity}}^{\text{Seeds}}.
\]

\[
gain@k = purity@k - max_{\text{purity}}^{\text{Seeds}}.
\]

\[\text{11} \text{this relation does not hold for seed concepts that are not core concepts, and as such they are not considered in the labelling topic graph}\]
The target interval and target range are independent on the centrality measure used for labelling. They help us interpret the results from the perspective of maximum attainable by following our graph extraction method, and also the maximum attainable by extracting the label from the disambiguated senses of the document (i.e., seeds).

In general, a good centrality-based measure for labelling, will produce purity values in the target interval, which means that the gain is positive. The closer the purity is to the upper margin of the interval the better. The maximum attainable gain is the target range. A small target range suggests that a ground truth label occurs mostly in graphs where it is a seed concept. A large target range suggests that a ground truth label occurs often in the topic graphs of the documents even when it is not a seed concept. This case suggests that although the ground truth label is not mentioned in a document, is well connected in DBpedia to the concepts found in the document.

Using these notions, we now get to reporting and interpreting the results.

Results Overview

The labelling purity results for each whole corpus are illustrated in Figure 6.14. Each solid line corresponds to one of the four labelling measures, $f_{RWB}$, $f_{IC}$, $f_{CC}$ and $f_{BC}$. The shaded area illustrates the target interval.

![Figure 6.14: Labelling purity scores per corpus; The shaded area illustrates the target interval; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@$k$](image)
The first noticeable aspect is the difference between StackExchange and the other two corpora. The performance of all methods dramatically drops in the case of the StackExchange corpus, as seen in Figure 6.14(c). Also the target range is narrower and the $\text{max\_purity}_{\text{graph}}$ is lower for the StackExchange corpus. This shows that the ground truth labels occur more seldom in the topic graphs, than in the case of the BBC and BAWE. Specifically, only in 55% of the documents a ground truth label is extracted at all in the StackExchange topic graphs. For comparison, this proportion is 80% for BBC and above 90% for BAWE.

Another big difference is the ranking of the labelling methods: while $\text{fCC}$’s performance on BAWE and BBC is poorest out of the compared methods, it is the best in the case of StackExchange.

On the other hand, in the case of BAWE and BBC, the results are very promising. In the case of BAWE, the values of $\text{fIC}$, $\text{fRWB}$ and $\text{fBC}$ are already in the target interval when top-2 labels are considered. On the BBC corpus, the $\text{gain}@2$ for $\text{fIC}$ is 0.1 and $\text{gain}@5$ approaches 0.3. It is also worth noticing that in the case of the BBC corpus, the purity@1 of $\text{fIC}$, $\text{fRWB}$ and $\text{fBC}$ is almost as high as its frequency as a seed concept. In other words, if we used $\text{fIC}$ on BBC documents, by taking the first topmost labels only, the method would already obtain an overall score similar to that of a method that extracts the labels from the document seed concepts with an accuracy of 100%.

These first results already show the great potential for the semantic network based approaches to labelling, and also underline the main challenges, mostly in the forum texts. Nevertheless, these values are computed over all documents from the corpora, and as we show in the following, the results greatly vary for different clusters within the same corpus.

The StackExchange corpus

As shown, all the graph-based labelling measures perform worst on the StackExchange corpus. Figure 6.15 shows the result on each of the eight document clusters of our StackExchange corpus. We notice that there are very big differences in the target ranges of the clusters as well as the performances of the methods. All methods score best on the Photography (Figure 6.15 (f)) cluster, where even at top-1, the scores of all methods are in, or very close to the target interval. On this cluster, the $\text{gain}@5$ is between 0.2 and 0.4 for all the labelling methods. This performance is followed by the scores obtained on the Cooking and Bicycles clusters, that also have high $\text{max\_purity}_{\text{graph}}$ values. Nevertheless, the methods do not bring a positive gain for the Bicycles, and analysing the actual labels of this cluster, we noticed that
most frequent labels are in any case strongly related to the ground truth ones: *Bicycle parts* and *Cycle types*.

![Graphs showing the ground truth label frequency on the StackExchange document clusters](image)

**Figure 6.15.** The ground truth label frequency on the StackExchange document clusters; The shaded areas illustrate the target intervals; **x-axis:** top-k considered labels as extracted by the methods; **y-axis:** purity@k

In the case of the *Web Application* and *WordPress* clusters, we notice that there are almost no documents whose topic graphs contain the ground truth label, except for the documents in which the ground-truth label is a seed concept. This leads to the very narrow target intervals seen in Figures 6.15 (h) and 6.15 (g). These situations arise when the senses of the words in text, are not connected in a 2-hop topic graph to the ground truth labels. Therefore the ground truth labels are not extracted in the topic graphs, unless they are the seed nodes. This poor graph connectivity between the ground truth label and the seed concepts also prevents the ground truth label from obtaining high centrality scores. This leads subsequently to the poor performance of our methods. In the case of *Web Applications*, none of the methods extracted the *Web Application* concept as a label in top-5. In the case of *WordPress* however, *fCC* gives the ground truth label a high score, making this measure highly competitive to any method that would only score the document seed concept nodes.
The other clusters with specific labels are Android, Game Development and Gaming. In these cases, the max\_purity\_graph values are under 0.6 and under 0.4 in the case of Gaming. This also shows a comparatively low connectivity between the seed concepts and the ground truth label and might explain the poor centrality scores. However, for all these three clusters, the methods we propose produce scores inside the target interval, so they perform better than methods that extract the labels from the seed concepts.

To better illustrate the performance of the methods, we present in Tables 6.4 and 6.5, the top-10 labels for the WordPress and Gaming clusters. These values show that the methods we propose are able to identify suitable labels for the clusters, but broader than the ground-truth labels.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>fRWB</th>
<th>fIC</th>
<th>fCC</th>
<th>fBC</th>
<th>fRWB</th>
<th>fIC</th>
<th>fCC</th>
<th>fBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plug-in (computing)</td>
<td>Word Wide Web</td>
<td>World Wide Web</td>
<td>Data</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>Data management</td>
<td>Content management systems</td>
<td>Information</td>
<td>Data management</td>
<td>Technology</td>
<td>Technology</td>
<td>Technology</td>
<td></td>
</tr>
<tr>
<td>Blog</td>
<td>Evaluation</td>
<td>Interdisciplinary fields</td>
<td>Uniform resource locator</td>
<td>Software</td>
<td>Evaluation</td>
<td>Evaluation</td>
<td>Evaluation</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>Technology</td>
<td>Data management</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>Equals sign</td>
<td>Content management systems</td>
<td>Blogging</td>
<td>Error</td>
<td>Interdisciplinary fields</td>
<td>Error</td>
<td>Interdisciplinary fields</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>Code (metadata)</td>
<td>Society</td>
<td>Philosophy maintenance categories</td>
<td>Editing</td>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>Metadata</td>
<td>Society</td>
<td>Metadata</td>
<td>Society</td>
<td>Society</td>
<td>Society</td>
<td>Society</td>
<td></td>
</tr>
<tr>
<td>Editing</td>
<td>Interdisciplinary fields</td>
<td>Computer data</td>
<td>Time</td>
<td>Compute data</td>
<td>Computer data</td>
<td>Computer data</td>
<td>Computer data</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.4.**: Top-10 most frequent labels in WordPress cluster of StackExchange

<table>
<thead>
<tr>
<th>Concepts</th>
<th>fRWB</th>
<th>fIC</th>
<th>fCC</th>
<th>fBC</th>
<th>fRWB</th>
<th>fIC</th>
<th>fCC</th>
<th>fBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game</td>
<td>Games</td>
<td>Society</td>
<td>Time</td>
<td>Games</td>
<td>Games</td>
<td>Society</td>
<td>Games</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Video game gameplay</td>
<td>Philosophy maintenance categories</td>
<td>Video games</td>
<td>Video games</td>
<td>Video games</td>
<td>Video games</td>
<td>Video games</td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>Society</td>
<td>Concepts in metaphysics</td>
<td>Video game gameplay</td>
<td>Weapons</td>
<td>Weapons</td>
<td>Weapons</td>
<td>Weapons</td>
<td></td>
</tr>
<tr>
<td>Level (music)</td>
<td>Computing</td>
<td>Video game gameplay</td>
<td>Video</td>
<td>Strategy</td>
<td>Computing</td>
<td>Strategy</td>
<td>Computing</td>
<td></td>
</tr>
<tr>
<td>Xbox</td>
<td>Gaming</td>
<td>Interdisciplinary fields</td>
<td>Games</td>
<td>Weapons</td>
<td>Structure</td>
<td>Weapons</td>
<td>Structure</td>
<td></td>
</tr>
<tr>
<td>Enemy</td>
<td>Gaming</td>
<td>Interdisciplinary fields</td>
<td>Games</td>
<td>Quantity</td>
<td>Interdisciplinary fields</td>
<td>Quantity</td>
<td>Interdisciplinary fields</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>Business</td>
<td>Games</td>
<td>Weapons</td>
<td>Business</td>
<td>Business</td>
<td>Business</td>
<td>Business</td>
<td></td>
</tr>
<tr>
<td>Weapon</td>
<td>Concepts in metaphysics</td>
<td>Computing</td>
<td>Information</td>
<td>Interdisciplinary fields</td>
<td>Business</td>
<td>Business</td>
<td>Business</td>
<td></td>
</tr>
<tr>
<td>El_extra</td>
<td>Structure</td>
<td>Video games by platform</td>
<td>Information</td>
<td>Interdisciplinary fields</td>
<td>Business</td>
<td>Business</td>
<td>Business</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.5.**: Top-10 most frequent labels in Gaming cluster of StackExchange

We notice that the labels extracted by fIC, fRWB and fBC are broader terms than the ground-truth labels. fCC labels are more specific and correlated to the most frequent concepts in the cluster. This resonates with the example in Table 6.1 on page 142, where we already noticed that fCC tends to select labels that are part of the seed concepts. The examples in Tables 6.4 and 6.5 also show that the reported results are obtained under quite restrictive conditions, and that while the methods might not retrieve the exact ground-truth label, they retrieve quite close alternatives. For example, Games, Video games and Video game gameplay might be considered good alternatives to the ground truth Gaming.
The BBC Corpus

The ground truth labels of the BBC corpus are more general terms, therefore they favour our labelling methods. Figure 6.16 shows the ground truth label frequencies over the five BBC clusters.

![Graph](image_url)

**Figure 6.16:** The ground truth label frequency on the BBC document clusters; The shaded areas illustrate the target intervals; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k

The first thing to notice is that the values of $\text{max \_ purity}_\text{graph}$ are very high on the BBC clusters, with the exception of the Sport cluster. This suggests that the ground truth labels are well connected in the DBpedia graphs to the concepts used in the news texts of BBC. On the Business, Entertainment, Politics and Sport clusters, the frequency scores of $f_{RWB}$, $f_{IC}$ and $f_{BC}$ are in the target interval from top-1 or top-2 labels. $f_{CC}$ performs with 0.2, up to 0.4 worse than $f_{IC}$. The poor performance of $f_{CC}$ on this corpus is most likely due to the
broader nature of the labels, that as previously discussed, favour mostly the $fIC$, $fRWB$ and $fBC$ measures.

However on the Technology cluster, $fCC$ performs best, and in the target interval but very close still to $max\_purity^{Seeds}$. Table 6.6 lists the top-10 most frequent labels in the Technology cluster and shows that the nodes Society, Computing and Business have bigger focused centralities overall than Technology, except for $fCC$.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>IRWB</th>
<th>BC</th>
<th>ICC</th>
<th>BBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>Society</td>
<td>Society</td>
<td>Technology</td>
<td>Society</td>
</tr>
<tr>
<td>Technology</td>
<td>Computing</td>
<td>Computing</td>
<td>Information</td>
<td>Business</td>
</tr>
<tr>
<td>Company</td>
<td>Business</td>
<td>Main topic classifications</td>
<td>Computing</td>
<td>Technology</td>
</tr>
<tr>
<td>Computer</td>
<td>Technology</td>
<td>Business</td>
<td>Software</td>
<td>Technology</td>
</tr>
<tr>
<td>Business</td>
<td>Main topic classifications</td>
<td>Technology</td>
<td>Time</td>
<td>Humans</td>
</tr>
<tr>
<td>Software</td>
<td>Humans</td>
<td>Digital media</td>
<td>United Kingdom</td>
<td>Main topic classifications</td>
</tr>
<tr>
<td>BBC</td>
<td>Entertainment</td>
<td>Philosophy maintenance categories</td>
<td>Video</td>
<td>Entertainment</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Law</td>
<td>Film and video technology</td>
<td>Internet</td>
<td>Law</td>
</tr>
<tr>
<td>Research</td>
<td>Evaluation</td>
<td>Television terminology</td>
<td>Research</td>
<td>Communication</td>
</tr>
<tr>
<td>System</td>
<td>Communication</td>
<td>Law</td>
<td>Data</td>
<td>Evaluation</td>
</tr>
</tbody>
</table>

**Table 6.6.:** Top-10 most frequent labels in Technology cluster of BBC

Although in the target interval, all methods perform poorly on the Sport documents, with $fRWB$ only reaching 0.3 $purity^{@5}$. Nevertheless as seen in Table 6.7, Sport is among the top-3 most frequent labels of the cluster as extracted by $fRWB$, $fIC$ and $fBC$. The relatively high gain which is positive starting with top-2 labels for both $fRWB$ and $fIC$, as well as the rankings in Table 6.7, and the (comparatively) very low $max\_purity^{graph}$, indicate that the reason for low purity scores achieved by the centrality measures might have different causes among which: (i) the news texts that are related to sport, do not necessarily use sport terminology, for example they might often refer to sport management and (ii) there is no dominant DBpedia concept used for categorising sports concepts, and multiple equivalent categories fulfil the same role: Sports terminology, Sports, Terminology used in multiple sports, all of which being highly ranked labels as seen in Table 6.7.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>IRWB</th>
<th>BC</th>
<th>ICC</th>
<th>BBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game</td>
<td>Terminology used in multiple sports</td>
<td>Society</td>
<td>Terminology used in multiple sports</td>
<td>Society</td>
</tr>
<tr>
<td>Side</td>
<td>Sports</td>
<td>Sports</td>
<td>Time</td>
<td>Terminology used in multiple sports</td>
</tr>
<tr>
<td>Team</td>
<td>Society</td>
<td>Sports terminology</td>
<td>Games</td>
<td>Sports</td>
</tr>
<tr>
<td>Injury</td>
<td>Sports terminology</td>
<td>United Kingdom</td>
<td>Football</td>
<td>Games</td>
</tr>
<tr>
<td>Match</td>
<td>Management</td>
<td>Terminology used in multiple sports</td>
<td>BBC</td>
<td>Sports terminology</td>
</tr>
<tr>
<td>Time</td>
<td>Philosophy maintenance categories</td>
<td>Philosophy maintenance categories</td>
<td>Victory</td>
<td>Management</td>
</tr>
<tr>
<td>Club</td>
<td>Association football players by competition</td>
<td>Ball games</td>
<td>Sports terminology</td>
<td>Ball games</td>
</tr>
<tr>
<td>Champion</td>
<td>Ball games</td>
<td>Ball games</td>
<td>Ireland</td>
<td>Sports equipment</td>
</tr>
<tr>
<td>Sport</td>
<td>Games</td>
<td>Games</td>
<td>England</td>
<td>Events in athletics (track and field)</td>
</tr>
<tr>
<td>England</td>
<td>Human behaviour</td>
<td>British Isles</td>
<td>Premier League players</td>
<td>Law</td>
</tr>
</tbody>
</table>

**Table 6.7.:** Top-10 most frequent labels in Sport cluster of BBC
British Academic Written English (BAWE) Corpus

The last analysed corpus, BAWE is also the one with the most clusters. Figure 6.17 contains the BAWE clusters where the methods obtained positive gain, while Figure 6.18 shows the results on the clusters where the methods do not obtain positive gain.

Figure 6.17: Purity scores on BAWE clusters with positive gain; x-axis: top-k considered labels as extracted by the methods; y-axis: purity@k; the shaded areas illustrate the target interval.
Comparing the plots in Figure 6.17 to those in Figure 6.18, there are two important aspects to notice:

- the \( \text{max}_k \text{ purity}^{\text{Seeds}} \) values are much higher on clusters with negative gain (Figure 6.18).
- while \( fIC \) performs best on the clusters with positive gain, and \( fCC \) performs substantially worse, the ranking is opposite on the clusters with negative gain.

These two effects might indicate that the \( fIC \), \( fRWB \) and \( fBC \) tend to *not* select the labels from the seed concepts, while \( fCC \) has a strong bias towards the seed concepts. Therefore \( fCC \), as opposed to all the other measures, is very conservative: the frequencies with which it selects the ground truth labels are closest to the frequency with which the ground truth labels occur in text. This means that \( fCC \) does not perform as well when the ground
truth label is a central node by all other measures, but it performs much better when the ground
truth label is a seed label that is not central as computed by the other measures.

As seen in Figure 6.17, there are clusters on which the labelling methods perform particularly
well, like Chemistry, Law, Linguistics, Economics, Medicine, Biology. $fIC$ has a gain@5
of more than 0.7 for the Biology cluster. The same method achieves gain@5 values higher
than 0.5 on Chemistry, Linguistics, Economics, Meteorology and Sociology clusters. The
purity@5 score for the Law cluster is 1 for all methods except $fCC$. $fIC$ reaches more than
0.9 purity@5 score on Chemistry, Linguistics, Economics, Medicine, Sociology.

The analysis of this corpus, and especially the clusters where our centrality methods
do not obtain a positive gain (Figure 6.18), or obtain a positive gain, but still low purities
(Figure 6.17 (l) and Figure 6.17 (n)) might indicate a problem related to word clustering
into topics, rather than centrality computation. Analysing the labels, we notice that on these
document clusters, the most frequent labels capture only one dimension of the cluster domain.
For example, in the Food Sciences cluster, the most frequent labels are Foods, Food and drinks
and Chemistry. Although Food Sciences node is present in the graphs as seen by the high
value of max_purity_graph, it is not as central in general to the seed nodes as the mentioned
concepts. A possible explanation is that word clustering separates the food related concepts
from the chemistry concepts, into separate topics. This would definitely cause a poor centrality
for Food Sciences concept, which most likely lies on the paths connecting concepts from
the two domains. A similar case is the History cluster. Among the most frequent labels we
obtain: Society, Political philosophy, Law, Economics. Again, this indicates that the words
of these documents, are split into separate topics and labelled separately. The labelling of the
Classics cluster encounters the same problem, where the most frequent labels are on one side
Family, Gender, and on the other side Greek mythology and Ancient Rome. Nevertheless, the
solution to this problem is not straightforward, as to merge the topics and label the combined
set of concepts leads to a much expanded topic graph, which would inevitably be more noisy.
This would lead to broad concepts being very central. Therefore, the solution should focus
particularly on finding the right balance between the label focused centrality and the label
specificity, and we plan to look into it in detail in future work. This was also the case with the
Technology cluster of the BBC corpus, in Figure 6.16 (e) and Table 6.6. The most frequent
labels only capture one dimension each of the domain: Society, Computing or Business.

We notice another important aspect when analysing the Tourism, Publishing and Agriculture
clusters. In all cases, labels Business, Marketing and Management score much higher than
the ground truth labels. This is an example of generalisation, and as seen in Section 6.6.1, the
focused centrality measures tend to generalise to broader terms. There are two directions that
we consider for remedying this problem: the first is related to devising a measure of specificity / abstraction of the DBpedia concepts, and use it to reduce this bias towards more abstract labels. The second direction we consider to solve this problem is to score the labels using a scheme that besides the focused centrality of the label, also considers how frequently that label occurs for the other documents in a corpus, so that it prefers the more discriminative labels.

**Results Discussion**

Overall, we can therefore draw some conclusions about the performance of our proposed labelling methods:

- As we have high variation in the performances within the same corpus, we conclude that the methods are more influenced by the domain and by the required granularity than by the text style;

- They perform best on well defined domains that are not too specific;

- On specific topics, like the ones we experimented with in the StackExchange corpus, the methods select more general labels. Overall, our labelling methods achieve worst results on this corpus. This indicates that in the specific areas of the DBpedia network, the suitable labels might not occupy a central location with respect to the seed concepts, because the ground truth labels are quite specific themselves. We plan to address this in future work by trying to balance the label’s specificity and its focused centrality. A promising direction is to weight higher the candidate labels with high focused centrality (for instance, high focused betweenness) but low traditional centrality (for instance, betweenness). We strongly expect this to lead to the selection of more specific labels.

- On multi-disciplinary domains like History, Technology, Food Sciences, the methods fail to capture the correct labels, and only focus on one dimension at a time. This might be due to the preliminary word clustering, and deeper analysis is required to determine the root cause exactly, as well as a solution;

- \( f_{IC} \), \( f_{BC} \) and \( f_{RWB} \) give priority to labels that are not seed concepts, while \( f_{CC} \) gives priority to seed concepts. As a consequence, \( f_{IC} \), \( f_{BC} \) and \( f_{RWB} \) generalise to broader concepts much more than \( f_{CC} \), and perform much better, especially when the ground truth label is not a seed concept. At the same time, \( f_{CC} \) often performs best on domains that are problematic for the other measures, for being too specific;
• Approaches that look for the label in the topic graphs rather than limiting the search to the seed concepts have a much greater potential, their purity upper margin approaching 1 is many domains.

Besides the benefits of an improved labelling quality over the text-based labelling methods, our approach integrates the document concepts in the semantic network that contains the extracted labels. It thus opens a varied set of opportunities for the exploration of background knowledge about the topics. We have shown how this knowledge can be used for topic labelling, but the opportunities go much further. The relations between concepts can be analysed together with the graph structure around them, resulting in the discovery of rich knowledge that is not necessarily obvious from the text itself. Advantages of such an analysis are manifold. Besides concept linking, it can serve to: (i) provide explanations for concept linkage; (ii) enrich texts with a wealth of background knowledge; (iii) provide starting points for further knowledge exploration; (iv) provide insights in the quality/quantity of information the knowledge base contains about the topics discussed in the text revealing possible knowledge gaps.

Our system Kanopy Demo demonstrates these advantages. Kanopy Demo\textsuperscript{12} implements the Kanopy pipeline described in Section 6.5, and it is deployed as a web application. We present it in Appendix E.

6.7. Conclusion

In this work, we investigated approaches for graph-based topic labelling using DBpedia. We extract the DBpedia subgraph of topic concepts and adapt network centrality measures to identify concepts that are suitable for labelling the topic. On the basis of a user-study, we showed that the graph-based approaches perform constantly better than a state-of-the-art text-based method. The most important improvements are (i) better corpus coverage, and (ii) much higher ability to identify broader labels.

We also experimented with the proposed centrality measures for labelling documents from many various domains. We showed that they can be used on fora, news and essays, and that the text style impacts the results less than the actual domain specificity. However, as fora tend to be dedicated to narrow (specific) domains, this strongly affects the overall performance of the measures on this type of text. We have also shown that the extraction of labels from the topic graphs expands the chance of retrieving the correct label, when compared to the retrieval

\textsuperscript{12}http://kanopy.deri.ie
of labels from the document concepts. Our experiments also revealed that measures perform differently depending on the domain, and currently there is no one-size-fits-all solution. We expect that further analysis of the properties of the ground-truth labels, and their relations to the topic concepts will provide us with more insights and ultimately lead to algorithms that further improve the labelling accuracy.

Linking topics from a corpus to external knowledge bases like DBpedia has more benefits than just topic labelling. We illustrated in our demonstration web application some of these benefits, like understanding the actual relations between concepts, and between concepts and the chosen labels. Finally, a network of topics can be obtained for a corpus that can serve as basis for corpus navigation.
Part III.

Conclusion
Chapter 7.

Conclusion

This thesis investigated the correspondence between topic models and DBpedia and how it can be exploited for word-sense disambiguation and finally for topic labelling. The main intuition behind all our contributions is that structural properties of concepts can give valuable insights for solving the two problems.

Word-sense disambiguation is a fundamental challenge that needs to be dealt with by most systems that need to link unstructured text to structured data. Regarding our word-sense disambiguation work, we proposed two novel approaches, each being particularly suitable for certain scenarios:

- a novel gloss-based approach that provides very good results when multiple sense inventories are used simultaneously, as well as when there is little knowledge about senses apart from their glosses. The idea is to create an ad-hoc weighted bipartite graph connecting senses to their gloss-words. The way these two types of nodes are interconnected in this graph is our main cue for disambiguation.

- a novel semantic-network based approach that only requires that the senses are interconnected in a pre-given graph-structured semantic network;

Both these approaches have been thoroughly evaluated and shown to provide superior results to related work.

We treat the WSD problem as a global optimisation problem and simultaneously disambiguate all related target words. This type of approach has been neglected in related work, in favour of the one-word-at-a-time approach. However, we demonstrate that a global optimisation approach can be significantly more accurate than the local approach. The drawback lies in its computational complexity, which is exponential with respect to the number of target words. In
this regard, we proposed an approximate approach that keeps the disambiguation search space to a manageable size by splitting the disambiguation context. The smaller obtained problems are then suitable for exhaustive solution search. This approach is founded by our experiments that reveal that disambiguation accuracy with as few as 4 simultaneously disambiguated words is as good or even better than when more words are jointly considered.

The second problem this thesis tackled is topic labelling. Topic labelling supports humans interpret topics that would otherwise require a substantial cognitive overload. We treat it as a problem of ranking the concepts in the semantic network with respect to their importance relative to the topic. The intuition is that concepts that have an important structural role in the semantic network are most suitable for labelling topics. To this end, we introduced the notion of focused centralities as adaptations of traditional centrality measures used mostly in social network analysis. We thoroughly evaluated our proposed measures for labelling both topics and documents. We placed a particular focus on understanding the benefits and limitations of searching for topic labels in external knowledge, that is knowledge or content not provided directly by the content of the documents. We therefore showed that:

- the labels provided by our methods are more in line with users’ preferences than labels extracted through statistical measures over a collection of documents;
- the results of our approach are overall better than the maximum attainable results by the type of approaches that select the label from the concepts referred to in text;

In the following we treat each of our contributions in more detail, we discuss their limitations and how we can deal with them, and finally we reflect on future work directions.
7.1. Discussion

7.1.1. ASV and E-WSD

Motivated by the versatility of the gloss-based approaches to word-sense disambiguation, we proposed an algorithm that

- strongly improves the performance of state-of-the-art gloss-based WSD approaches;
- is unsupervised and straightforward to implement;
- works with any dictionary that provides a gloss, including with multiple dictionaries simultaneously;

The main challenge for such an approach is to define a measure for computing relatedness between glosses. The drawbacks of related work approaches can be overcome if the gloss-words have different weights, corresponding to their relevance for the disambiguation problem. We proposed such a gloss-based sense model, adaptive sense vectors (ASV), that adapts the weights of the gloss-words with respect to their relevance to all the senses in the joint disambiguation solution.

Furthermore, we are proposing an eigenvalue-based coherence measure, E-WSD, that captures the magnitude of the reinforcement between the senses (hubs) and their gloss-words (authorities). As such, it selects the disambiguation solution that has the highest connectivity between senses. Combining this measure and the adaptive sense vectors, the resulting system (ASV+E-WSD) achieves a disambiguation accuracy with 0.2 higher than the state-of-the-art gloss-based WSD approaches, as shown in our experiments.

Consequently, we use this approach in all core chapters of this thesis. In Chapter 5 we used it as a baseline for disambiguation with DBpedia. Its performance surpasses the reported results of state-of-the-art systems like DBpedia Spotlight. Afterwards, in Chapter 6 we used it in the disambiguation step that precedes our proposed topic labelling approach.

The main limitation of E-WSD is the computational complexity that characterises all global-disambiguation approaches. In order to understand how to address this, we evaluated its performance for different problem sizes, varying from 3 to 10 words requiring simultaneous disambiguation. We found that joint disambiguation does not require many words as context, since most methods achieved their best performance with 4 words. We therefore proposed in Chapter 5 an approximate search approach. It uses a wrapper function that selects the
words from the context to disambiguate iteratively. After each batch of disambiguated words, their selected senses are added to the disambiguation context of the following batch of target words. In our experiments, although not directly compared, this method did not appear to reduce the results ASV+E-WSD obtained with complete search. Our experience with global disambiguation suggests that the word-sense disambiguation domain provides sufficient heuristics for keeping the search space to a minimum without affecting the disambiguation accuracy. We therefore plan to investigate in more detail the efficiency and effectiveness of various approximation search methods, so that we can take full advantage of the high performance of global disambiguation.

7.1.2. Sen-Dis

In Chapter 5, we studied a word-sense disambiguation approach, Sen-Dis, that relies solely on DBpedia’s underlying graph structure, Sen-Dis. All previous work that disambiguates with DBpedia relies on the encyclopedic knowledge of Wikipedia associated with the DBpedia concepts. Our approach is the first to break the ties of DBpedia concepts with their Wikipedia articles, and focus only on the semantic relations between the DBpedia concepts. The premise for such an approach is that the graph structure captures relatedness between senses. We validated this by experimenting with several graph proximity measures, and comparing the graph proximity of nodes in the DBpedia graph to human assessment on their semantic relatedness. We showed that indeed, both path-based and neighbourhood proximity measures capture semantic relatedness. As for global disambiguation, Sen-Dis achieved very good results, surpassing results reported by related work in similar settings.

We also introduced the concept of stop-URIs, and our experiments show that their removal is crucial for the usefulness of the structure of the DBpedia graph. The existence of stop-URIs in the Categories graph of DBpedia has a negative impact on the meaningfulness of paths between DBpedia concepts. We quantified this impact and our results strongly suggest that any future work that uses the path lengths between DBpedia nodes, must find ways of dealing with these stop-URIs concepts. Therefore, we semi-automatically compiled a list of such stop-URIs that we remove from all the DBpedia subgraphs we extract. Our assumption is that these stop-URIs have such great impact because they exhibit some graph properties that the other nodes in the graph do not. As such we expect that analysis of graph patterns should detect them as outliers. For instance, they provide shortcuts to otherwise unrelated concepts, and bring together parts of the semantic network that would otherwise be more distant. This should be detectable by a lower clustering coefficient than most other nodes. We are planning to closely investigate
the properties of these nodes, so that we can detect them automatically. Nodes of this type may exist in many other semantic networks and therefore we find this direction of research necessary in light of generalising Sen-Dis to work on other graph-structured knowledge bases.

### 7.1.3. Topic Labelling with Focused Centralities

In Chapter 6, we tackled the problem of topic labelling using DBpedia. We investigated our assumption that the concepts that are suitable for labelling a topic are central in the DBpedia graph with respect to the topic concepts. We therefore proposed novel focused graph centrality measures. Our labelling algorithm that uses these centralities obtains much better results than text-based methods. We evaluated their performance for labelling both topics and documents.

However, the results we obtained in our user-study and our document cluster label evaluation show that in many cases the proposed focused centralities select more general labels. On the one hand this is good because they complement text-based approaches which cannot identify broader labels, often because these labels are not mentioned in texts. On the other hand, we need to understand and be able to control the required specificity of the labels. This is needed furthermore because topic labelling is itself a subjective task, and while experts might prefer specific labels, novices might prefer broader ones. In any case, we need to deepen our understanding of the graph properties of the ground-truth labels, their ego-networks and their relations to the topic concepts. This kind of analysis will open the possibility of adjusting label specificity to the needed level and the ability of “zooming-in and out” in the document topics.

### 7.1.4. Topics in Semantic Networks

Throughout this thesis, we used the concept of topic as the building block for text semantics. Our word-sense disambiguation methods are used for disambiguating words in topics. When disambiguating words from documents, the preliminary step we introduce is to group these words into topics, for instance by clustering them. We focus on labelling topics and then expand the scope of the labels to the documents of the topics. In Chapter 5, we showed that this preliminary grouping of words had great benefits for disambiguation of words in texts, improving disambiguation accuracy more than 0.15, compared to using all text words, even on one paragraph texts (less than 100 words in total).

As topics are the result of distributional semantics methods, our work can be viewed as an attempt to reconcile the statistically based topics produced by distributional semantics and
explicit semantics captured in semantic networks. Throughout this thesis we have quantified to what extent the topics are reflected in the structure of DBpedia. Our findings from Chapters 5 and 6 show a substantial correspondence between distributional semantics topics and the DBpedia graph. This opens up great opportunities for further integrating the knowledge extracted through statistical analysis of text and background knowledge. In this direction, we investigated the opportunity of topic labelling. We have also demonstrated some other benefits of this integration, such as the enrichment of documents with structured background knowledge in a demonstration web application. In the following section, we focus on future work directions that would not just deepen the research presented in this thesis, but also extend it to other applications.

7.2. Directions for Future Research

We now treat some of the research questions that this thesis does not attempt to answer, but that we plan to approach in our future work.

Do the graph-based approaches generalise to other knowledge bases? One of the first questions that needs answering is whether we can use our graph-based approaches on other knowledge bases than DBpedia. The use of general graph measures for disambiguation as well as for topic labelling is backed up by the motivation of finding measures that can be used (ideally) on any graph-structured knowledge base. The fundamental requirement for using Sen-Dis or centrality measures for topic labelling is that the topic model and the background knowledge base use the same criteria for associating concepts. Therefore, the first required check before attempting their use on a new knowledge base is the assessment of whether the connectivity of the topic concepts in the knowledge base is stronger than between random concepts. We showed that this is the case with DBpedia and the document collections we experimented with, and we plan to also investigate this direction, with other knowledge bases, both general domain (e.g., Freebase\(^1\)) and domain-specific (e.g., LinkedMDB\(^2\))

What does the topic graph tell us about the background knowledge? Throughout this thesis, we focused on judging what the background knowledge can tell us about the topics extracted from text. However, for reconciling distributional semantics to the semantics captured in external knowledge, we need to understand the capabilities and

\(^1\)http://freebase.com
\(^2\)http://linkedmdb.org
limitations of both sides. Distributional semantics captures which words are used together in text. Knowledge bases capture a more static view of the knowledge in a domain. A lack of synchronisation between these two might indicate that the knowledge base contains important knowledge gaps.

**Can topic graphs meaningfully guide the navigation of a collection of documents?**

In this thesis, we dealt with topics one by one independently. This work and mostly the extraction of the topic graphs, opens up a new way for finding connections between topics. By mapping all topics extracted from a document collection, we can find for example overlapping topics, the areas of the knowledge base that the collection deals with, and so on. Our plan is to use the structure of the knowledge base for document collection browsing/navigation.

Our vision is for a system that can link any document collection to any external graph-structured knowledge base, providing a means of summarising and navigating the main themes and the relations between them. The only requirement for such a system would be that the knowledge base covers to a certain degree the concepts of the document collection.
Appendix A.

Branch and Bound for Word Sense Disambiguation

A.1. Formalisation of the objective function

Let $H(V^H, E^H)$ be a subgraph of the relatedness graph $G^R(V, E)$. We define the vector $x$ of variables $x_{ij}$, each corresponding to a node in $V$, such that $x_{ij} = 1$ if and only if node $s_{ij} \in V$ also belongs to $V^H$, and $x_{ij} = 0$ otherwise. Therefore $x$ defines combinations of senses. Let us also define the $y$ vector of $y_{ijkp}$ variables, each corresponding to the $e_{ijkp} \in E$ edge between the nodes $s_{ij} \in V$ and $s_{pk} \in V$, and having the value 1 if the edge $e_{ijkp}$ also belongs to $E^H$ and 0 otherwise. Instantiations of the pairs $(x, y)$ define all possible subgraphs of $G^R$.

Then the objective function of the joint WSD problem can be expressed as $f(x, y) = \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} \sum_{k=i+1}^{m_k} \sum_{p=1}^{n} y_{ijkp} \text{relatedness}(s_{ij}, s_{kp})$, and it must be maximised under the constraints posed by our word-sense disambiguation problem. Then, we can formalise our optimisation
problem as:

$$ (x^*, y^*) = \arg \max_{(x,y)} \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{p=1}^{m_k} \text{relatedness}(s_{ij}, s_{kp}), \quad (A.1) $$

subject to

$$ y_{ijkp} = x_{ij}x_{kp}, \forall i \in \{1, \ldots, n - 1\}, j \in \{1, \ldots, m_i\}, \forall k \in \{i + 1, \ldots, n\}, p \in \{1, \ldots, m_k\} \quad (A.2) $$

$$ \sum_{j=1}^{m_i} x_{ij} = 1, \forall i \in \{1, \ldots, n\}, \quad (A.3) $$

and

$$ x_{ij} \in \{0, 1\}, y_{ijkp} \in \{0, 1\} \quad (A.4) $$

The imposed constraints define the set of feasible solutions, $FS$, or from the graph perspective, the set of maximal cliques of $G^R$. The first constraint, in Formula (A.3) ensures that an edge exists in the subgraph if and only if its corresponding nodes exist. Therefore it is not possible for two nodes to be part of the subgraph without having an edge between them, a condition posed by the clique property of the searched subgraph. The second constraint, Formula (A.4) ensures that in the subgraph there is exactly one sense from each word. Mathematically, these two constraints impose that we are looking for a subgraph of $G^R$ of exactly $n$ nodes, having $n \times (n - 1)/2$ edges, whose weighted sum is maximum.

### A.2. The Bounding Function

The bounding function that we suggest is best presented from an edge perspective. As shown earlier, a feasible solution of $n$ senses, corresponds to a maximal clique, therefore it contains $\frac{n(n-1)}{2}$ edges. Partial solutions correspond to cliques therefore a partial solution of $p$ senses contains $\frac{p(p-1)}{2}$ edges. Then the bounding function $g(PS)$ must approximate from above the maximum sum of edge weights that a clique of $\frac{n(n-1)}{2}$ edges can achieve, if it contains the $\frac{p(p-1)}{2}$ edges of the partial solution. The idea for this estimation is to relax the clique constraint, and look for the maximum sum of edge weights achievable for any graph of $\frac{n(n-1)}{2}$ edges included in $G^R$, that contains the $\frac{p(p-1)}{2}$ partial solution edges. This optimisation problem is much simpler than the original one, and can be solved greedily. $g(PS)$ is the function that achieves this greedy optimisation.
Let us denote by $W^{PS} \subseteq W$ the set of words that have a sense in the partial solution $PS$, and by $W^{-PS}$ the set of words whose senses are not set in $PS$, therefore $|W^{PS}| = p$ and $|W^{PS}| + |W^{-PS}| = n$. To obtain a graph of $\frac{n(n-1)}{2}$ edges, whose sum of edge weights is maximised, starting with the $\frac{p(p-1)}{2}$ edges of $PS$, we need to select $\frac{n(n-1)}{2} - \frac{p(p-1)}{2}$ more edges from $G^R$. We need this graph to be as close as possible to a feasible solution, so that it provides a close bound, so we impose the following greedy selection of the remaining edges:

- for each sense $s$ in $PS$, select the top highest weighted $n - p$ edges that are adjacent to $s$ and to any sense of any word in $W^{-PS}$;
- select the top highest weighted $\frac{(n-p)(n-p-1)}{2}$ edges that are adjacent only to senses of words in $W^{-PS}$;

This selection fulfils the constraint of adding up $\frac{n(n-1)}{2}$ edge weights as

$$\frac{p(p-1)}{2} + p(n-p) + \frac{(n-p)(n-p-1)}{2} = \frac{n(n-1)}{2};$$

Therefore the bounding function $g$ we define is:

$$g(PS) = \sum_{s_i \in PS} \sum_{s_j \in PS, j \neq i} relatedness(s_i, s_j)$$

$$+ \sum_{s_i \in PS} \sum_{k=1}^{n-p} top(s_i, W^{-PS})[k]$$

$$+ \frac{1}{2} (n-p)(n-p-1)$$

$$+ \sum_{k=1}^{top(W^{-PS})[k]};$$

(A.7)

where $top(s_i, W^{-PS})$ represents the descending sorted list of weights of edges between $s_i$ and senses of words from $W^{-PS}$, $top(W^{-PS})$ denotes the descending sorted list of weights of edges between senses of words from $W^{-PS}$.

### A.3. Example

Let us consider four words: $W = \{A, B, C, D\}$, with the following possible senses: $M_A = \{s_{a,1}, s_{a,2}, s_{a,3}\}$, $M_B = \{s_{b,1}, s_{b,2}\}$, $M_C = \{s_{c,1}, s_{c,2}, s_{c,3}\}$ and $M_D = \{s_{d,1}, s_{d,2}\}$ Then for WSD, the feasible solutions set $FS$ contains 36 feasible solutions and 27 partial solutions, $FS = \{\{s_{a,1}, s_{b,1}, s_{c,1}, s_{d,1}\}, \{s_{a,1}, s_{b,1}, s_{c,2}, s_{d,1}\}, \{s_{a,1}, s_{b,1}, s_{c,3}, s_{d,2}\}, \ldots\}$. Some examples of
partial solutions are: \( \{s_{a,1}\}, \{s_{b,2}\}, \{s_{a,1}, s_{c,1}, s_{d,1}\} \). The search space has a tree-like shape, and is illustrated in Figure A.1. The relatedness graph \( G^R \) is presented in Figure A.2.

Let us now exemplify of how the algorithm works when the pairwise relatedness score are set as in Table A.1. For fast computation of the bounding function, we prepare a sorted list of neighbours for each sense as in Table A.2, and a sorted list of all relatedness scores.

The order in which the words are being considered in the building of the partial solutions is ascending by number of possible senses: B,D,A,C. Table A.3 shows all computation needed to find the optimal feasible solution for this example. The total number of possible sense combinations is 36, and the number of partial solutions is 27, but the system computes the objective function for only three feasible solutions and the bounding function for seven partial solutions until reaching the optimal solution.
Sorted neighbour senses

<table>
<thead>
<tr>
<th>Sorted neighbour senses</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(s1, s2)$</td>
</tr>
<tr>
<td>$(s3, s2)$</td>
</tr>
</tbody>
</table>

Table A.1.: Example values for pairwise relatedness between the senses of four words.

Table A.2.: The sorted lists of pairwise relatedness

After the branching of each node, new partial solutions are stored in the *Live* priority queue. When possible solutions are found, the *Live* priority queue is updated by removing all the partial solutions whose bound value is lower than the score of the newly found solution. Therefore in Step 4, after the first possible solution is computed, one partial solution is removed from *Live*, and after computing the third possible solution, all the entries in *Live* are removed, leading to the end of the process, and to the certainty that the best possible solution has been found.
Step 1
Branch $B$:

$\text{bound}(b_1) = \text{top}(3, b_1) + \text{top}(3, \text{All} \setminus B) = 0.6 + 0.55 + 0.5 + 0.75 + 0.55 + 0.55 = 3.5$
$\text{bound}(b_2) = \text{top}(3, b_2) + \text{top}(3, \text{All} \setminus B) = 0.7 + 0.65 + 0.65 + 0.75 + 0.55 + 0.55 = 3.85$

$\text{Live} = \{(b_2, 3.85), (b_1, 3.5)\}$

Step 2
$\text{current} = \text{Live.pop()} = (b_2, 3.85)$;

Branch $b_2$:

$\text{bound}(b_2d_1) = \text{rel}(b_2, d_1) + \text{rel}(b_2, d_1) + \text{top}(1, b_2 \setminus (A \cup D)) + \text{top}(1, d_1 \setminus (A \cup B)) + \text{top}(1, a_1 \setminus (B \cup D)) = 0.55 + 0.6 + 0.5 + 0.7 + 0.5 + 0.55 = 3.4$
$\text{bound}(b_2d_2) = 3.7$
$\text{bound}(b_2d_3) = 3.15$

$\text{Live} = \{(b_2d_1a_2, 3.7)(b_2d_2, 3.55)(b_1, 3.5)\}
{(b_2d_1a_3, 3.4)(b_2d_2a_3, 3.15)\}$

Step 3
$\text{current} = \text{Live.pop()} = (b_2d_1, 3.7)$;

Branch $b_2d_1$:

$\text{bound}(b_2d_1a_1) = \text{rel}(b_2, d_1) + \text{rel}(b_2, a_1) + \text{rel}(d_1, a_1) + \text{top}(1, b_2 \setminus (A \cup D)) + \text{top}(1, d_1 \setminus (A \cup B)) + \text{top}(1, a_1 \setminus (B \cup D)) = 0.55 + 0.6 + 0.5 + 0.7 + 0.5 + 0.55 = 3.4$
$\text{bound}(b_2d_1a_2) = 3.7$
$\text{bound}(b_2d_1a_3) = 3.15$

$\text{Live} = \{(b_2d_1a_2, 3.7)(b_2d_2, 3.55)(b_1, 3.5)\}
{(b_2d_1a_3, 3.4)(b_2d_2a_3, 3.15)\}$

Step 4
$\text{current} = \text{Live.pop()} = (b_2d_1a_2, 3.7)$;

Branch $b_2d_1a_2$:

$\text{bound}(b_2d_1a_2c_1) = f(b_2d_1a_2c_1) = \text{rel}(b_2, d_1) + \text{rel}(b_2, a_2) + \text{rel}(b_2, c_1) + \text{rel}(d_1, a_2) + \text{rel}(d_1, c_1) + \text{rel}(a_2, c_1) = 0.55 + 0.65 + 0.65 + 0.55 + 0.3 + 0.5 = 3.2$

$\text{Live} = \{(b_2d_2, 3.55)(b_1, 3.5)(b_2d_1a_1, 3.4)\} f(b_2d_1a_2c_2) = 3.2$

$\text{Live} = \{(b_2d_2, 3.55)(b_1, 3.5)(b_2d_1a_1, 3.4)\} f(b_2d_1a_2c_2) = 3.6$

OptimumSolution = $a_2b_2c_1d_1$

Table A.3.: The B&B WSD algorithm applied to example data
Appendix B.

DBpedia Concept Index

For fast retrieval of candidate DBpedia resources, we indexed all the DBpedia concepts extracted from the DBpedia datasets, and indexed them in a Lucene Index by their name and the names of the pages that redirect to them. We also save for every DBpedia resource the ambiguous pages it disambiguates, and the definition extracted from the rdfs:comment property, as shown in Figure B.1. For the ambiguous concepts, which we recognise as they have the property dbpedia-owl:wikiPageDisambiguates, we index the name of the concept, and we save the possible disambiguations, as in Figure B.2. We also store and index the standardised text version of all aforementioned fields. For standardisation we lemmatise the words and remove the stopwords from phrases. The Alternative names field contains the names of the DBpedia resources that have a dbpedia-owl:wikiPageRedirects property that points to the resource in question.

| Name(INDEXED): International; Alternative names(INDEXED): Intl., Internationally, Intl, Int’l, Inernational; Ambiguous Page(STORED): International_(disambiguation); Definition(STORED): International mostly means something involving more than one country. The term international as a word means involvement of, interaction between or encompassing more than one nation, or generally beyond national boundaries. For example, international law, which is applied by more than one country and usually everywhere on Earth, and international language which is a language spoken by residents of more than one country; Is Ambiguous Page(STORED): False; |

Figure B.1.: Index Entry for DBpedia resource: http://dbpedia.org/resource/International

For every noun phrase that we extract from text, we extract the candidate concepts by querying the previously described DBpedia resource index. Four queries are run on the index:
**Name**(INDEXED): International_(disambiguation);
**Is Ambiguous Page**(STORED): True;

Figure B.2: Index Entry for DBpedia resource: [http://dbpedia.org/resource/International_ (disambiguation)]

query for the exact noun-phrase as **Name**, query for the exact noun-phrase as **Alternative name**, query for the standardised noun-phrase as **Standardised name** and query for the standardised noun-phrase as **Standardised alternative name**. The last two queries are run only if the first two queries do not return any match.
Appendix C.

DBpedia Candidate Sense pre-filter

In the DBpedia semantic network, the traversal of multiple consecutive `dbpedia-owl:wikiPageDisambiguates` and `dbpedia-owl:wikiPageRedirects` can generate anomalous candidate senses. For example, consider the example in Figure C.1(a). If the term “obama” is queried, the retrieved concept is `dbres:Obama`, which is redirected to `dbres:Barack_Obama`. This can be interpreted that the DBpedia’s default concept for the word “Obama” is Barack Obama. However, DBpedia also contains the `dbres:Obama_(disambiguation)` concept, that points to more disambiguation options for the word “obama”, for example the Mount Obama. Therefore, when redirected to a concept, the concept retrieval process must also check if that concept is actually a default out of many more disambiguation choices, and retrieve all the disambiguation choices.

However, this strategy entails the risk of retrieving many disambiguation concepts that are not suitable for the queried phrase. For example, consider the case of the term “panthera onca”, illustrated in Figure C.1(b). Following the same reasoning as for the previous example, in this case, the concept `dbres:Jaguar_Cars` is retrieved as a disambiguation choice for the term “panthera onca”. While “panthera onca” is not an ambiguous word, “jaguar” is and this situation occurs very often in DBpedia in case of synonyms. These disambiguation candidates can bring a lot of noise in the disambiguation process. In situations of this type, we only retrieve an option of the disambiguation page if its label is similar to the queried term. We measure this similarity by the Monge Elkan \cite{monge_elkan} string similarity. We consider two strings similar if their Monge Elkan similarity score is higher than 0.8, so that it captures misspellings and acronyms. Also, note that for string inclusion, for example “Obama” and “Barack Obama”, the Monge Elkan similarity is 1.

Another aspect that we consider when retrieving possible sense candidates, is the named entity type of the term in text. If the term is a named entity in text, as annotated by Stanford...
CoreNLP package\(^1\), then we only retrieve the candidate senses that are named entities. We check if a DBpedia concept is a named entity based on a predefined list of DBpedia and YAGO classes that only contain named entities, and whose entries are enumerated in Appendix D. Given a concept, we consider it a named entity if this list contains any of its class ancestors.

In case the term is a common noun in text, then we retrieve for disambiguation all candidates, including named entities, because the false negatives are relatively frequent in named entity annotation, especially when capitalisation is not respected. However, we only retrieve the candidates if the set of all candidates contains at least one sense that is not a named entity. We enforce this condition in order to avoid penalising the WSD system when DBpedia does not contain the right sense of a word.

\(^1\)[http://nlp.stanford.edu/software/corenlp.shtml]
Appendix D.

List of DBpedia and Yago classes containing named entities

D.1. DBpedia

http://schema.org/Person
http://xmlns.com/foaf/0.1/Person
http://dbpedia.org/ontology/Artist
http://dbpedia.org/ontology/FictionalCharacter
http://dbpedia.org/ontology/Writer
http://dbpedia.org/ontology/Person
http://dbpedia.org/ontology/Scientist
http://dbpedia.org/ontology/Politician
http://dbpedia.org/ontology/Island
http://dbpedia.org/ontology/Village
http://dbpedia.org/ontology/Park
http://dbpedia.org/ontology/Hotel
http://dbpedia.org/ontology/MountainRange
http://dbpedia.org/ontology/Place
http://dbpedia.org/ontology/PopulatedPlace
http://dbpedia.org/ontology/AdministrativeRegion
http://dbpedia.org/ontology/City
http://dbpedia.org/ontology/Town
http://dbpedia.org/ontology/Airline
http://dbpedia.org/ontology/College
List of DBpedia and Yago classes containing named entities

http://dbpedia.org/ontology/SportsTeam
http://dbpedia.org/ontology/Company
http://dbpedia.org/ontology/Broadcaster
http://dbpedia.org/ontology/TelevisionStation
http://dbpedia.org/ontology/PoliticalParty
http://dbpedia.org/ontology/Organization
http://dbpedia.org/ontology/Band
http://dbpedia.org/ontology/Agent
http://dbpedia.org/ontology/Film
http://dbpedia.org/ontology/Aircraft
http://dbpedia.org/ontology/Island
http://dbpedia.org/ontology/Airport
http://dbpedia.org/ontology/Bridge
http://dbpedia.org/ontology/Volcano
http://dbpedia.org/ontology/College
http://dbpedia.org/ontology/SportsTeam
http://dbpedia.org/ontology/Canal
http://dbpedia.org/ontology/Software
http://dbpedia.org/ontology/Actor
http://dbpedia.org/ontology/Broadcaster
http://dbpedia.org/ontology/TelevisionStation
http://dbpedia.org/ontology/Scientist
http://dbpedia.org/ontology/Politician
http://dbpedia.org/ontology/PoliticalParty
http://dbpedia.org/ontology/MountainRange
http://dbpedia.org/ontology/Automobile
http://dbpedia.org/ontology/MusicFestival
http://dbpedia.org/ontology/GolfLeague
http://dbpedia.org/ontology/Writer
http://dbpedia.org/ontology/Artist
http://dbpedia.org/ontology/Place
http://dbpedia.org/ontology/PopulatedPlace
http://dbpedia.org/ontology/AdministrativeRegion
http://dbpedia.org/ontology/Ship
http://dbpedia.org/ontology/City
http://dbpedia.org/ontology/FictionalCharacter
http://dbpedia.org/ontology/Saint
http://dbpedia.org/ontology/Town
http://dbpedia.org/ontology/River
http://dbpedia.org/ontology/Lake
http://dbpedia.org/ontology/Play
http://dbpedia.org/ontology/Organization
http://dbpedia.org/ontology/Band
http://dbpedia.org/ontology/Agent
http://dbpedia.org/ontology/MusicalWork
http://dbpedia.org/ontology/Single
http://dbpedia.org/ontology/Song
http://dbpedia.org/ontology/Album
http://dbpedia.org/ontology/Work
http://dbpedia.org/ontology/TelevisionShow

D.2. YAGO

http://dbpedia.org/class/yago/MusicalComposition107037465
http://dbpedia.org/class/yago/Location100027167
http://dbpedia.org/class/yago/Company108058098
http://dbpedia.org/class/yago/Organization108008335
http://dbpedia.org/class/yago/Program106748466
http://dbpedia.org/class/yago/Newspaper106267145
http://dbpedia.org/class/yago/PrintMedia106263609
http://dbpedia.org/class/yago/2000Singles
http://dbpedia.org/class/yago/2001Singles
http://dbpedia.org/class/yago/2002Singles
http://dbpedia.org/class/yago/2003Singles
http://dbpedia.org/class/yago/2004Singles
http://dbpedia.org/class/yago/2005Singles
http://dbpedia.org/class/yago/2006Singles
http://dbpedia.org/class/yago/2007Singles
http://dbpedia.org/class/yago/2008Singles
http://dbpedia.org/class/yago/2009Singles
http://dbpedia.org/class/yago/2010Singles
http://dbpedia.org/class/yago/2011Singles
http://dbpedia.org/class/yago/Album106591815
http://dbpedia.org/class/yago/GirlGroups
http://dbpedia.org/class/yago/1960sMusicGroups
http://dbpedia.org/class/yago/1990sMusicGroups
http://dbpedia.org/class/yago/CanadianFolkMusicGroups
http://dbpedia.org/class/yago/All-femaleBands
http://dbpedia.org/class/yago/FictionalCharacter109587565
http://dbpedia.org/class/yago/MusicalOrganization108246613
http://dbpedia.org/class/yago/DramaticComposition107007684
http://dbpedia.org/class/yago/Writing106362953
http://dbpedia.org/class/yago/Chip103020034
http://dbpedia.org/class/yago/YagoLegalActor
http://dbpedia.org/class/yago/YagoGeoEntity
http://dbpedia.org/class/yago/Game100456199
http://dbpedia.org/class/yago/EndProduCt103287178
http://dbpedia.org/class/yago/Creation103129123
http://dbpedia.org/class/yago/TarotCard104394821
http://dbpedia.org/class/yago/PlayingCard103963982
Appendix E.

Demonstration

Kanopy Demo is a web application that implements all the algorithms presented in this thesis within the Kanopy pipeline (see Section 6.5). Its user interface is shown in Figure E.1.

Figure E.1.: UI of Kanopy Demo web application demonstrating document labelling

On the left-hand side, the text area contains the text that has been processed. In this example we used the body of a news-text related to research about the Indian tigers endangered by extinction due to poor genetic diversity\(^1\). The results are shown on the right-hand side of the user interface, on two columns: Extracted Concepts and Categories. The first one contains the concepts extracted from the text, clustered and linked and / or disambiguated.

\(^1\)http://www.bbc.com/news/uk-wales-22536571
The second column contains the extracted labels. For experimentation, the application allows manual setting of the clustering method as well as the granularity of the clusters.

Some topics that Kanopy Demo identifies on the example text with the default settings are about locality (India), scientific research, and genetics. For all document topics, the system also allows the exploitation of the topic graph. The third topic is of particular interest. As the column Extracted Concepts shows, it brings together different concepts found in the text, such as “Preservation breeding”, “DNA”, “Gene pool”, “Extinction” and “Genetic structure”. Opening the topic graph, the user interface displays a compressed version of the topic graph as shown in Figure E.2.

![Figure E.2](image)

**Figure E.2:** UI of Kanopy Demo web application demonstrating the possible exploration of the topic graphs and relation between concepts and labels

This compressed graph is formed by displaying only the shortest-path between all pairs of concepts and labels. Although the computation is all done on undirected graphs, ignoring the actual semantics of the relations, the user interface displays the actual direction of the original DBpedia properties between nodes. The seed concepts are evidenced by the light yellow background, while the labels are evidenced by the green background whose intensity increases as the score of the label increases. Therefore in the example in Figure E.2, “Genetics” has the highest score. The concepts with the grey background are the nodes that enable the relations between the concepts and labels, for example, ”Ecology” is the node responsible for the 2-edge path between the seed concept ”Gene pool” and the label ”Biology”.

For experimentation, the user interface allows setting the used centrality measure out of the ones we presented in this chapter, as well as the number of top labels to display. Therefore, the showcased topic graph highlights that "Genetics" is directly connected to the text mentions of "Genetic structure" and "DNA", a fact that explains its high score as a label. The graph also clarifies that "Population Genetics", the second-ranked label, is part of "Genetics" – which in turn is part of "Biology". None of these recommended labels occurred in the original text. Besides these multi-concept topics, the user interface also displays single concepts that remained isolated, and were not used or evaluated for labelling, such as "Tiger", together with the DBpedia categories and classes they belong to.
Bibliography


[95] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM ’08,


[105] Hilary Nesi, Sheena Gardner, Paul Thompson, and Paul Wickens. British academic written english (bawe) corpus. Universities of Warwick, Reading and Oxford Brookes,


[129] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4 (11):992–1003, 2011.


Linguistics.