Dissertationen aus der Naturwissenschaftlich-
Technischen Fakultät I der Universität des Saarlandes

Graph-based Methods for Large-Scale Multilingual Knowledge Integration

Gerard de Melo

universaar
Universitätsverlag des Saarlandes
Saarland University Press
Presses Universitaires de la Sarre

Gerard de Melo

Graph-based Methods for Large-Scale Multilingual Knowledge Integration

universaar
Universitätsverlag des Saarlandes
Saarland University Press
Presses Universitaires de la Sarre

D 291

Universitätsverlag des Saarlandes
Saarland University Press
Presses Universitaires de la Sarre

Postfach 151150, 66041 Saarbrücken

ISBN 978-3-86223-028-0 gedruckte Ausgabe
ISBN 978-3-86223-029-7 Online-Ausgabe
URN urn:nbn:de:bsz:291-universaar-278

Doctoral Dissertation, Saarland University
Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften (Dr.-Ing.) der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

Defense: Saarbrücken, 15 December 2010
Committee: Gert Smolka, Gerhard Weikum, Hans Uszkoreit, Hinrich Schütze, Sebastian Michel
Dean: Holger Hermanns

Projektbetreuung *universaar*: Isolde Teufel

Satz: Gerard de Melo
Umschlaggestaltung: Julian Wichert

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Abstract

Given that much of our knowledge is expressed in textual form, information systems are increasingly dependent on knowledge about words and the entities they represent. This thesis investigates novel methods for automatically building large repositories of knowledge that capture semantic relationships between words, names, and entities, in many different languages. Three major contributions are made, each involving graph algorithms and statistical techniques that combine evidence from multiple sources of information.

The lexical integration method involves learning models that disambiguate word meanings based on contextual information in a graph, thereby providing a means to connect words to the entities that they denote. The entity integration method combines semantic items from different sources into a single unified registry of entities by reconciling equivalence and distinctness information and solving a combinatorial optimization problem. Finally, the taxonomic integration method adds a comprehensive and coherent taxonomic hierarchy on top of this registry, capturing how different entities relate to each other.

Together, these methods can be used to produce a large-scale multilingual knowledge base semantically describing over 5 million entities and over 16 million natural language words and names in more than 200 different languages.

Kurzfassung

Da ein großer Teil unseres Wissens in textueller Form vorliegt, sind Informationssysteme in zunehmendem Maße auf Wissen über Wörter und den von ihnen repräsentierten Entitäten angewiesen. Gegenstand dieser Arbeit sind neue Methoden zur automatischen Erstellung großer multilingualer Wissensbanken, welche semantische Beziehungen zwischen Wörtern, Namen und Entitäten formal erfassen. In drei Hauptbeiträgen werden jeweils Indizien aus mehreren Wissensquellen mittels graphtheoretischer und statistischer Verfahren verknüpft.

Bei der lexikalischen Integration werden statistische Modelle zur Disambiguierung erlernt, um Wörter mit den von ihnen repräsentierten Entitäten in Verbindung zu setzen. Bei der Entitäten-Integration werden semantische Einheiten aus verschiedenen Quellen unter Berücksichtigung von Äquivalenz und Verschiedenheit durch Lösung eines kombinatorischen Optimierungsproblems zu einem kohärenten Register von Entitäten zusammengefasst. Dieses wird schließlich bei der taxonomischen Integration durch eine umfassende taxonomische Hierarchie ergänzt, in der Entitäten zueinander in Verbindung gesetzt werden.

Es zeigt sich, dass diese Methoden zusammen zur Induzierung einer großen multilingualen Wissensbank eingesetzt werden können, welche über 5 Millionen Entitäten und über 16 Millionen Wörter und Namen in mehr als 200 Sprachen semantisch beschreibt.

Summary

Much of our knowledge is expressed in textual form, and it is by keywords that humans most commonly search for information. Information systems are increasingly facing the challenge of making sense of words or names of objects, and often rely on background knowledge about them. While such knowledge can be encoded manually, this thesis examines to what extent existing knowledge sources on the Web and elsewhere can be used to automatically derive much larger knowledge bases that capture explicit semantic relationships between words, names, and entities, in many different languages. At an abstract level, such knowledge bases correspond to labelled graphs with nodes representing arbitrary entities and arcs representing typed relationships between them. The problem is approached from three complementary angles, leading to three novel methods to produce large-scale multilingual knowledge bases. In each case, graph algorithms and statistical techniques are used to combine and integrate evidence from multiple existing sources of information.

The *lexical integration* method considers the task of connecting words in different languages to the entities that they denote. Translations and synonyms in a graph are used to determine potential entities corresponding to the meanings of a word. The main challenge is assessing which ones are likely to be correct, which is tackled by learning statistical disambiguation models. These models operate in a feature space that reflects local contextual information about a word in the graph. This strategy allows us to turn an essentially monolingual resource like the commonly used WordNet database into a much larger multilingual lexical knowledge base.

The *entity integration* method addresses the problem of extending the range of potential entities that words can refer to by adding large numbers of further entities from separate sources. Given some prior knowledge or heuristics that reveal equivalence as well as distinctness information between entities from one or more knowledge sources, the aim is to combine the different repositories into a single unified registry of entities. Semantic duplicates should be unified, while distinct items should be kept as separate entities. Reconciling conflicting equivalence and distinctness information can be modelled as a combinatorial optimization task. An algorithm with a logarithmic approximation guarantee is developed that uses linear programming and region growing to obtain a consistent registry of entities from over 200 language-specific editions of Wikipedia.

Finally, the *taxonomic integration* method adds another layer of organization to this registry of entities, based on taxonomic relationships that connect instances to their classes and classes to parent classes. The central challenge is to combine unreliable and incomplete taxonomic links into a single comprehensive taxonomic hierarchy, which captures how entities in the knowledge base relate to each other. We achieve this by relying on a new Markov chain algorithm.

Together, these methods can be used to produce a large-scale multilingual knowledge base that substantially goes beyond previous resources by semantically describing over 5 million entities and over 16 million natural language words and names in more than 200 different languages.

Zusammenfassung

Da ein großer Teil des menschlichen Wissens in textueller Form vorliegt, und auch die Informationssuche primär durch Suchbegriffe erfolgt, sind Informationssysteme in zunehmendem Maße darauf angewiesen, Wörter und andere Begriffe semantisch interpretieren zu können, oftmals unter Zuhilfenahme von Hintergrundwissen. Gegenstand dieser Arbeit ist die Frage, inwiefern große multilinguale Wissendatenbanken automatisch anhand existierender Wissensquellen, etwa aus dem Web, erstellt werden können. Inhalt dieser Wissensbanken sollen unter anderem explizite semantische Beziehungen zwischen Wörtern, Namen und Entitäten sein. Konzeptuell gesehen handelt es sich somit um Graphen, deren Knoten beliebige Entitäten repräsentieren, und deren Kanten typisierte Beziehungen zwischen Entitäten wiedergeben.

Dieses Ziel wird aus drei komplementären Blickwinkeln betrachtet, welche zu drei neuen Methoden zur Erstellung multilingualer Wissensbanken führen. In jedem dieser Fälle wird auf graphtheoretische und statistische Verfahren gesetzt, um Indizien aus mehreren Wissensquellen zu verknüpfen.

Die *lexikalische Integrationsmethode* setzt sich zum Ziel, Wörter verschiedener Sprachen mit den von ihnen repräsentierten Entitäten zu verbinden. Potenzielle Kandidaten werden anhand von Übersetzungen und Synonymen bestimmt. Primäre Herausforderung ist die Beurteilung, welche der möglichen Kandidaten tatsächlich adäquat sind. Der gewählte Ansatz beruht auf statistischen Modellen zur Disambiguierung, deren Merkmalsräume kontextuelle Eigenschaften eines Wortes im Graphen wiedergeben. In der Praxis ermöglicht dies die Erweiterung einer monolingualen lexikalischen Ressource wie das vielfach verwendete WordNet zu einer wesentlich größeren multilingualen Ressource.

Ziel der *Entitäten-Integration* ist eine Erweiterung des Repertoires möglicher semantischer Entitäten, die durch Wörter repräsentiert werden können. Die Grundidee ist, verschiedene existierende Repertoires zu vereinigen, so dass äquivalente Einheiten verbunden und als verschieden bekannte Einheiten klar voneinander abgegrenzt werden. Schwierig wird dies aufgrund der Tatsache, dass sich Informationen über Äquivalenzen und Nichtäquivalenzen widersprechen können. Die Auflösung derartiger Widersprüche wird als kombinatorisches Optimierungsproblem formalisiert, für das ein Approximationsalgorithmus mit logarithmischer Approximationsgarantie vorgestellt wird. Der Algorithmus verwendet Lineare Programmierung und ein spezielles Regionenexpansionsverfahren, um aus über 200 sprachspezifischen Versionen der Enzyklopädie Wikipedia ein unifiziertes Register von Entitäten zu bilden.

Dieses wird schließlich bei der *taxonomischen Integration* durch eine zusätzliche Organisationsform erweitert. Mittels taxonomischer Relationen werden individuelle Instanzen mit Klassen und Klassen mit allgemeineren Oberklassen verbunden. Die Herausforderung hierbei ist die Verknüpfung unvollständiger und unzuverlässiger taxonomischer Einzelbeziehungen zu einer umfassenden kohärenten Hierarchie, in der alle durch die Wissensbank beschriebenen Entitäten zueinander in Verbindung gesetzt werden. Erreicht wird dies durch einen auf Markov-Ketten basierenden Algorithmus.

Es zeigt sich, dass diese Methoden in zusammenwirkender Form zur Induzierung einer großen multilingualen Wissensbank eingesetzt werden können, welche über 5 Millionen Entitäten und über 16 Millionen Wörter und Namen in mehr als 200 verschiedenen Sprachen semantisch beschreibt, und somit weit über den Rahmen früherer Ressourcen hinausgeht.

Acknowledgements

First and foremost, I would like to thank Gerhard Weikum for giving me the opportunity to carry out this research, providing the perfect balance between academic freedom on one side and excellent guidance and advice on the other. I wish to thank him and all of my colleagues at the Max Planck Institute for Informatics for providing such a pleasant and stimulating work environent. I am also grateful to all of my other collaborators for their inspiring discussions and ideas, among them Adam Pease, Stefan Siersdorfer, Fabian Suchanek, Martin Suda, Geoff Sutcliffe, and Niket Tandon. I would like to express my gratitude to the additional reviewers and examiners of my doctoral thesis, Hans Uszkoreit and Hinrich Schütze, and Gert Smolka. Last but not least, I am greatly indebted to Xian as well as my family and friends for their constant support throughout the years.

Contents

1 Introduction **1**
- 1.1 Motivation 1
- 1.2 Main Contributions 6
- 1.3 Outline 8

2 Graph-Based Knowledge Representation **11**
- 2.1 Knowledge Base Paradigms 12
- 2.2 Framework 17
- 2.3 Nodes and Entities 19
- 2.4 Arcs and Statements 23

3 Lexical Integration **27**
- 3.1 Introduction 27
- 3.2 Previous Work 31
- 3.3 Initial Graph Construction 34
- 3.4 Iterative Graph Refinement 41
- 3.5 Results 54
- 3.6 Discussion 75

4 Entity Integration **77**
- 4.1 Introduction 77
- 4.2 Previous Work 83
- 4.3 Knowledge Sources 86
- 4.4 Considering Distinctness Information 88
- 4.5 Approximation Algorithm 96
- 4.6 Results 107
- 4.7 Discussion 115

5 Taxonomic Integration **117**
5.1 Introduction 117
5.2 Previous Work 121
5.3 Knowledge Extraction 124
5.4 Linking Functions 126
5.5 Taxonomy Induction 133
5.6 Results 146
5.7 Discussion 160

6 Conclusion **163**
6.1 Summary 163
6.2 Outlook 164

List of Figures **165**

List of Tables **167**

List of Algorithms **169**

Bibliography **171**

Index **191**

CHAPTER 1

Introduction

1.1 Motivation

Semantic Knowledge. Information systems are increasingly expected to have some sort of knowledge about the world. When a user wishes to obtain a list of art schools in the UK ordered by founding year, the system should know that the Royal College of Art is an art school located in London, that London is located in the United Kingdom, and even seemingly trivial pieces of knowledge like the fact that '*UK*' refers to the United Kingdom. Additionally, the system needs to have access to explicit factual knowledge like the founding year of the Royal College of Art. For this reason, capturing information in the form of machine-readable semantic *knowledge bases* has been a long-standing goal in computer science, information science, and knowledge management. Well-known knowledge bases include WordNet (Fellbaum, 1998), Cyc (Lenat and Guha, 1989), and more recently DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), WikiTaxonomy (Ponzetto and Strube, 2008), and Freebase (Bollacker et al., 2008). At an abstract level, many of these can be thought of as directed graphs with nodes representing entities and labelled arcs representing their relationships.

Semantic resources of this sort have the potential to spark new technological developments in many different fields by allowing us to overcome the traditional knowledge acquisition bottleneck. WordNet, for instance, has been cited thousands of times, has given rise to large

multi-million dollar EU projects (Vossen, 1998; Atserias et al., 2004b; Tufiş et al., 2004; Vossen et al., 2008), and entire workshops and recurrent conferences have been dedicated to it (Bhattacharyya et al., 2010). Tasks that such semantic resources have already been shown to facilitate to date include query expansion (Gong et al., 2005), semantic search (Milne et al., 2007; Bast et al., 2007), and question answering (Schlaefer et al., 2007; Frank et al., 2007) in information retrieval, or machine translation (Chatterjee et al., 2005), document enrichment (Mihalcea and Csomai, 2007), syntactic parsing (Bikel, 2000; Fujita et al., 2007), and various other tasks (Harabagiu, 1998) in natural language processing. Additional applications include database schema matching (Madhavan et al., 2001), data cleaning (Kedad and Métais, 2002), biomedical data analysis (Rubin et al., 2006), textual entailment (Bos and Markert, 2005), mobile services (Becker and Bizer, 2008), Web site navigation (Kobilarov et al., 2009), visual object recognition (Marszałek and Schmid, 2007), and many more.

Multilingual Knowledge. Much of humanity's accumulated knowledge is expressed as textual data on the Web and elsewhere, and it is by means of keywords and phrases that humans most commonly search for information. For an application, these are initially just sequences of characters. However, if the knowledge base stores information about human languages and their lexicons (so-called *lexical knowledge*), these character sequences can be related to entities described more formally by the knowledge base and used in various ways.

The knowledge base could capture that, in English, the string '*UK*' refers to the United Kingdom, and that, in Mandarin, '艺术学院' means art school. With the increasing degree of Internet penetration all over the world, the English language represents a constantly decreasing fraction of the Web. China and the European Union each have greatly surpassed the US in the number of Internet users, and other regions are expected to follow. *Multilingual* knowledge bases address this development by capturing relationships between words and concepts in multiple languages, thereby making their semantic connections explicit. For example, an application could query the database to determine the relationship between the English word '*intern*' and the Spanish word '*becario*' in order to assess to what degree two news headlines are related. Knowing that the French words '*étudiant*', '*élève*', '*écolier*' are synonymous can aid in query

expansion. Knowing that '*lycée*', '*école*', '*université*', '*académie*' are all specific types of what is called an '*educational institution*' in English is helpful for question answering. Similarly, knowing that the French name '*Royaume-Uni*' refers to the United Kingdom is useful in cross-lingual information retrieval.

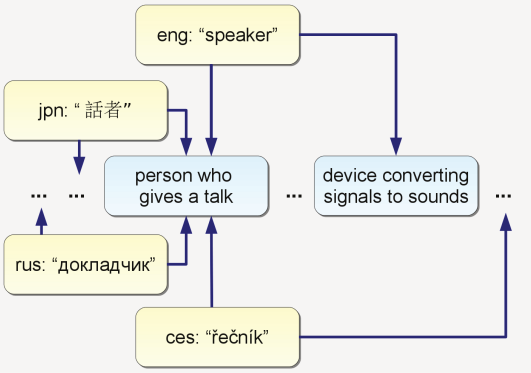


Figure 1.1: Universal index of meaning

Vision. With this in mind, the vision driving this thesis is the goal of establishing a universal multilingual knowledge base. Such a resource would include a universal index of meanings, where we envision being able to look up the meaning of any word or name in any language and obtain a list of its meanings. This is illustrated in Figure 1.1, where the English word '*speaker*' has two different meanings, and other words sharing one or more of those meanings are connected to the same meaning nodes whenever appropriate. Additionally, the meanings should be connected to each other in terms of different relations. The most important of these would be taxonomic relations that relate individual entities like `Stanford University` to classes like `University`, which in turn are linked to more general classes, in this case `Educational institution`, `Institution`, and so on, as shown in Figure 1.2. Applications can then more easily assess how different words and entities relate to each other. For instance, while Stanford University is a university and the École

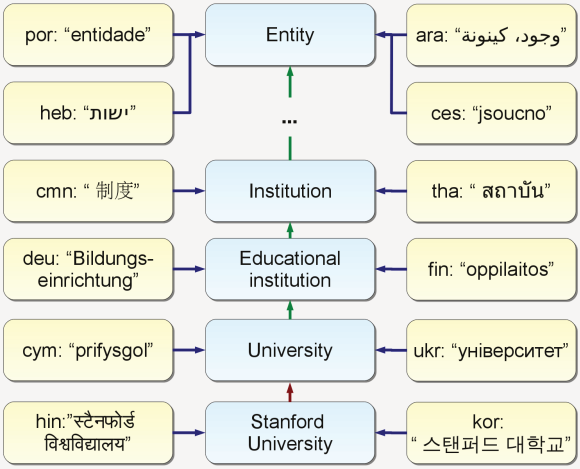


Figure 1.2: Hierarchical relations

Nationale Supérieure des Beaux-arts is an art school, both are educational institutions. A similar relationship can be identified between a Finnish ‘*ammattikorkeakoulu*’ and a Danish ‘*handelshøjskole*’. Unfortunately, existing knowledge bases have not been able to come sufficiently close to making this vision of a universal multilingual knowledge base a reality. Projects like EuroWordNet (Vossen, 1998) cover only a limited number of languages, while knowledge bases like YAGO, DBpedia, and WikiTaxonomy lack a multilingual taxonomic organization as well as large numbers of language-specific entities.

Paradigms. In the past, two opposing paradigms for creating lexical knowledge bases like WordNet and other semantic resources could be observed.

- **Manual compilation**: For many years, the dominating approach was to rely on human labour to manually supply knowledge to an information system, often by system experts working together with domain experts. There is a long history of lexicographical practices for compiling dictionaries, thesauri, and lexical

knowledge bases. Similarly, encyclopedic knowledge has been encoded manually in expert systems, in general-purpose knowledge bases like Cyc (Lenat and Guha, 1989), in ontologies like the biomedical OBO collection (Smith et al., 2007), and in search engines like Wolfram Alpha (*www.wolframalpha.com*).

- **Automatic methods**: In the past few decades, a separate line of research has considered the problem from a more empiricist perspective, aiming at inducing semantic information automatically from text using statistical methods (Schütze, 1992; Kilgarriff, 1997). For example, many projects have investigated creating thesauri by clustering words with respect to their context, based on the assumption that '*a word is characterized by the company it keeps*' (Firth, 1957). Examples include Pereira et al. (1993) who developed a form of hierarchical clustering based on distributional similarity, Schütze (1998) who proposed using vector spaces capturing more reliable second-order co-occurrences, and Lin (1998) who used dependency parse information to model the context. Significant research efforts have also been put into systems that attempt to harvest explicit relationships between words (Hearst, 1992; Snow et al., 2004; Girju et al., 2006; Tandon and de Melo, 2010) or between entities (Pantel and Pennacchiotti, 2006; Banko et al., 2007; Suchanek et al., 2009) from text.

Our Strategy. Both paradigms have their advantages and disadvantages. Manual building generally is a cumbersome, slow, and costly process that tends to lead to small, incomplete resources. At the same time, unsupervised automatic approaches have not been able to attain a comparable level of sophistication, giving us larger but generally more noisy and rather weakly structured knowledge bases. Our work attempts to take a pragmatic middle road, combining the best of both worlds.

- **High quality by relying on manually built resources:** Rather than starting from scratch, we make use of the fact that there are already many existing, highly curated sources of knowledge, including lexical databases like WordNet (Fellbaum, 1998) and machine-readable dictionaries like FreeDict. In recent years, we

have additionally seen the advent of semi-structured resources like Wikipedia and Wiktionary that are collaboratively created by large numbers of users on the Web. Projects like DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), and Freebase (Bollacker et al., 2008) have shown that these can be transformed into machine-readable knowledge bases.

- **High coverage by relying on graph-based methods:** The landscape of existing sources is vast, and apart from a few genuine lexical knowledge bases, includes many large knowledge sources that require additional processing to be useful for our purposes. We rely on graph-based methods to interlink the different knowledge sources and induce a single clean and coherent multilingual knowledge base. Carefully combining over 200 language-specific editions of Wikipedia as well as information from WordNet and translation dictionaries allows us create a unified knowledge base with a very broad coverage surpassing that of previous resources.

1.2 Main Contributions

These key insights lead us to graph-based algorithms and techniques that start out with multiple existing knowledge sources and heuristic methods, and then rely on structural and statistical properties of the input to produce much more valuable integrated knowledge bases. There are three complementary aspects that we tackle:

- **Lexical Integration:** One means of producing a large multilingual lexical knowledge base is taking an existing monolingual knowledge base with its corresponding inventory of meanings, and then using further knowledge sources to incorporate new words into it. If large numbers of additional words in many different languages are attached to those meanings, the resulting knowledge base becomes multilingual. As the existing monolingual database, we mainly rely on WordNet (Fellbaum, 1998), which in its original form describes commonly used English words and their meanings much like we outlined in our long-term vision, but does not cover languages other than English.

In this thesis, we show how WordNet can be expanded to capture over 1.5 million connections from words in many different languages to their meanings, greatly surpassing previous attempts at porting WordNet to other languages in terms of coverage. In terms of quality, we pursue a machine learning strategy that is much more sophisticated than previous automatic approaches, which relied on manually specified heuristic rules. This gives rise to the first universal version of WordNet that is not limited to a specific small set of languages.

- **Entity Integration:** Often, a single existing knowledge source will not exhaustively describe all the possible meanings we would like to consider, so we need to augment it with entities corresponding to further meanings taken from additional knowledge sources. Given multiple knowledge sources with overlapping inventories of entities as input, the challenge is to produce a single unified repository of entities.
  We propose an optimization model and an algorithmic framework to reconcile information about possible equivalences within and across data sources with information about distinctness of entities. Unlike most previous work on thesaurus and ontology mapping as well as record linkage, this framework accounts for distinctness between arbitrary subsets of entities from more than just two knowledge sources. In addition to having a logarithmic approximation guarantee for the objective function of the model, the algorithm is shown to produce even better near-optimal results in practice. We demonstrate how this framework can be applied to generate a unified database of entities from over 200 multilingual editions of Wikipedia.
- **Taxonomic Integration:** Additionally, we may have different sources and heuristics identifying taxonomic relationships between entities. Such links include `instance` links between individual entities and the classes they are members of, e.g. `Paris` is an instance of a `City`. They also include `subclass` links that connect classes to more general parent classes, e.g. `City` and `Geopolitical entity`. We propose an algorithm called Markov Chain Taxonomy Induction to integrate an incomplete, unreliable set of individual taxonomic links into a single, more consistent taxonomy. The

experiments indicate that this algorithm is able to yield output that is of higher quality than its initial noisy input. We show that, in conjunction with a set of linking heuristics, we can use this algorithm to create a large multilingual taxonomy of entities. Together with additional information from the lexical integration step as well as encyclopedic factual knowledge from Wikipedia, this gives us a large multilingual knowledge base that goes far beyond previous resources by semantically describing over 5 million entities with over 16 million natural language words and names in different languages, realizing much of the long-term vision of a universal multilingual knowledge base outlined earlier. The resulting UWN/MENTA resource is freely available for download at `http://www.mpi-inf.mpg.de/yago-naga/menta/`.

Some results of this thesis have been published in the proceedings of international conferences and in international journals, including among others:

- CIKM 2009 (de Melo and Weikum, 2009b)
- ACL 2010 (de Melo and Weikum, 2010d)
- CIKM 2010 (de Melo and Weikum, 2010a) – Best Interdisciplinary Paper Award
- GWC 2008 (de Melo and Weikum, 2007)
- ICGL 2008 (de Melo and Weikum, 2008b) – Best Paper Award
- LREC 2008 (de Melo and Weikum, 2008c)
- GWC 2010 (de Melo and Weikum, 2010c)
- LREC 2010 (de Melo and Weikum, 2010b)
- Springer Journal Language Resources and Evaluation (de Melo and Weikum, 2011) – to appear

1.3 Outline

The organization of this thesis reflects these central contributions. Chapter 2 begins by introducing the idea of capturing knowledge in labelled graphs and formally defines our knowledge representation framework. Chapter 3 describes how words and other lexical items from different languages can be integrated into an existing lexical knowledge base.

Chapter 4 proposes an algorithm that integrates semantic entities from different lexical knowledge bases, based on information about their equivalence and distinctness. Chapter 5 investigates how taxonomic relationships between entities can be integrated to produce more coherent knowledge bases, and presents the final large-scale multilingual knowledge base that we obtain using our methods. Finally, Chapter 6 concludes by discussing the implications of the presented results.

CHAPTER 2

Graph-Based Knowledge Representation

Representing Knowledge. Since the beginning of computing, people have sought to make their systems operate on *representations* of real world phenomena, from simple Boolean and integer variables to sequences of integer codes representing text strings, to identifier strings that in turn represent people or cities and structured data models representing knowledge about such entities.

Users now routinely expect their systems to behave in a way that seems intelligent in some sense. For example, a word processor is generally expected to recognize '*accomodation*' as a misspelling of '*accommodation*'. A search engine might be expected to find the Web site of a '*Used Vehicles Dealer*' when we search for '*buy used cars*'. Increasingly, we also want our search engines to respond with '*Brasília*' to a query like '*capital of Brazil*', or to be able to provide a list of Chinese cities sorted by population size.

Often, knowledge required by an application is encoded explicitly into the program code or recorded in program-specific data files. At the same time, there have been endeavours to create resources capturing knowledge that can be re-used in different contexts, from spell checking libraries all the way to modern knowledge bases like WordNet (Fellbaum, 1998), DBpedia (Auer et al., 2007), and YAGO (Suchanek et al., 2007).

Knowledge Bases. A *knowledge base* is a database holding machine-readable representations of knowledge. Some of the well-known knowledge base paradigms will be introduced in Section 2.1. In this thesis, we consider knowledge bases as graphs that represent relationships between entities, including but not limited to lexical relationships between natural language words (or names of objects) and their possible meanings, as well as ontological relationships that form a taxonomic hierarchy of entities. A knowledge base of this form could describe an entity `CMU` as having the names '*CMU*' and '*Carnegie Mellon University*' in English and '卡内基梅隆大学' in Chinese. Additionally, it could describe `CMU` as an *instance* of a `University`, `University` as a *subclass* of `Educational institution`, `Educational institution` as a subclass of `Institution`, and so on, up to the taxonomy's most general root node, often called `Entity`. This is made more formal in Section 2.2.

2.1 Knowledge Base Paradigms

Before delving into the details of our framework, we survey the spectrum of existing knowledge representation paradigms and simultaneously clarify their relationship to our framework.

2.1.1 Lexical Knowledge Bases

Lexical knowledge bases are knowledge bases that focus on describing a particular aspect of the world, the realm of words and their relationships. We earlier saw examples in Figures 1.1 and 1.2 in Chapter 1 of how words can be regarded as having certain meanings and meanings can be related to other meanings.

WordNet. Resulting from research under the direction of George Miller at Princeton University, WordNet (Fellbaum, 1998) is a well-known lexical database for the English language. WordNet captures information about English words and their meanings (word senses) as well as semantic relationships between words or word senses. WordNet 3.0 consists of approximately 150,000 terms (words or short expressions) and around 120,000 so-called *synsets*. A synset is a set of words that express

the same concept or meaning, and in our framework will correspond to a so-called *semantic entity*.

Relations include word-to-synset relations that connect words with their possible senses (`means` in our framework) and vice versa. Additionally, binary relationships between synsets are captured. The hypernymy relation, for example, in its original form holds between a more specific term and a more general one, e.g. '*school*' has '*educational institution*' as a hypernym. In WordNet, this relation is expressed at a more abstract level as a relation between synsets, which will roughly correspond to the so-called `subclass` relation in our framework. Similarly, WordNet provides meronymic relations between synsets that can be reinterpreted as mereological part/whole relations (`partOf`) between the entities corresponding to the respective synsets.

Thesauri. A *thesaurus*, according to the ANSI/NISO Z39.19 standard, is 'a controlled vocabulary arranged in a known order and structured so that the various relationships among terms are displayed clearly and identified by standardized relationship indicators' (ANSI/NIZO, 2005). Synonyms are grouped together, whereas homonyms are distinguished. Well-structured thesauri in this sense can be regarded as WordNet-like lexical databases and hence can easily be cast into our framework.

In contrast, the kind of thesauri used by laypeople and professional authors as writing aids tend to be of a somewhat different flavour. Thesauri of this sort often provide alphabetical or thematically organized registers of headwords with lists of rather loosely related terms for each headword rather than synonyms with equivalent meanings. Roget's Thesaurus, first published in the 19th century, is the most famous such resource for the English language, which we examine in further detail later on in Section 3.5.6 (p. 65).

Etymological Word Networks. Etymology is the study of word origins. For example, the English word '*doubtless*' is derived from '*doubt*', which comes from Old French '*douter*', which in turn evolved from the Latin word '*dubitare*'. Such relationships are often expressed very verbosely, and even digital standards like TEI P5 (Burnard and Bauman, 2009) only define a semi-structured representation of etymological knowledge. In de Melo and Weikum (2010c), we showed how ety-

mological and derivational relationships between words can often be exposed much more clearly using network-like knowledge graphs as will be defined shortly. Navigating such a graph, one can easily uncover interesting connections, e.g. the historical connection between the English word '*muscular*' and the German word '*Fledermaus*' (bats in the biological sense), shown in Figure 2.1. Recursively parsing the semi-structured resource Wiktionary, we were able to obtain a graph with 1,000,000 terms, 200,000 etymological links between terms, and 1,700,000 derivational links between terms, freely available for download from `http://www.mpi-inf.mpg.de/~gdemelo/etymwn/`.

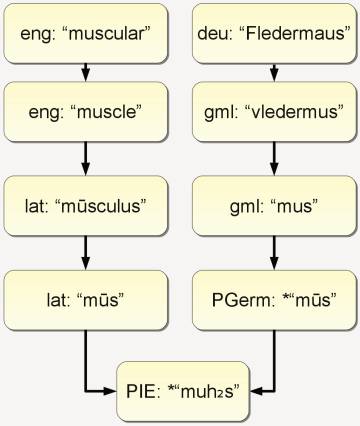


Figure 2.1: Excerpt from Etymological WordNet

2.1.2 Formal Knowledge and Data Models

Ontologies. An *ontology*, from ancient Greek '*-λογία*' (science) and '*ὄντος*' (of being), is a theory of what possesses *being*, i.e. exists, in the world or in a limited domain. In computational applications, ontologies provide formal descriptions of entities and are used to support the sharing and reuse of knowledge by different applications. Gruber (1993) characterizes an ontology as an 'explicit specification of a conceptualization' and refers to Genesereth and Nilsson (1987) who define a

conceptualization as 'the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them'. Still, there is a considerable degree of dissent on what precisely constitutes an ontology. Traditionally, formal languages like the *Knowledge Interchange Format* (KIF) were in common use, aiming at providing axiomatic descriptions. In recent years, formalisms based on subject-predicate-object triples like the *Web Ontology Language* (OWL) have dominated. Such triples are often regarded as labelled graphs and for the most part can be cast into our knowledge representation framework. While our framework does not formally specify any logical entailments, applications are free to apply additional reasoning on top of what is explicitly captured in a given graph.

Generally, formal ontologies make use of symbols that represent entities, classes of entities, or logical constructions. The relationship between these symbols and what they represent is called the *interpretation*. In computational settings, symbols representing entities (or classes) are often called *entity identifiers*. While the symbols for logical operations are normally standardized in advance by frameworks like OWL, the interpretation of most entity identifiers is specific to particular ontologies.

Since these entity identifiers can be chosen arbitrarily, many OWL ontologies essentially contain nothing more than information of the form: `C87` is a subclass of `C34`, `C34` is a subclass of `C0`, and so on. Fortunately, lexical knowledge aids in restricting the range of possible interpretations of such identifiers. For example, if we assume that the meaning of `means` is known a priori (the relation holding between a word and its meaning) and that entity identifiers for character strings or words are interpreted in the standard way, then a statement about the English term '*pupil*' standing in a `means` relation to `C87` reveals that `C87` can only refer to entities that are called '*pupil*' in the English language. This is still ambiguous, because '*pupil*' could refer to the hole in the iris of an eye, or to students. In a multilingual knowledge base, we may find another statement expressing that the French term `étudiant` stands in the same relation to `C87`, which reduces or perhaps eliminates the ambiguity.

The Semantic Web. The Semantic Web is a proposal by Tim Berners-Lee to extend the existing World Wide Web, which consists mainly of HTML pages for human consumption, with additional machine-

processable knowledge. *Uniform resource identifiers (URIs)* are used not only to refer to traditional Web resources such as Web pages but also to so-called *non-information resources* like people, organizations, and other entities. URIs are entity identifiers that can be used globally, across individual knowledge sources, in a global shared namespace.

The *Resource Description Framework* (Hayes, 2004), or RDF for short, provides a standard model for expressing knowledge about such entities. RDF statements are based on triples consisting of a subject, a predicate, and an object. For instance, the well-established predicate `dcterms:creator` (Nilsson et al., 2008) enables us to express that Leonardo da Vinci is the creator of the Mona Lisa. The triple-based formalism means that RDF data can easily be cast into a graph-based knowledge base as defined later on, if we allow arbitrary URIs and RDF literals as nodes and assign additional identifiers to RDF's so-called blank (or anonymous) nodes.

Conversely, knowledge bases in our framework can easily be brought into an RDF form, if so-called reification (Hayes, 2004) is used to capture the statement weights that we include in our model. Additionally, new URIs may need to be defined to represent the entities identified by the nodes and by the arc labels in our framework. In fact, our Lexvo.org project (de Melo and Weikum, 2008a) has already defined re-usable global URIs for most of the relevant entities. The term URIs defined by Lexvo.org also address the problem that, in RDF, string literals currently may not serve as the subject of a statement, which makes it difficult to express knowledge about terms. Lexvo.org is part of the emerging Linked Data Web (Bizer et al., 2009), an effort to create a Web of Data that makes large amounts of interlinked datasets available using Semantic Web standards.

Relational Databases. Much of the world's digital data is stored in relational databases. The underlying relational model (Codd, 1970) is based on relations saved in tables consisting of rows and columns. Rows store records with multiple fields, each associated with a column of the table. For example, a record could describe a person and the individual fields could correspond to the first name, last name, employer, address, and date of birth.

In general, this model is content-neutral and may be used to store arbitrary kinds of data. Normally, however, a database schema is not merely an abstract syntactic template for a set of tables but is derived from a conceptual analysis of a particular domain and intended to represent some aspect of the world. Following Codd (1990, p.4), every row coupled with the corresponding relation name can be regarded as representing an assertion. Reification (Hayes, 2004) allows us to break down n-ary relations with $n > 2$ into binary relations. Relational data can thus be cast into our framework if care is taken to specify what the specific entity identifiers are.

For example, assume we have a row about a person with columns for the person's last name (stored as a string) and her employer (as a so-called foreign key referencing rows in another `Employer` table). We can assign arbitrary entity identifiers to rows of this table and of the separate `Employer` table, and then express as two binary relationships that `Person1234` stands in a `hasLastName` relationship with a string like '*Doe*', and in a `worksFor` relationship with an employer `Company123`. Sahoo et al. (2009) provide an extensive survey of techniques to map relational databases to triple- or graph-based representations. Conversely, knowledge modelled in terms of graphs can easily be stored in a relational database, e.g. if one wishes to harness the advanced querying capabilities of relational database management systems.

2.2 Framework

Requirements. The framework adopted in our work is intended to be generic and flexible enough to capture lexical knowledge as given by WordNet as well as simple formal knowledge as captured in ontologies of the more *lightweight* sort without complex axioms. Knowledge bases adopting graph- or triple-based representation paradigms generally assume that our world can be described in terms of discrete entities and binary relationships between entities. In such frameworks, more complex descriptions, e.g. in terms of sophisticated first or higher-order logic rules and axioms, would have to be encoded into node or arc labels rather than being first class citizens. In Section 5.6.6, we discuss an extension of our work that is integrated with a more axiomatic formal ontology.

Entities and their Relationships. Knowledge bases describing relationships between entities can quite naturally be regarded as labelled graphs, as we saw in Figures 1.1 and 1.2 in Chapter 1. In such graphs, nodes are *entity identifiers* that refer to arbitrary entities, including individual entities like `CMU`, abstract classes and conceptualizations like `University`, and words or names used in human languages like '*university*' and '*Carnegie Mellon University*'. It is important to stress that these entities need not possess any sort of physical existence, e.g. we typically specify entity identifiers for numbers like 15213 (which happens to be the ZIP code of Carnegie Mellon), and we could indeed even define an identifier for Scotty, CMU's mascot Scottish terrier.

Arcs in such a graph represent *statements* about entities. Arcs are given labels like `means` or `instance` that reveal to us which specific relationships hold between two entities. Additionally, they are assigned weights in order to characterize the confidence we have in the corresponding statements. Formally, multiple relations can simultaneously hold between two entities, so we need to allow multiple arcs between two nodes, leading to the following definition.

Definition 2.1 (Knowledge Base) *A* knowledge base *is a weighted labelled multi-digraph* $G = (V, A, \Sigma)$ *where:*

- V *is a set of entity identifiers that constitute the nodes in the graph*
- $A \subseteq V \times V \times \Sigma \times \mathbb{R}_0^+$ *is a set of weighted labelled arcs (that may include multiple arcs between two nodes as well as loops, i.e. arcs from a node to itself)*
- Σ *is the labelling alphabet for arcs, i.e. the set of possible arc labels (which represent relationships between entities)*

Semantics. The nodes of the graph are entity identifiers that represent arbitrary entities, while arc labels r are entity identifiers that represent arbitrary relations between entities. Specific examples are given below. An arc $a = (u, v, r, w) \in V \times V \times \Sigma \times \mathbb{R}_0^+$ expresses that the two entities represented by the nodes u,v are assumed to stand in a relationship given by r to each other with weight w. A weight of 0 means there is no evidence for this, and strictly positive values quantify the degree of confidence in the statement being true.

Definition 2.2 (Neighbourhood) *For brevity, we use the notation*

$$\Gamma_{\mathrm{i}}(v, A) = \{v' \mid \exists l, w : (v', v, l, w) \in A\}$$

to denote the in-neighbourhood, and

$$\Gamma_{\mathrm{o}}(v, A) = \{v' \mid \exists l, w : (v, v', l, w) \in A\}$$

for the out-neighbourhood of a node, given a set of arcs A.

2.3 Nodes and Entities

2.3.1 Overview

In some frameworks, a distinction is made between individuals and classes. As described earlier, we accept a very broad definition of entities, that includes both of these categories. More specifically, in the knowledge bases we describe in the following chapters, the set of nodes V will generally include the following sets of entity identifiers.

a) $T \times L$: For *term nodes* representing lexical items (words or expressions, or *term entities* in general) in a specific language, where T is the set of NFC-normalized Unicode character strings (Davis and Dürst, 2008), and L is the set of ISO 639-3 language identifiers. For instance, the English word '*school*' would be stored as a tuple ('*school*', `eng`).
b) $S \times C$: For *semantic nodes* that represent *semantic entities* (individual named entities as well as concepts), where S is a set of meaning (or sense) identifiers as provided e.g. by Princeton WordNet 3.0, and C is the set of lexical categories (`noun`, `verb`, `adjective`, etc.). For example, the principal meaning of the English word '*school*' corresponds to a semantic node (`8276720`,`noun`), where the number is taken from WordNet 3.0's internally used offsets. This entity identifier describes schools as educational institutions (as opposed to schools as buildings, for instance, for which there is a separate semantic node).
c) Additional semantic nodes based on Wikipedia, representing the subject of a Wikipedia page, as discussed in Chapter 5. Examples

include `Stanford University` representing the well-known university, as described by the English Wikipedia, or `de:Helmholtzschule` representing a specific school in Germany, as described by the German Wikipedia.

2.3.2 Term Entities

Characterization. As term entities or simply *terms* we consider all kinds of lexical items, including regular words (e.g. '*school*', '*college*'), multi-word expressions (e.g. '*primary school*', to '*drop out*'), and idiomatic expressions (e.g. to '*learn the hard way*'). When devising entity identifiers for terms, different levels of abstraction can be considered. For the term entities, we choose to consider two homonyms, e.g. the animal noun '*bear*' and the verb '*bear*', as the same term entity, because, typically, one wishes to look up terms in the knowledge base without already knowing what senses exist. Such distinctions are made only at the level of semantic entities, not for term entities. In contrast, we do consider the Italian term '*burro*', which means '*butter*', distinct from the Spanish term '*burro*', which means '*donkey*'. With this level of abstraction, relationships between words in different languages correspond directly to arcs between nodes in the graph.

Normalization. There are a few subtleties of term identity with respect to string encoding. For multilingual applications, the ISO 10646 / Unicode standards offer an appropriate set of characters for encoding words and expressions from a wide range of writing systems. Unicode allows storing a character like '*à*' in either a composed ('*à*') or in a decomposed form ('*a*' + '`'), with even more complex compositions for languages like Arabic and Vietnamese. To avoid duplicate entity identifiers, we consider two strings identical if they match after NFC normalization (Davis and Dürst, 2008) is applied to bring them into a canonical form. In practice, this means that terms taken from one source can correctly be identified with terms from another source, and a lookup will not fail just because of different encoding choices.

2.3.3 Semantic Entities

Characterization. As semantic entities, we consider all entities that could correspond to meanings of terms. While, in principle, terms could also refer to other terms, in practice, the semantic entities we deal with will be based on WordNet synsets and Wikipedia pages, so the set of semantic entities will be disjoint from the set of term entities.

Underlying most lexical knowledge bases is the assumption that possible meanings of a word can be enumerated discretely. While it is clear that there is no single correct way of drawing the lines, this simplifying assumption facilitates not only many computational applications but also underpins much of our human reliance on dictionaries. We do not require such enumerations to be non-overlapping or exhaustive.

Our framework remains somewhat agnostic as to the precise nature of word semantics. In most model-theoretic knowledge bases, people like John Locke or buildings like the Duke University Chapel would be considered real-world entities, while identifiers like `Academic freedom` would be interpreted as referring to some sort of abstract concept. Terms like '*Socrates*', '*Troy*', and '*Hesperus*' demonstrate that it is not always obvious in which cases we can consider only physical referents. Even cities like Cambridge and London have not always had clearly defined physical boundaries. In our framework, whenever semantic entities are regarded as having some sort of conceptual aspect, if suffices to interpret factual relationships (like `locatedIn`) accordingly (i.e. as applying with respect to entities of that sort).

Multilingual Generalization. In the EuroWordNet approach (Vossen, 1998), additionally adopted for BalkaNet and other related projects (Tufiş et al., 2004; Atserias et al., 2004b), each individual wordnet has its own inventory of semantic entities, and a separate interlingual index (ILI) is intended to serve as an external language-neutral register of semantic entities. Whenever possible, entities from the individual wordnets are linked to the ILI by means of equivalence and near-equivalence relations.

Such a representation can be transformed into one where terms in different languages are directly connected to the same semantic entity whenever the respective meaning can be regarded as being realized in multiple languages. The underlying idea is that two words can often be

thought of as *sharing* the same sense when they are near-synonymous or translational equivalents of each other with respect to specific contexts. Such sharing is in fact one major difference between lexical knowledge bases like WordNet and conventional dictionaries in the first place: In WordNet, synonymous terms like '*bus*' and '*coach*' in Figure 2.2 are tied to a single shared semantic entity identifier, while in traditional dictionaries the respective meanings are listed in distinct, unconnected entries. What WordNet does for synonymous terms within a language can be generalized to terms across languages. Figure 2.3 provides an example of this idea. We see additional words in other languages linked to the same semantic entities as the English words in Figure 2.2. Additionally, there are `language` arcs as dotted lines that link from terms to semantic entities for languages.

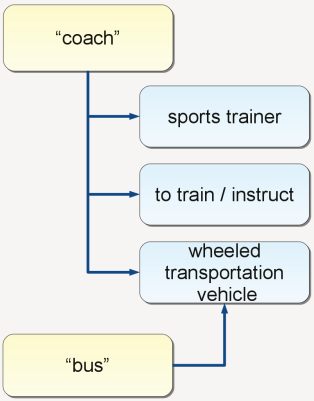


Figure 2.2: Monolingual lexical knowledge

Language-Specific Idiosyncrasies. It must be pointed out that this principle by no means impels us to neglect language-specific subtleties. Distinct semantic entities may co-exist whenever semantic differences persist. For example, if in one language the word for '*tree*' has a meaning that includes shrubs, then the semantic entity that embodies this meaning should not be conflated with the semantic entity for the meaning of

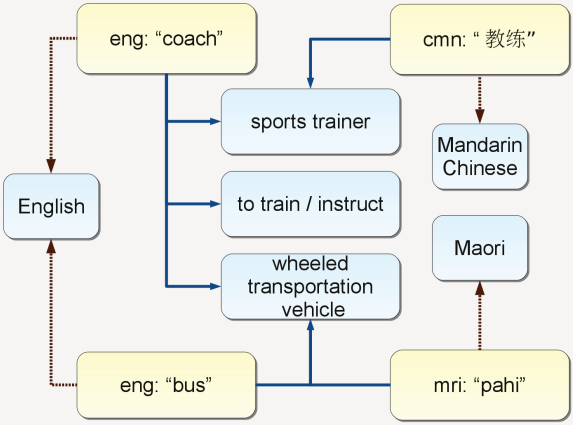


Figure 2.3: Multilingual lexical knowledge

the English word ‘*tree*’, which generally is not taken to include shrubs. In a similar vein, if in one language birds and insects are considered animals and in another they are not, then there are actually distinct concepts of animals that need to be demarcated. This is similar to how the vernacular English concept of ‘*nuts*’ should ideally be distinguished from the corresponding botanical concept, which is also part of the English language but excludes peanuts and almonds. Chapter 4 addresses this problem of separating conflated concepts in greater detail.

2.4 Arcs and Statements

In our knowledge base, the set of acceptable arc labels Σ will encompass the following subsets.

a) $\{\texttt{means}\} \times \mathbb{N}_0 \times \mathbb{N}_0$: For arcs representing relationships between terms and semantic entities corresponding to their meanings, with:
   - a value in $\mathbb{N}_0$ representing the synset rank of the synset in WordNet (1 for the first, most relevant sense, 2 for the second, and so on), or 0 if no rank information is available

  - a value in $\mathbb{N}_0$ representing sense frequencies in a corpus, or 1 if no such information is available (cf. Section 3.3.1)

  For instance, the English word '*head*' can mean a specific part of a body, and at the same time, it can also stand in a `means` relation to a semantic entity for people who are in charge of something (as in '*the head of the department*'). The terms connected to a semantic entity are referred to as the *lexicalizations* of the semantic entity.

b) $\{$`language`$\}$: For arcs from a term to a semantic entity characterizing the language of that term, e.g. the English word '*head*' could be linked to the entity `English`, and the Maori word '*pahi*' would be linked to `Maori`.

c) $\{$`translation`$\} \times C \times C$: For term-to-term arcs that represent translational equivalence and connect term nodes to other term nodes corresponding to their translations into other languages, with source and target lexical categories in C (e.g. `noun`, `verb`, etc., or most commonly `unknown` if no such information is available). For example, the Chinese word '教练' is a translation of the English word '*coach*', where both words are nouns. However, translation arcs do not reveal whether '*coach*' in this context refers to a sports trainer or to a wheeled transportation vehicle – the answer is provided by the `means` relationships in Figure 2.3.

d) $\{$`synonym`$\} \times C \times C$: For arcs representing (near-)synonymy, with source and target lexical categories in C.

e) $\{$`related`$\}$: For term-to-term arcs that provide generic indications of semantic relatedness, e.g. between '*teach*' and '*university*'.

f) $\{$`equals`$\}$: For arcs between two nodes representing the same entity, e.g. sometimes WordNet and Wikipedia both describe the same entity, and an `equals` arc can be used to connect the two respective entity identifiers.

g) $\{$`subclass`$\}$: For arcs between two semantic nodes u, v when v denotes a subsuming generalization of the semantic entity associated with u, e.g. u could denote high schools and v could denote educational institutions in general. We interpret the `subclass` relation as slightly broader than the type of formal subsumption considered in axiomatic ontologies in order to be closer to WordNet's hypernymy relation between synsets. Ontologically, the two

entities need not be what one would typically consider classes of instances, e.g. u could also represent *common knowledge* and v could represent *knowledge* in general.

h) {`instance`}: For arcs between two semantic nodes u, v, when u refers to a single instance of the type designated by v (its class, type, or role), e.g. u could refer to the Berkeley Sather Tower and v could represent towers or buildings.

i) {`partOf`, `opposite`, ...}: Additional semantic relationships derived from WordNet or other sources.

j) {`hasGloss`}: For arcs from a semantic node to a node consisting of a human-readable string defining or at least characterizing the meaning.

k) {`locatedIn`, `bornIn`, ...}: For arcs representing factual knowledge about entities that can be extracted from Wikipedia.

Different chapters will emphasize different relations. In Chapter 3, we consider a lexical integration strategy, where new terms are integrated into a knowledge base using the `means` relation. In Chapter 4, we integrate entities from different sources and pay special attention to the `equals` relation that connects equivalent nodes. In Chapter 5, we additionally interlink entities that are not equivalent by means of the taxonomic relations `subclass` and `instance`.

CHAPTER 3

Lexical Integration

3.1 Introduction

One way of obtaining a large-scale multilingual knowledge base is to start out with a monolingual one and integrate large numbers of additional words in different languages into it by attaching them using the `means` relation. This idea, which is sketched in Figure 3.1, will be pursued in this chapter.

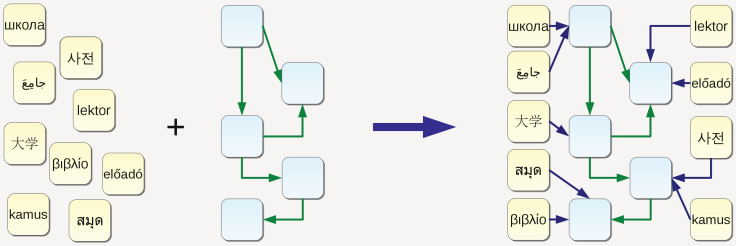


Figure 3.1: Lexical integration strategy (simplified)

Motivation. This chapter will show that one can take a small, essentially monolingual knowledge base and use statistical methods to derive a large-scale multilingual lexical database that organizes over 800,000

words from over 200 languages in a hierarchically structured semantic network. This universal wordnet, *UWN*, provides over 1.5 million disambiguated links from words to semantic entities, and addresses a large part of the applications described in Chapter 1.

UWN is bootstrapped from the original Princeton WordNet, a well-known lexical database for the English language (Fellbaum, 1998) that we introduced earlier in Section 2.1. As a reminder, WordNet describes around 150,000 terms (words or short phrases) and about 120,000 semantic entities ("synsets" in the original terminology). It connects terms to semantic entities reflecting their meanings, thus providing a fairly comprehensive database of *synonymy* and *polysemy*. Additionally, it interlinks these semantic entities using semantic relationships like *hypernymy*, which is similar to the `subclass` relation and hence induces a hierarchical organization, as well as the *meronymy* (part/whole) relation, among others. For instance, '*university*' and '*high school*' are hyponyms of '*educational institution*' (see also Figure 3.2), and '*classroom*' is regarded as a meronym (part) of '*schoolhouse*'. We use the name '*WordNet*' to refer to the original version created at Princeton University, in contrast to the generic term *wordnet*, which includes other WordNet-like knowledge bases.

Having lexical knowledge for a given language is an important requirement in many different applications. Fellbaum (1998) has been cited several thousand times, and recent editions of the Language Resources and Evaluation Conference (LREC) have attracted over 1,000 participants. Similar wordnets do exist for about 50 different languages, but none of them are nearly as complete as the original English WordNet – in fact, a large number are small and unmaintained. Moreover, for many actively used languages, no such lexical databases exist at all.

Our work not only addresses this gap but additionally goes beyond the notion of monolingual wordnets by constructing an integrated multilingual wordnet that maps terms (words, phrases) of many languages to their meanings in the language-independent space of semantic entities (essentially concepts). This allows, for example, finding Greek generalizations of the German word '*Hochschule*' (university, college) or Korean words expressing the opposite of the French word '*grand*' (big). An application can discover that the Swahili word '*darasa*' refers to something that is part of a '*schoolhouse*': a classroom. Knowledge of this

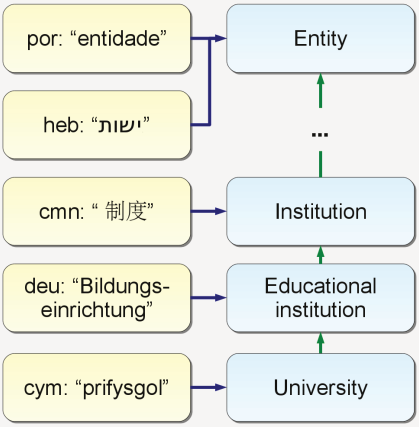


Figure 3.2: Semantic relations – A university is a kind of educational institution, which is a kind of institution, and so on. Additionally, terms in many languages should be linked to each semantic entity.

sort is useful for query expansion, faceted browsing, opinion mining, and many other applications. This level of semantic connections and support for IR and AI tasks can never be reached by a mere translation dictionary between two languages.

Problem Statement. The input will consist of i) existing, possibly monolingual lexical knowledge bases like WordNet, and ii) additional sources like translation dictionaries, thesauri, and parallel corpora, which provide a significant quantity of simple lexical data, consisting mostly of `translation` (or `synonym`) statements. The output should be an extended knowledge base, where terms in different languages from the lexical data sources have been integrated into the knowledge base. Both input and output can be represented as graphs.

Figure 3.3 illustrates the central challenges. Part (a) depicts the input coming from monolingual lexical knowledge bases. Arrowed lines represent `means` arcs from term nodes to semantic nodes. Part (b) shows the input graph G_0 after adding `translation` arcs that can be

derived from bilingual translation dictionaries (each non-arrowed line represents two reciprocal `translation` arcs). Part (c) gives the desired output graph where several words in different languages originally only linked indirectly via `translation` arcs have been connected to the semantic nodes that represent their disambiguated meanings (via dotted lines), leading to a more multilingual knowledge base. The same is possible using `synonym` arcs instead of `translation` ones, when one is interested in integrating missing synonyms in the same language into a lexical database.

Due to the ambiguous nature of words, the central difficulty is determining which semantic entities apply to which translations (or synonyms). For example, a simple English word like '*class*' has 9 meanings listed in WordNet, '*form*' has 23 meanings, and there are examples such as the word '*break*', for which 75 different meanings are enumerated.

Contribution. We present a framework that accomplishes this task using statistical learning techniques. This symbiosis of relying on preexisting manually compiled knowledge and automatic statistical techniques turns out to be particularly fruitful. A machine learning approach can solve the disambiguation challenge with much greater success than reported in previous studies. Factors that contribute to this include careful feature engineering, more evidence by considering a single large multilingual graph and relying on multiple iterations, as well the power of the learning algorithm to benefit even from weak signals, much better than typical hand-crafted rules. We show that this approach leads to the first massively multilingual version of WordNet.

Overview. Our method for building UWN starts with a limited number of existing (monolingual) lexical knowledge bases to derive a large set of possible word meanings, represented in a graph G_0 of term nodes and semantic nodes (cf. Section 2.2). This graph is extended by extracting information from a range of sources including translation dictionaries and thesauri, as well as by applying automatic preprocessing procedures. Statistical methods are then used to link terms in different languages to adequate semantic entities by analysing this graph. We attempt to discern disambiguation information in a series of graph refinements. To this end, we construct a rich set of numeric features for

assessing the validity of candidate arcs being considered for inclusion in the output graph. We train a support vector machine (SVM) over this feature space with a small number of hand-labelled arcs. Then the SVM can automatically discriminate arcs that are likely to be valid from spurious ones. The algorithm runs iteratively, i.e. several graphs G_i may be constructed, each refining the previous graph G_{i-1} by recomputing features and re-applying the SVM learner.

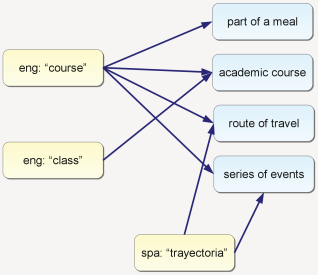
The rest of the chapter is organized as follows. Section 3.2 reviews WordNet and related work. Section 3.3 describes the initial graph construction phase. Section 3.4 presents the feature space and learning model for graph refinement. Section 3.5 shows experimental results that confirm the high recall and precision of our method, and demonstrates the benefit for tasks like cross-lingual text classification. Section 3.6 summarizes and discusses the implications of these results.

3.2 Previous Work

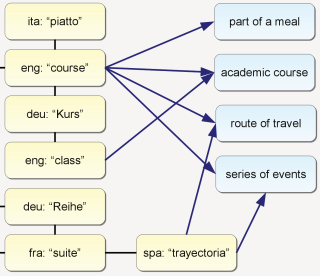
WordNet. The original WordNet (Fellbaum, 1998) was manually compiled at Princeton University to evaluate hypotheses about human cognition, but eventually became one of the most widely used lexical resources in natural language processing. WordNet is the fruit of over 20 years of manual work. For information about its internal structure, see Section 2.1.

Non-English Wordnets. The original WordNet has sparked a number of endeavours aiming at similar databases for other languages, most importantly perhaps the EuroWordNet (Vossen, 1998) and BalkaNet projects (Tufiş et al., 2004) that targeted many European languages. Individual institutes have made similar efforts for further languages, often under the auspices of the Global WordNet Association. Unfortunately, the work on such resources has not resulted in a unified multilingual wordnet, as there are different meaning identifiers, formats, licences, etc.

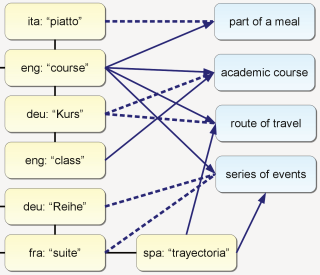
Previous attempts to address this situation are still in their infancy. Marchetti et al. (2006) proposed a Semantic Web tool for managing and interlinking wordnets in order to create a multilingual grid, however they do not focus on the problem of actually populating this grid. Another ambitious project that started in 2006, the Global Wordnet Grid



(a) Input from (English, Spanish) wordnets



(b) Input graph G_0 including translations



(c) Desired output graph G_i, new arcs dotted

Figure 3.3: Excerpt of input and desired output graph

(Fellbaum and Vossen, 2007), only contains very limited sets of concepts for English, Spanish, and Catalan, as of December 2010.

Recently, Navigli and Ponzetto (2010) presented a multilingual knowledge base called BabelNet. The main contribution is an alignment of WordNet with Wikipedia, which is discussed in Chapter 5. Additionally, they also propose using an existing machine translation system to translate a sense-annotated collection of English sentences to other languages. For each English word sense, the most frequent translation is then linked to the synset. While sense-annotated input sentences are not readily available in large quantities, this approach is interesting because the machine translation system only makes use of the local context of a word in text for disambiguation. Using the Google Translate API, the reported precision is 72%. Hence, this approach is likely to provide a valuable complementary signal to the graph-based features we consider in our approach.

Automatic Construction. A central problem in establishing wordnets is the laborious manual compilation process, which typically leads to insufficient coverage for practical applications. Several authors have attempted to automatically or semi-automatically construct a wordnet for a not yet covered language using existing wordnets (Okumura and Hovy, 1994; Atserias et al., 1997; Daudé et al., 2000; Pianta et al., 2002; Sathapornrungkij and Pluempitiwiriyawej, 2005; Fišer, 2008). Scannell (2003) followed a similar approach to translate Roget's Thesaurus to Irish Gaelic.

Our approach adopts some of the basic intuitions of these studies, but goes beyond simple heuristics by computing more sophisticated features that can account for very subtle differences between correct and incorrect `means` arcs, and then learning a model to make the final prediction. In de Melo and Weikum (2008b), we showed that our machine learning strategy leads to an output of higher quality and better recall than previous work. Many of the prior approaches experienced difficulties with polysemous terms and were applied to nouns only, while our technique works particularly well for commonly used polysemous terms. Isahara et al. (2008) attempted to use multiple existing wordnets to combine information from multiple translation dictionaries, however with precision scores of 54% at best. None of these previous

studies have explored the ideas of letting automatically established mappings for different languages reinforce each other or of exploiting evidence from multilingual translation graphs. Finally, none of the previous approaches have been applied to the task of building a large-scale multilingual wordnet.

Multilingual Lexical Knowledge Bases. There are not many other approaches to building multilingual lexical databases automatically. The PANGLOSS ontology (Knight and Luk, 1994) was created in the 1990s to facilitate machine translation. Interesting linking heuristics were used; however no learning techniques were employed, and the final coverage was limited to around 70,000 entities in two languages.

Cook (2008) created a semantic network that incorporates WordNet and links nouns in three languages to WordNet nodes based on simple heuristics as well as manual work. The heuristics yield high-quality results but apply to monosemous nouns only and hence fail to account for most commonly used words, as these tend to be polysemous.

A much larger lexical resource has been created by Etzioni et al. (2007) and Mausam et al. (2009), who use translation dictionaries and Wiktionary to create a very large translation graph, which is then exploited for cross-lingual image search. Their central aim, however, is to derive a translation resource rather than constructing a semantic network with terms and semantic entities equipped with additional relations like hypernymy (or `subclass`), meronymy (or `partOf`), etc.

Michelbacher et al. (2010) used graph-based techniques to generate a multilingual thesaurus that for a given term provides semantically related terms in a second language. Such resources do not differentiate between specific semantic relations or offer a taxonomy.

Finally, knowledge bases like YAGO (Suchanek et al., 2007) and DBpedia (Auer et al., 2007), while drawing on Wikipedia's interwiki links to provide multilingual entity labels, do not possess a multilingual upper-level ontology.

3.3 Initial Graph Construction

On our way towards producing a multilingual knowledge base, we work with multiple graphs of the form described in Chapter 2. The initial

input graph G_0 will be the result of an extraction and synthesis of data from existing sources, while further graphs G_i ($i \geq 1$) constructed later on will extend G_0 with statistically derived information that eventually yields the multilingual UWN graph.

3.3.1 Information Extraction and Acquisition

The initial graph $G_0 = (V, A_0, \Sigma)$ will contain nodes V representing terms and semantic entities, and arcs A_0 representing simple lexical relationships (with arc labels in Σ). To populate G_0 with such nodes and arcs, we draw on a range of different knowledge sources. Existing lexical knowledge bases and translation sources are the essential ones here, together with a small set of manually classified arcs that are required for the learning. The other knowledge sources are optional. This means that, apart from the `means` and other semantic arcs taken from inputs like the English WordNet, most of the imported information will consist of `translation` arcs (or `synonym` arcs).

Existing Wordnet Instances. To bootstrap the construction, we rely on existing wordnets to provide term-to-meaning `means` arcs for a limited set of languages, as well as meaning-to-meaning arcs. Since relations like hypernymy, meronymy and so on apply to entire synsets rather than just individual words in WordNet, we treat them as implying relations between semantic entities (`subclass`, `partOf`, and so on).

Apart from Princeton WordNet 3.0, `means` information is also taken from the Arabic (Rodríguez et al., 2008), Catalan (Benitez et al., 1998), Estonian (Orav and Vider, 2005), Hebrew (Ordan and Wintner, 2007), and Spanish (Atserias et al., 2004a) wordnets, as well as from the human-verified parts of MLSN (Cook, 2008). These resources all use entity identifiers compatible with Princeton WordNet, however many of them are aligned with older versions of WordNet, so we apply mappings between different WordNet versions (Daudé et al., 2003) to obtain canonical entity identifiers for semantic entities.

The arcs that we create mostly have a weight of 1, except in some isolated cases where the mappings between different WordNet versions had a lower weight. Sense rank information and sense frequency information based on the sense-annotated SemCor corpus (Fellbaum, 1998) is

incorporated as an annotation into `means` arc labels as specified earlier in Section 2.4. Such sense frequency information reveals to us how often for example the word '*school*' was used to refer to a school building in the corpus.

Translation Dictionaries. A considerable number of `translation` arcs between two terms are imported from over 100 open-source translation dictionaries that are freely available on the Web[1]. As only few of these resources consist of well-structured markup (like XML), making their content amenable to machine processing frequently requires custom preprocessing steps. These involve separating the actual terms from annotation information such as part-of-speech (e.g. `adverb`), semantic domain (e.g. `chemistry`), etc. We treat translation information as many-to-many relationships between words, adding source or target part-of-speech labels to the translation arcs whenever they are given. The arcs are assigned a weight of 1.

Wiktionary. The community-maintained Wiktionary project[2] offers a plethora of lexical information but relies on simple text-based mark-up rather than an explicit, precise database schema. We thus use rule-based information extraction techniques to mine `translation` and other arcs from eight different language-specific editions of Wiktionary (Catalan, English, French, German, Greek, Portuguese, Spanish, and Swedish).

Multilingual Thesauri and Ontologies. Translations are also obtained from concept-oriented resources such as the GEneral Multilingual Environmental Thesaurus (GEMET[3]), OmegaWiki[4], as well as from OWL ontologies (Buitelaar et al., 2004). For each semantic entity (concept) x, we consider its set of natural language labels $\mathrm{terms}(x)$ in the resource,

[1]Sources: 5Lingue Table, Apertium, CEDICT, dict-fef, DictionaryForMIDs, Ding, Ding Spanish-German, English-Hungarian dictionary (Egyeki Gergely), ER-Dict, es-ita Dictionary, FreeDict, GIDIC, Greek-English UTF8 Dictionary, HanDeDict, Heinzelnisse Norwegian-German, Laws Maori-English, Magic-Dic, Sdict English-Thai, Sdict Ukrainian-English, Slovnyk English-Russian, Termcat Terminologia Oberta, trasvasesno Spanish-German, XDXF English-Armenian, XDXF German-Russian

[2]`http://www.wiktionary.org`

[3]`http://www.eionet.europa.eu/gemet/`

[4]`http://www.omegawiki.org`

and then add a `translation` arc to the graph for each $t, t' \in \text{terms}(x)$ $(t \neq t')$, unless they are from the same language, in which case we create a `synonym` arc instead.

Parallel Corpora. Text from conventional multilingual corpora, translation memories, film subtitles, and software localization files can be word-aligned to harness additional translation information for many language pairs. We make use of GIZA++ (Och and Ney, 2003) and Uplug (Tiedemann, 2003) to produce lexical alignments for a subset of the OPUS corpora (Tiedemann, 2004), which includes the OpenSubtitles corpus (Tiedemann, 2007). Since word alignments tend to be unreliable, we compile alignment statistics and add `translation` arcs to the graph between pairs of nodes where the respective term pair is encountered with a high frequency (above a specified threshold).

Monolingual Thesauri. Monolingual thesauri from the OpenOffice software distribution[5] provide `related` arcs between the terms of a single language, revealing e.g. that '*college*' is semantically related to '*university*'.

Manually Classified Arcs. As our approach is based on supervised learning, we also depend on a limited amount of manually classified `means` arcs from terms to semantic entities, obtained via a collaborative Web contribution interface (see also Figure 5.6 on page 157). Such arcs are either labelled as positive (correct, adequate, with weight 1) or negative (incorrect, inadequate, with weight 0). Details are given in Section 3.5.3.

3.3.2 Graph Enrichment and Pruning

After the initial information extraction, we apply additional preprocessing methods to the input graph.

Inverse Links

First of all, we assume the `translation` and `synonym` relations are symmetric and add inverse links to ensure that all connections are reciprocal.

[5] `http://wiki.services.openoffice.org/wiki/Dictionaries`

Criterion 3.1 (Symmetry). Given an arc $a = (n_1, n_2, r, w) \in A_0$ where r is a `translation` or `synonym` label, we add (n_2, n_1, r', w) to A_0 if no comparable arc already exists, where r' matches r except for inverted source and target lexical categories.

Triangulation

Additionally, although the two relations are not genuinely transitive, we use so-called triangulation heuristics to reduce the sparsity of translations and synonymy links. For instance, when the Italian word '*scuola*' has an English translation '*school*' and a French translation '*école*', and the latter two both have a Malay translation '*sekolah*', then we can infer that this Malay word is also a likely translation for the Italian term.

Criterion 3.2 (Triangulation). Translation (or synonymy) arcs (n_1, n_2, r, w) between two term nodes n_1, n_2 are added to A_0 if

$$|\{n' | n' \in \Gamma_{\mathrm{o}}(n_1, A_0) \cap \Gamma_{\mathrm{i}}(n_2, A_0)\}| \geq m_{\min}$$

and no comparable arc already exists.

Here, Γ_{o} and Γ_{i} refer to the out- and in-neighbourhood, respectively (Definition 2.2). We empirically chose $m_{\min} = 5$ for high accuracy.

(Near-)Duplicate Merging

Subsequently, the graph is pruned by merging duplicate and near-duplicate arcs as follows. It is clear that duplicate arcs from different sources can be merged, but additionally there can also be arcs between two nodes that are nearly the same, except for some of the additional information captured as part of the arc label (see Section 2.4). Hence, we define a partial ordering $\leq_\Sigma$ over arc labels that captures when a label is considered less specific than (or as specific as) another one.

Definition 3.1 *Given two arc labels $r, r' \in \Sigma$, we define the partial ordering $\leq_\Sigma$ over Σ as follows: $r \leq_\Sigma r'$ if and only if r and r' express the same relation (e.g. `translation`), and the additional information captured in r is in all cases less specific than (i.e., lexical category `unknown`, synset rank 0, or synset frequency 1) or just as specific as the corresponding information in r'.*

The rationale for defining a partial ordering is that when we have two near-duplicate arcs, it often makes sense to keep only the more precise one and assume that the other one simply lacks certain detail. For instance, when we have two translation arcs, one without lexical category information (`unknown`) and one with such information (e.g. `noun`), we will keep only the latter, although, of course, it might still be the case that the translation applies with respect to other lexical categories like `verb` as well.

In practice, this assumption means that we iterate over all arcs $a = (n_1, n_2, r, w) \in A_0$, discarding a according to the following criterion.

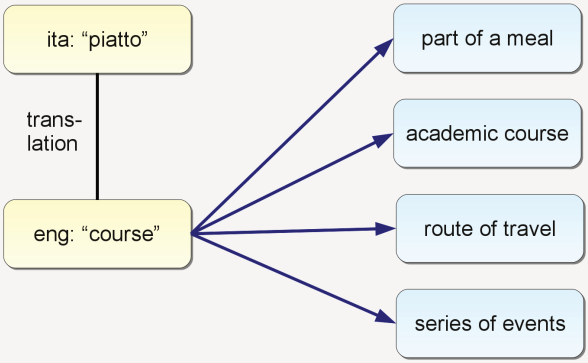
Criterion 3.3 (Pruning). An arc $a = (n_1, n_2, r, w) \in A_0$ is pruned from A_0 whenever there exists another arc $a' = (n_1, n_2, r', w') \in A_0$ with $a \neq a'$, $r \leq_\Sigma r'$, $w \leq w'$.

3.3.3 Candidate Arc Creation

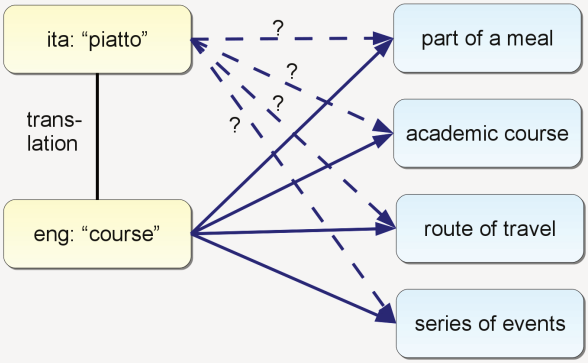
As a final preprocessing step that concludes the construction of G_0, we create a large set of zero-weighted arcs that denote *potential* `means` relationships between words and semantic entities that will later be evaluated. In Figure 3.3b, we see that the Italian word '*piatto*' has a translation arc to '*course*', so it is likely that they share some meaning. All four semantic nodes linked from '*course*' in G_0 correspond to potential meanings of the word '*piatto*'. At this point, we do not know which ones are correct, but we can create four *candidate arcs* to express these potential `means` relationships between the word '*piatto*' and the respective semantic entities, as illustrated in Figure 3.4. In general, we consider as candidate entities all meanings of translations or synonyms of a given term. We iterate over the graph and define the set of possible candidate arcs according to the following definition.

Definition 3.2 (Candidate Arcs) *The set of candidate arcs A_{cand} consists of all possible $(n_0, n_2, r_{\text{m}}, w)$ such that*

1. *n_0 is a term node,*
2. *n_2 is a semantic node,*
3. *the arc label $r_{\text{m}} = (\texttt{means}, 0, 1)$,*
4. *$w \in [0, 1]$,*



(a) Original Input



(b) Input with candidate arcs

Figure 3.4: Candidate arc creation – Terms are linked to meanings of their translations.

5. *there exists a 2-hop path of the form*

$$\{(n_0, n_1, r, w_1), (n_1, n_2, r', w_2)\} \subset A_0,$$

where the arc label r is a `translation` *or* `synonym` *one, and the arc label r' is a* `means` *one,*

6. *and no arc $(n_0, n_2, r, w') \in A_0$ already exists with a* `means` *label r.*

Each candidate arc $(n_0, n_2, r_{\mathrm{m}}, w)$ links a term n_0 to one of its potential meanings n_2. While A_{cand} includes all possible instantiations with different weights $w \in [0, 1]$, what we actually add to A_0 at this point is the set

$$\{(n_0, n_2, r_{\mathrm{m}}, w) \in A_{\mathrm{cand}} \mid w = 0\}.$$

This means that the arcs are initially set up with a weight of 0. Later on, we iterate over all candidate arcs in the graph and establish more appropriate weights. In Figure 3.4, the arc to the semantic node described as '*part of a meal*' should later receive a higher weight, as it is an adequate meaning for '*piatto*', while the other semantic entities, e.g. `academic course` are inadequate, and the corresponding candidate arcs should no longer be present in the final output graph.

3.4 Iterative Graph Refinement

In each iteration, a new graph $G_i = (V, A_i, \Sigma)$ is constructed that is topologically identical to G_{i-1} and thus to G_0. However, the weights of all candidate `means` arcs in the graph are re-assessed to reflect a refined measure of confidence in them being correct.

3.4.1 Scoring Model

Overview. To this end, our approach is to learn a statistical model for assessing the validity of candidate arcs. In each iteration i, we employ a supervised regression model w_i, obtained by training on the small set of hand-labelled arcs included in G_0, which are labelled either as correct (positive training samples) or incorrect (negative training samples). For a given candidate arc, the model predicts a weight in $[0, 1]$ that represents the degree of confidence in the respective arc being correct, given the previous graph G_{i-1}.

Feature Space. The regression model operates with respect to an appropriately defined feature space. In our approach, the feature space is recomputed with each new graph G_i of the refinement process. This is in the spirit of relaxation labelling methods and belief propagation methods for graphical models (Getoor and Taskar, 2007). Directly applying

such algorithms to the huge input graph in our task would face tremendous scalability problems, since we need to capture non-straightforward dependencies between outputs for different arcs even when they are multiple hops apart. Instead, we embed information about the neighbourhood of an arc into its feature vector. In the ideal case, the weight of an arc, given its feature vector, will then be conditionally independent of the weights of other arcs, allowing us to use a wide range of standard learning algorithms. In each iteration i, the previous graph G_{i-1} is used as the basis to derive a feature vector $\mathbf{x}_i(n_0, n_2) \in \mathbb{R}^m$ for each candidate arc (n_0, n_2, r, w) in G_i (where m is the number of features). Details will be given in Section 3.4.2.

Model. Using the feature vectors for the hand-labelled training set, we train an RBF-kernel support vector machine (SVM). SVMs are based on the idea of computing a separating hyperplane that maximizes the margin between positive and negative training instances in the feature space or in a high-dimensional kernel space (Vapnik, 1995; Cortes and Vapnik, 1995; Duda et al., 2000).

For each feature vector $\mathbf{x}_i(n_0, n_2)$, standard regularized SVM classification initially yields values $f(\mathbf{x}_i(n_0, n_2)) \in \mathbb{R}$ that correspond to distances from the separating hyperplane in the kernel space. To obtain new weights w for candidate arcs $(n_0, n_2, r, w') \in A_i \cap A_{\text{cand}}$, we adopt Platt's method of estimating posterior probabilities $P(w = 1|\mathbf{x}_i(n_0, n_2))$ using a sigmoid function. This means that w is set to an estimate of the posterior probability computed as

$$w_i(n_0, n_2) = \frac{1}{1 + \exp(a_i f(\mathbf{x}_i(n_0, n_2)) + b_i)},$$

where parameter fitting for a_i and b_i is performed using maximum likelihood estimation on the training data of iteration i (Platt, 2000; Lin et al., 2007).

This regression model allows us to obtain new arc weights $w = w_i(n_0, n_2) \in [0, 1]$ for all candidate arcs from term nodes n_0 to semantic nodes n_2. G_i can be constructed as (V, A_i, Σ) where

$$A_i = (A_{i-1} \backslash A_{\text{cand}}) \cup \{(n_0, n_2, r, w_i(n_0, n_2)) \mid (n_0, n_2, r, w') \in A_{i-1} \cap A_{\text{cand}}\}$$

and A_{cand} is the set of possible candidate arcs (Definition 3.2).

3.4.2 Feature Computation

Feature Vectors. The regression model determines the new arc weights based on the feature vectors $\mathbf{x}_i(n_0, n_2)$. These vectors need to provide some sort of evidence that would indicate whether a given arc is correct. For each candidate `means` arc (n_0, n_2, r, w) in G_i, we quantify evidence from the graph as an m-tuple of numerical feature scores

$$\mathbf{x}_i(n_0, n_2) = (x_{i,1}(n_0, n_2), \ldots, x_{i,m}(n_0, n_2)) \in \mathbb{R}^m,$$

to allow the learning algorithm to assess whether the arc should be accepted. We expect to see strong evidence for this arc if n_2, a semantic node, represents one of the meanings of the term designated by n_0.

Given the previous graph G_{i-1}, the individual scores $x_{i,j}(n_0, n_2)$ are computed as listed in Table 3.1. In Equation 3.1, two nodes are directly compared by means of a cosine-based context similarity score, which will be explained in Section 3.4.5.

Semantic Overlap. The underlying idea for Equations 3.2 and 3.3 (where ϕ_1, ϕ_2, γ are arc and path weighting functions) is that a word's most likely meanings can be determined by considering likely meanings n_2' of its translations and related terms $n_1 \in \Gamma_{\mathrm{o}}(n_0, A_{i-1})$. Equation 3.2 considers each successor node n_1, and then assesses how similar the successors of n_1 are to n_2.

For instance, in the simplest case, if we use an identity test as a similarity function for comparing those successors n_2' to n_2, then this score effectively computes a weighted count of the number of two-hop paths from n_0 to n_2. In the input graph in Figure 3.3b, there are multiple paths from the German word '*Kurs*' to the `academic course` semantic node, which means that it is more likely to represent a correct meaning.

With more sophisticated similarity measures, we can also take into consideration when there are multiple successors with distinct yet similar meanings, e.g. one translation could have an academic course meaning and another could refer to a group of students who are taught together. In this case, these two very similar meanings are more likely to be correct than meanings that are completely unrelated to the meanings of other translations.

Table 3.1: Feature computation formulae

$$x_{i,j}(n_0, n_2) = \quad \mathrm{sim}(n_0, n_2) \tag{3.1}$$

$$x_{i,j}(n_0, n_2) = \sum_{n_1 \in \Gamma_o(n_0, A_{i-1})} \phi_1(n_0, n_1)\, \mathrm{sim}^*_{n_0,\phi_2}(n_1, n_2) \tag{3.2}$$

$$x_{i,j}(n_0, n_2) = \sum_{n_1 \in \Gamma_o(n_0, A_{i-1})} \phi_1(n_0, n_1)\, \frac{\mathrm{sim}^*_{n_0,\phi_2}(n_1, n_2)}{\mathrm{sim}^*_{n_0,\phi_2}(n_1, n_2) + \mathrm{dissim}^+_{n_0,\phi_2}(n_1, n_2)} \tag{3.3}$$

where

$$\mathrm{sim}^*_{n_0,\phi_2}(n_1, n_2) = \max_{n_2' \in \Gamma_o(n_1, A_{i-1})} \gamma(n_0, n_1, n_2')\, \phi_2(n_1, n_2')\, \mathrm{sim}(n_2, n_2')$$

(maximum weighted similarity between n_2 and successors of n_1) (3.4)

$$\mathrm{dissim}^+_{n_0,\phi_2}(n_1, n_2) = \sum_{n_2' \in \Gamma_o(n_1, A_{i-1})} \gamma(n_0, n_1, n_2')\, \phi_2(n_1, n_2')(1 - \mathrm{sim}(n_2, n_2'))$$

(weighted sum of dissimilarities between n_2 and successors of n_1) (3.5)

Polysemy. Equation 3.3 is similar to Equation 3.2, but adds an additional normalization with respect to the number of alternative choices in the denominator. In the simplest case, the dissim^+ function will simply count how many alternative semantic entities there are, so if the term represented by n_1 has one meaning corresponding to n_2, and 4 other meanings, it would return 4, and lead to a summand of $\frac{1}{1+4}$ for n_1, which reflects the probability of arriving at n_2 from n_1 when randomly selecting `means` arcs. Equation 3.3 is also applied in the opposite direction to quantify reachability information from a semantic node to a term node.

Weighted Scores. More sophisticated scores are obtained by applying additional weighting and normalization. This is addressed by having the scores depend on a number of auxiliary formulae, in particular combinations of arc weighting functions ϕ_1, ϕ_2, as described in Section 3.4.3, path weighting functions γ, described in Section 3.4.4, and measures of semantic relatedness, which will be described in Section 3.4.5.

For example, in Equation 3.3 we may wish to not count *all* alternative meanings, instead producing a weighted score where alternative meanings are not fully considered if they are very similar or if their lexical category tags do not match.

3.4.3 Arc Weighting Functions

Arcs from Terms to Terms. Not all translations or related terms for a word are equally important. The different versions of ϕ_1 listed in Table 3.2 estimate the relevance of a connection from a term n_0 to a translation, synonym or related term n_1.

- Equation 3.6 simply filters out `related` arcs, as these are less reliable than `translation` and `synonym` ones. This weighting function is combined with the other instantiations of ϕ_1. Within the formula, V_{T} is the set of all term nodes.
- Equation 3.7 normalizes with respect to the size of the out-neighbourhood of n_0, counting the number of terms that have outgoing `means` arcs (arcs to nodes in V_{S}, the set of all semantic nodes in the graph). This can lead to more comparable scores across different terms n_0, as some terms have significantly more translations than others.
- Equation 3.8 is similar to Equation 3.3 but normalizes with respect to a weighted in-degree of n_1 for terms from the same language. Essentially, it checks how many terms connected to the term n_1 are from the same language as n_0. If there are few or no alternatives, then the connection between the two nodes is expected to be stronger.
- Equation 3.9 defines the helper function $\phi_1^{\ln}(n_0, n_0')$ that is used to check if two term nodes have the same language.

Table 3.2: Arc weighting functions plugged into the formulae in Table 3.1

Filtering

$$\phi_1^{\mathrm{f}}(n_0, n_1) = \begin{cases} 1 & \exists (n_0, n_1, r, w) \in A_{i-1} : r \neq \texttt{related}, n_1 \in V_{\mathrm{T}} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

Normalization

$$\phi_1^{\mathrm{nm}}(n_0, n_1) = \frac{\phi_1^{\mathrm{f}}(n_0, n_1)}{|\{n_1 \in \Gamma_{\mathrm{o}}(n_0, A_{i-1}) \mid \Gamma_{\mathrm{o}}(n_1, A_{i-1}) \cap V_{\mathrm{S}} \neq \emptyset\}|} \tag{3.7}$$

Weighted In-Degree

$$\phi_1^{\mathrm{bt}}(n_0, n_1) = \phi_1^{\mathrm{f}}(n_0, n_1) \frac{\mathrm{sim}^*_{n_0, \phi_1^{\mathrm{ln}}}(n_1, n_0)}{\mathrm{sim}^*_{n_0, \phi_1^{\mathrm{ln}}}(n_1, n_0) + \mathrm{dissim}^+_{n_0, \phi_1^{\mathrm{ln}}}(n_1, n_0)} \tag{3.8}$$

Language Matching

$$\phi_1^{\mathrm{ln}}(n_0, n_0') = \begin{cases} 1 & n_0 = (t, l),\ n_0' = (t', l) \text{ share language } l \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Arc Weights

$$\phi_2^{\mathrm{t}\alpha}(n_1, n_2) = \begin{cases} 1 & \exists (n_1, n_2, r, w) \in A_{i-1} : w > \alpha, n_2 \in V_{\mathrm{S}} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

Corpus Frequencies

$$\phi_2^{\mathrm{cf}}(n_1, n_2) = \phi_2^{\mathrm{t}\alpha}(n_1, n_2) \frac{\mathrm{freq}(n_1, n_2)}{\sum\limits_{n_2' \in \Gamma_{\mathrm{o}}(n_1, A_{i-1})} \phi_3^{\mathrm{slc}}(n_2, n_2')\, \mathrm{freq}(n_1, n_2')} \tag{3.11}$$

Sense Rank

$$\phi_2^{\mathrm{r}}(n_1, n_2) = \begin{cases} 0 & \phi_2^{\mathrm{t}\alpha}(n_1, n_2) = 0 \\ \dfrac{1}{\mathrm{rank}(n_1, n_2) + \frac{1}{2}} & \mathrm{rank}(n_1, n_2) \neq 0 \\ \dfrac{1}{|\{n_2' \in \Gamma_{\mathrm{o}}(n_1, A_{i-1})\} \cap V_{\mathrm{S}}|} & \text{otherwise} \end{cases} \tag{3.12}$$

Semantic Node Lexical Category

$$\phi_3^{\mathrm{slc}}(n_2, n_2') = \begin{cases} 1 & n_2 = (s, c),\ n_2' = (s', c) \text{ share category } c \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

Arcs from Terms to Semantic Entities. Instantiations of ϕ_2 estimate the relevance of connections from translations, synonyms or related terms n_1 to semantic nodes n_2.

- Equation 3.10 considers the weights of `means` arcs, allowing us to ignore unreliable candidate arcs.
- Equation 3.11 uses sense-specific corpus frequencies, where the function $\mathrm{freq}(n_1, n_2)$ yields the frequency of term n_1 with sense n_2 in the SemCor corpus or 1 if n_1 does not occur in the corpus. This information was embedded into the arc labels in Section 3.3.1. The helper function ϕ_3^{slc} (Equation 3.13) compares the part-of-speech of semantic nodes.
- Equation 3.12 uses the sense ranking, where $\mathrm{rank}(n_1, n_2)$ gives the WordNet-specific rank of n_2 for term n_1 (1 for the first sense, 2 for the second, and so on, or 0 if unavailable). The ranks reflect the importance assigned to different senses of a word by WordNet's editors.

3.4.4 Cross-Lingual Lexical Category Heuristics

Motivation. Several features described in Table 3.1 integrate a function γ that assigns weights to entire paths in the graph. Apart from the trivial choice of setting it to a constant value ($\gamma^{id}(n_0, n_1, n_2) = 1$), we use γ^{lc} as a version that considers lexical categories (part-of-speech tags) associated with nodes in the graph. For instance, we may have a path from a German noun like '*Schule*' to the English translation '*school*', and then on to a verbal meaning of '*school*' (to school someone). This path should have a very low weight if we are sure that the German word '*Schule*' can only be a noun. Many of the previous studies on automatically building wordnets dealt with nouns exclusively, whereas our approach handles all lexical categories. Hence the need for some means of preventing a noun from being mapped to a verbal or adjectival meaning, for example.

Path Scores. We accomplish this by relying on different types of hints provided by the graph, in conjunction with machine learning to possibly detect the part-of-speech of words for which no hints are available. The path weighting scores $\gamma^{lc}(n_0, \ldots, n_k)$ are supposed to provide an

estimate of whether the nodes along the path from n_0 to n_k have the same or at least compatible lexical categories.

Definition 3.3 *We define*

$$\gamma^{lc}(n_0, \ldots, n_k) = \max_{c \in C} \prod_{i=1}^{k-1} \mu_c(n_i, n_{i+1})$$

where $\mu_c(n_i, n_{i+1}) \in [0, 1]$ *provides an estimate of whether a local transition from* n_i *to* n_{i+1} *is possible with lexical category* $c \in C$*, computed as*

$$\mu_c(n, n') = \begin{cases} 1 & \exists (n, n', r, w) \in A_0 : r, c \textit{ compatible} \\ \min(\mu_c(n), \mu_c(n')) & \textit{otherwise.} \end{cases}$$

Here, $\mu_c(n) \in [0, 1]$ *estimates the probability of a node* n *having lexical category* c *among one of its possible lexical categories.*

In the formula for $\mu_c(n_i, n_{i+1})$, we first check whether a `translation` or `synonym` arc from n_i to n_{i+1} exists that provides explicit lexical category information as part of the arc label. We explained earlier in Sections 3.3 and 3.3.1 that some dictionaries and other sources provide such information. If this is the case, we just need to match it with the lexical category c under consideration.

Node Scores. If the arcs do not provide a clear answer, we compare *possible* categories of individual nodes n_i and n_{i+1}, relying on estimates $\mu_c(n) \in [0, 1]$. The estimates depend on the type of node, and are computed using Algorithm 1. The algorithm performs the following steps.

1. If the node n is in V_S, the set of semantic nodes, we can simply check the lexical category encoded in the entity identifer (see Section 2.3).
2. For term nodes, we check if the term has *any* incoming or outgoing `translation` or `synonym` arc labelled with lexical categories, or *any* `means` arc to a semantic node. If there are labelled arcs, but none of them are labelled with c or with `unknown`, then $\mu_c(n) = 0$.

Algorithm 1: Lexical category compatibility $\mu_c(n)$ for nodes

procedure NODELC($n, c, G = (V, A, \Sigma)$)
  if $n \in V_S$ **then** ▷ if n is a semantic node
    return $\begin{cases} 1 & \exists s : n = (s, c) \\ 0 & \text{otherwise} \end{cases}$ ▷ check c encoded in node label
  $R \leftarrow \{\texttt{translation}, \texttt{synonym}\}$
  $C_1 \leftarrow \{c_1 \mid \exists (n, n', r, w) \in A, l \in R, c_2 : r = (l, c_1, c_2)\}$
    ▷ set of source lexical categories captured in arc labels
  $C_2 \leftarrow \{c_2 \mid \exists (n', n, r, w) \in A, l \in R; c_1 : r = (l, c_1, c_2)\}$
    ▷ set of target lexical categories captured in arc labels
  $C_3 \leftarrow \{c_3 \mid \exists (n, n', r, w) \in A, s : r$ is a `means` arc, $n' = (s, c_3)\}$
    ▷ set of lexical categories captured in semantic nodes n'
  if $c \in (C_1 \cup C_2 \cup C_3)$ **then**
    return 1
  else if $|C_1 \cup C_2 \cup C_3| > 0$ **and** `unknown` $\notin (C_1 \cup C_2 \cup C_3)$ **then**
    return 0
  else if $w_c(n)$ available **and** $w_c(n)$ reliable **then**
    return $w_c(n)$
  else
    return 0.5

3. If this fails, we attempt to use learnt models for surface properties of term strings, which often reveal likely lexical categories. For each lexical category and language, we check whether the above criteria provide us with sufficient examples to create a training set and a withheld validation set of part-of-speech labelled terms. The validation set is a separate labelled set that is disjoint from the training set and can be used to assess how well the trained model applies to new, unseen terms. If we have enough information to create both labelled sets, we learn surface form properties as described below.
4. If none of the aforementioned steps apply, a default score of 0.5 may be used, which means that we assume the chance of a compatible lexical category to be 50%.

Surface Form Learning. The surface form learning for term nodes is carried out by growing C4.5 decision trees (Quinlan, 1993; Duda et al., 2000) with the following features:

1. Prefixes and suffixes of a word up to a length of 10 (without case conversion): In many languages, affixes mark the part-of-speech tag of a word. For instance, in Italian, lemma forms of virtually all verbs end in '*-are*', '*-ere*', or '*-ire*'.
2. Boolean features for first character capitalization and complete capitalization: In many languages, capitalized words tend to be nouns (e.g. acronyms such as '*USA*', proper nouns like '*London*', all nouns in German, Luxemburgish).
3. Term length: In some languages, nouns tend to be longer than verbs, for example.

The reliability of the decision tree depends largely on the language. For each lexical category and language, we evaluated on the respective validation set, obtaining F_1-scores between 0.03 and 0.99 (see Section 3.5.2 for an introduction to such evaluation metrics). Later on, for a given term to be analysed, the confidence estimate $w_c(n)$ from the decision tree's leaves is considered reliable in the following cases:

1. the F_1-score on the validation set was high
2. $w_c(n) > 0.5$ and the precision on the validation set was high
3. $w_c(n) < 0.5$ and the recall on the validation set was high

3.4.5 Measures of Semantic Relatedness

Motivation. The feature vector computation also uses a set of different semantic relatedness measures. To see the potential benefit of this, consider the following example. The single meaning of '*schoolhouse*' is related to the educational institution meaning of the word '*school*', but not to the meaning of '*school*' that corresponds to groups of fish. So, if a term node has translation arcs to both '*school*' and '*schoolhouse*', their semantic relatedness tells us that the educational meanings of '*school*' are much more likely to be correct than the one referring to fish. We consider four different measures of semantic relatedness.

Identity Relatedness

The first weighting function $\mathrm{sim}_{\mathrm{id}}(n_1, n_2)$ is the trivial identity indicator function.

Definition 3.4 *The weighting function* $\mathrm{sim}_{\mathrm{id}}(n_1, n_2)$ *is computed as*

$$\mathrm{sim}_{\mathrm{id}}(n_1, n_2) = \begin{cases} 1 & n_1 = n_2 \\ 0 & \textit{otherwise.} \end{cases}$$

Neighbourhood Relatedness

A more sophisticated weighting function $\mathrm{sim}_{\mathrm{n}}(n_1, n_2)$ considers the neighbourhood in the graph. For a given path in the graph, we compute a proximity score multiplicatively from relation-specific arc weights $w(a) \in [0, 1]$ obtained by optimizing application-specific scores using a local search procedure (de Melo and Siersdorfer, 2007), e.g. 0.8 for hypernymy, 0.7 for holonymy, and so on. The similarity is then defined to be the maximum score for all simple paths $p \in \mathrm{P}(n_1, n_2, A_0)$ between n_1 and n_2 in A_0 if this maximum is above or equal a pre-defined threshold $\alpha_{\mathrm{n}} = 0.35$, and 0 otherwise. In practice, such scores can be computed efficiently using an adaptation of Dijkstra's shortest-path algorithm (de Melo and Siersdorfer, 2007).

Definition 3.5 *The weighting function* $\mathrm{sim}_{\mathrm{n}}(n_1, n_2)$ *is defined as*

$$\mathrm{sim}'_{\mathrm{n}}(n_1, n_2) = \max_{p \in \mathrm{P}(n_1, n_2, A_0)} \prod_{a \in p} w(a)$$

$$\mathrm{sim}_{\mathrm{n}}(n_1, n_2) = \begin{cases} \mathrm{sim}'_{\mathrm{n}}(n_1, n_2) & \mathrm{sim}'_{\mathrm{n}}(n_1, n_2) \geq \alpha_{\mathrm{n}} \\ 0 & \textit{otherwise.} \end{cases}$$

Here, $\mathrm{P}(n_1, n_2, A_0)$ *denotes the set of paths between* n_1 *and* n_2 *in* A_0*,* $w(a)$ *yields a relation-specific weight for an arc* a *in some path* p*, and* α_{n} *is a threshold.*

Contextual Relatedness

Another weighting function $\mathrm{sim}_{\mathrm{c}}(n_1, n_2)$ uses the cosine similarity of context strings for nodes. For semantic entities, context strings are

constructed by concatenating English meaning descriptions (WordNet glosses) and terms linked to the original semantic entity and neighbouring semantic entities. For terms, the set of all English translations is used. Two context strings are compared by stemming using Porter's method, creating TF-IDF vectors $\mathbf{x}_1$, $\mathbf{x}_2$, and computing the cosine of the angle between them.

Definition 3.6 *The weighting function* $\mathrm{sim_c}(n_1, n_2)$ *is defined as*

$$\mathrm{sim_c}(n_1, n_2) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{||\mathbf{x}_1||_2 \ ||\mathbf{x}_2||_2}$$

for TF-IDF vectors $\mathbf{x}_1$, $\mathbf{x}_2$ *corresponding to* n_1, n_2.

Maximum Relatedness

The final measure of semantic relatedness $\mathrm{sim_m}(n_1, n_2)$ combines the power of $\mathrm{sim_n}$, and $\mathrm{sim_c}$, which are each based on rather different characteristics of the semantic entities.

Definition 3.7 *The weighting function* $\mathrm{sim_m}(n_1, n_2)$ *is computed as*

$$\mathrm{sim_m}(n_1, n_2) = \max\{\mathrm{sim_n}(n_1, n_2), \mathrm{sim_c}(n_1, n_2)\}.$$

3.4.6 Overall Algorithm

Iterative Refinements. Algorithm 2 describes the overall algorithm. Our learning procedure not only makes use of the small set of manually classified `means` arcs supplied as training instances, but also benefits from the enormous numbers of originally unlabelled instances by running in multiple iterations. In each iteration i, a model w_i is learnt using feature scores computed on the output graph G_{i-1} of the previous iteration, as described earlier. There is frequently some form of mutual reinforcement between correct and highly weighted (but not known to be correct) arcs and there is some gradual down-weighting of incorrect arcs in the course of the iterations. Thus, our method can be seen as a form of semi-supervised learning.

As a stopping criterion, we use either a withheld validation set of manually classified arcs (not used for training) or apply cross-validation with the training data, and check if a loss function $L(G_i)$ (such as 1

Algorithm 2: Lexical integration

procedure ADDTERMS($G_0 = (V_0, A_0, \Sigma), L, \epsilon, i_{\max}, L', w_{\min}, \hat{w}_{\min}$)
  $i \leftarrow 0$
  repeat
    $i \leftarrow i + 1$ ▷ current iteration number
    learn model w_i given G_{i-1} ▷ as described in Section 3.4.1
    $A_i \leftarrow \{(n_0, n_2, r, w_i(n_0, n_2)) \mid (n_0, n_2, r, w') \in A_{i-1} \cap A_{\text{cand}}\}$
    ▷ re-evaluate candidate arcs with w_i
    $A_i \leftarrow A_i \cup (A_{i-1} \setminus A_{\text{cand}})$ ▷ other arcs unchanged
    $G_i \leftarrow (V, A_i, \Sigma)$
  until $L(G_{i-1}) - L(G_i) < \epsilon$ or $i = i_{\max}$
  $i^* \leftarrow \arg\min_i L'(G_i)$ ▷ determine best iteration
  $A \leftarrow \{(n, n', r, w) \in A_0 \mid r$ not a `means` label$\}$ ▷ semantic relations
  $A_{i^*} \leftarrow A_{i^*} \cap A_{\text{cand}}$ ▷ retain only candidate arcs
  for all $(n_0, n_2, r, w) \in A_{i^*}$ **do**
    if $w > w_{\min}$ **then**
      $A \leftarrow A \cup \{(n_0, n_2, r, w)\}$ ▷ threshold $w_{\min}$
    else if $w > \hat{w}_{\min} \wedge \neg\exists n_2', r', w' : (n_0, n_2', r', w') \in A_{i^*}, w' > w$ **then**
      $A \leftarrow A \cup \{(n_0, n_2, r, w)\}$ ▷ threshold $\hat{w}_{\min}$
  $V \leftarrow \{n \in V \mid \Gamma_{\text{o}}(n, A) \cup \Gamma_{\text{i}}(n, A) \neq \emptyset\}$ ▷ prune node set
  return $G = (V, A, \Sigma)$

minus F_1, cf. Section 3.5.3) shows a reduction $L(G_{i-1}) - L(G_i) < \epsilon$ (where ϵ may also be slightly negative). The number of iterations can optionally be limited by setting the $i_{\max}$ parameter. In practice, we observed that 2-4 iterations suffice to stabilize the precision and recall measures on the graph.

Output Graph. Having determined the most profitable iteration i^* with a loss function L' (possibly different from L), Algorithm 2 then proceeds to transform G_{i^*} into the final UWN graph G. This involves the following steps:

(i) We add to G language-independent semantic relationships from Princeton WordNet (see Section 3.5.5 for details), but none of the `means` arcs from the original input sources.

(ii) For *candidate* `means` arcs, we threshold by enforcing a minimal weight $w_{\min}$ or possibly a slightly lower minimal weight $\hat{w}_{\min}$

in the absence of better alternative arcs for a node n_0. This allows us to obtain a lexical database that retains only high-quality links. It is, however, possible to set $w_{\min}$,$\hat{w}_{\min}$ to a value like $-\infty$, which allows us to obtain a statistical form of lexical database with edge weights providing the degree of confidence of a statement. Weighted edges can be useful in certain application settings.

(iii) Finally, we remove all nodes of degree 0.

Our specific choices of loss functions and thresholds are given in the following section on experimental results.

3.5 Results

3.5.1 System Architecture

We used the Java programming language to develop a platform-independent knowledge base processing framework. For efficiency reasons, the weighted labelled multi-digraphs were stored in custom binary format databases, where we could encode arc labels and weights very compactly. Both the index and the actual data are cached in memory to the extent possible, to reduce the level of disk access. In some cases, we also relied on Bloom filtering to probabilistically avoid unnecessary disk reads when no target nodes are available for a given pair (n, r) of source node n and arc label r.

This framework allowed us to flexibly plug together information extraction modules (as required in Section 3.3.1), knowledge base processors (as used for preprocessing in Section 3.3.2 and for the iterative arc reweighting in Section 3.4), as well as exporters and analysis modules to form knowledge base processing pipelines. Our graph refinement procedure is integrated as a mapper that assesses links between two entities and produces new weights. For statistical learning, it relies on the LIBSVM implementation (Chang and Lin, 2001) using an RBF kernel $K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{1}{m}(||\mathbf{x} - \mathbf{y}||_2)^2)$ where m is the number of features.

The main bottleneck of the iterative graph refinement is the storage access to lookup direct and indirect neighbours of a node in the knowledge base, as required to compute the feature scores. In each of up to $i_{\max}$ iterations, up to $O(|V_0|^2)$ candidate arcs are evaluated. The feature computation in Equations 3.2 and 3.3 is based on $O(|V_0|^2)$ 2-hop paths,

and weighting functions and similarity measures can require additional arc lookups. Fortunately, in practice, terms have a limited outdegree, i.e. a limited number of `translation`, `synonym` and `means` arcs, so the graph is extremely sparse.

Additionally, since the model is only updated once per iteration i and embeds neighbourhood properties in G_{i-1} into a given feature vector, each of the large number of candidate arc assessments made within an iteration can be made independently. Hence, parallelizing the mapping process is trivial.

3.5.2 Evaluation Metrics

For evaluating the results, we rely on the standard metrics *precision* and *recall*. These two measures have their roots in document retrieval. Recall scores reveal what fraction of all correct (relevant, desired) items is in the set of items selected by the system. This is also known as the sensitivity. Precision scores, in contrast, indicate what fraction of all items selected by the system are actually correct items. This is also known as the positive predictive value. In document retrieval, the items are documents or publications, and the correct ones are the ones that users consider relevant for a given query. In our context, the items we wish to assess are `means` arcs from term nodes to semantic nodes and the correct ones are the ones where the semantic node indeed reflects one of the senses of the term designated by the respective term node.

These scores can also be expressed in a slightly different terminology. Correct items are often called *positives*, and incorrect items are accordingly called *negatives*. If a system tells us an item is a positive, then it can either be a true positive (assessed as correct by the system, and indeed correct) or a false positive (assessed as correct by the system, but not really correct), and similarly for negatives.

Definition 3.8 *If P_{T}, P_{F}, N_{T}, N_{F} are the sets of true positives, false positives, true negatives, and false negatives, respectively, then precision*

p and recall r can be defined as follows:

$$p = \begin{cases} \dfrac{|P_{\mathrm{T}}|}{|P_{\mathrm{T}} \cup P_{\mathrm{F}}|} & P_{\mathrm{T}} \cup P_{\mathrm{F}} \neq \emptyset \\ 1 & \textit{otherwise} \end{cases} \tag{3.14}$$

$$r = \begin{cases} \dfrac{|P_{\mathrm{T}}|}{|P_{\mathrm{T}} \cup N_{\mathrm{F}}|} & P_{\mathrm{T}} \cup N_{\mathrm{F}} \neq \emptyset \\ 1 & \textit{otherwise} \end{cases} \tag{3.15}$$

Neither precision nor recall is necessarily very useful on its own. A trivial classifier that accepts all items has a perfect recall of 1, while a classifier that does not accept any items obtains an optimal precision of 1. In order to compare different evaluation results, one of several composite measures can be used, for instance the break-even point of precision and recall (criticized in Sebastiani, 2002, p.36), or the F_β-measure (van Rijsbergen, 1979, ch.7).

Definition 3.9 *If p denotes the precision score, and r denotes the recall score, then the F_β-measure is computed as*

$$F_\beta = \begin{cases} \dfrac{(\beta^2 + 1)pr}{\beta^2 p + r} & \beta p + r \neq 0 \\ 0 & \textit{otherwise} \end{cases} \tag{3.16}$$

for $\beta \geq 0$, where $\beta = 2$ for example would imply that precision is weighted twice as much as recall.

Most commonly, $\beta = 1$ is selected, which leads to the well-known F_1-measure (also known as F-score) that is equivalent to the harmonic mean of precision and recall.

$$F_1 = \begin{cases} \dfrac{2pr}{p + r} & p + r \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

Finally, we often speak about the *accuracy* of a resource. In relation to the terminology used above, the accuracy could be computed as

$$\frac{|P_{\mathrm{T}} \cup N_{\mathrm{T}}|}{|P_{\mathrm{T}} \cup P_{\mathrm{F}} \cup N_{\mathrm{T}} \cup N_{\mathrm{F}}|},$$

i.e. the fraction of all items that were correctly assessed. However, when evaluating a sample of a resource, all items involved have already been accepted by the system, so N_{T} and N_{F} are 0, and the accuracy is equivalent to the precision.

3.5.3 Dataset

Initial Graph. Following Section 3.3, G_0 was constructed with:

- 448,069 existing `means` arcs (from the input wordnets, mainly English, Spanish, Catalan),
- 10,805,400 `translation` arcs (from the dictionaries, Wiktionary, thesauri, and parallel corpora),
- 10,343,601 candidate `means` arcs (generated following Section 3.3.3, on average 7.7 per term node).

It contained roughly 129,500 semantic nodes and 1.3 million term nodes with candidate arcs (5 million overall).

Training Set. We added 2,445 human-classified `means` arcs for training, out of which 610 were positive, 1,835 were negative examples. The training set was compiled by manual annotation of candidate `means` arcs as either positive or negative for randomly selected French and German terms, rather than for randomly selected arcs. This means that the risk of overfitting is reduced and the learner is channelled to focus explicitly on the distinction between negative and positive examples for a given word rather than coincidental differences between different words.

Validation Set. We additionally used a validation set of 2,901 candidate `means` arcs for French and German terms, manually annotated as positive or negative using the same methodology, and selected 1 minus F_1 scores with respect to this validation set on the output graph for $w_{\min} = 0.6$, $\hat{w}_{\min} = 0.5$ as the loss function. A perfect F_1 score (and zero loss) would be obtained if all correct candidate `means` arcs in this validation set (i.e. those manually assessed as positive) are accepted when applying these thresholds, and none of the incorrect `means` arcs in the validation set (those manually assessed as negative) are accepted.

Table 3.3: Iterations of algorithm with validation set scores (for $w_{\min} = 0.7$, $\hat{w}_{\min} = 0.6$)

Graph	Precision	Recall	F_1	# Accepted `means` Arcs
G_0	N/A	0.00%	0.00%	0
G_1	83.96%	67.42%	74.79%	1,540,206
G_2	83.70%	68.48%	75.33%	1,594,652
G_3	83.89%	68.64%	75.50%	1,595,763
G_4	83.90%	67.88%	75.04%	1,573,395

Table 3.4: Precision of UWN result graph

Dataset	Sample Size	Precision (Wilson)
French	311	$89.23\% \pm 3.39\%$
German	321	$85.86\% \pm 3.76\%$
Mandarin Chinese	300	$90.48\% \pm 3.26\%$

3.5.4 Results for Means Arcs

Algorithm. The algorithm ran for four iterations until it failed to improve the F_1-score on the validation set, as shown in Table 3.3, taking multiple days to complete on a single machine. The input graph G_0 does not cover any of the validation arcs, and thus has a recall and F_1-score of 0%. English is the most widely represented language within the input graph, both with respect to the input wordnets and for the translations, so the first iteration provided for the most significant gains and already delivered excellent results. In the next iteration, G_1 served as the input graph, leading to an improved F_1-score for G_2 because a larger range of translation terms are equipped with non-zero `means` arcs in G_1 compared to G_0. These improvements decrease very quickly, since the additional amount of information available to the feature computation process, compared to previous iterations, keeps diminishing.

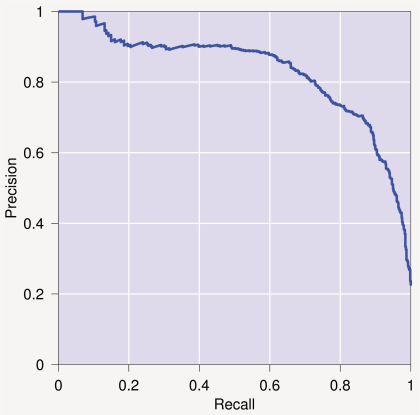


Figure 3.5: Precision-recall curve on validation set for G_3 when $w_{\min} = \hat{w}_{\min}$

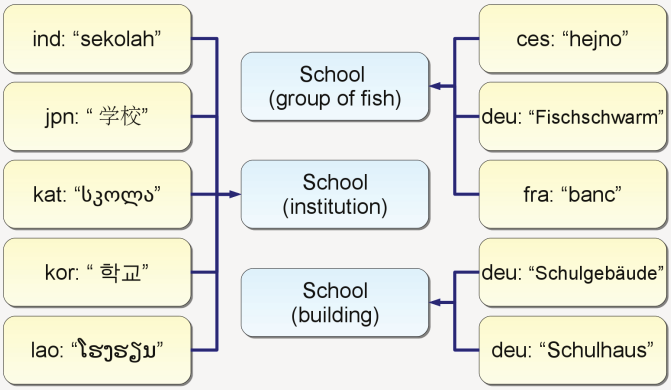


Figure 3.6: Excerpt from UWN graph with `means` arcs from terms to three semantic nodes

Precision-Recall Tradeoff. At this point, we have the choice of preferring high precision, e.g. G_2 has 91.59% precision at 44.55% recall for $w_{\min} = 0.9$, $\hat{w}_{\min} = 0.75$, or high recall, e.g. G_3 gives us 73.92% precision at 80.30% recall for $w_{\min} = 0.3$, $\hat{w}_{\min} = 0.25$. Our loss function balances precision and recall, making G_3 the most profitable graph. Figure 3.5 shows the tradeoff between precision and recall on G_3. For the final UWN output graph, we chose $w_{\min} = 0.6$, $\hat{w}_{\min} = 0.5$ as it provided good coverage at a reasonable precision.

Assessment. Figure 3.6 provides an excerpt from this graph, highlighting how words in different languages have been disambiguated and linked to appropriate semantic entities constituting meanings of the English word '*school*', e.g., in French, the term '*banc*' is used to refer to a school of fish. We recruited human annotators for French, German, and Mandarin Chinese, which were asked to evaluate randomly chosen arcs in the respective language from the output graph. We rely on Wilson score intervals at $\alpha = 0.05$ (Brown et al., 2001) to generalize our findings in a statistically significant manner, as listed in Table 3.4. These randomly chosen arcs are not related to the training or validation sets, which moreover did not contain any Mandarin Chinese terms, so the results show that a surprisingly high level of precision can be obtained even cross-lingually.

It must be pointed out that it is not possible to reliably evaluate the accuracy of a wordnet using pre-existing wordnets, as they do not fulfil the closed world assumption, i.e. a `means` arc not occurring in an existing wordnet does not warrant the conclusion that the link is false. This is particularly true for current non-English wordnets, which often have limited coverage and semantic inventories based on older versions of WordNet.

Examples of false positives include for instance the German word '*Schuljahr*', which was correctly linked to a semantic node representing academic years, but also linked to a semantic node for '*schoolday*', which is not correct. Cases of false negatives include the German word '*Schule*', which was correctly linked to semantic nodes for '*school*' in the sense of educational institutions and groups of artists or thinkers, among others, but was not linked to the semantic node for the temporal meaning of '*school*' or '*schoolday*', as in '*stay after school*'.

Table 3.5: Coverage of final UWN graph with respect to accepted candidate `means` arcs as well as terms.

	`means` Arcs	Distinct Terms
Overall	1,595,763	822,212
By Language		
German	132,523	67,087
French	75,544	33,423
Esperanto	71,247	33,664
Dutch	68,792	30,154
Spanish	68,445	32,143
Turkish	67,641	31,553
Czech	59,268	33,067
Russian	57,929	26,293
Portuguese	55,569	23,499
Italian	52,008	24,974
Hungarian	46,492	28,324
Thai	44,523	30,815
Others	795,782	427,216
By Lexical Category		
Nouns	1,048,003	589,536
Verbs	221,916	88,189
Adjectives	289,328	147,257
Adverbs	36,095	26,254

Coverage. Table 3.5 shows the coverage of the output graph. Bearing in mind that the final UWN graph retains only *candidate* `means` arcs, these figures do not include any `means` arcs imported from the input wordnets, and only count term nodes that are connected to semantic nodes via these new candidate `means` arcs. There are terms in more than 200 languages in UWN.

The most well-represented languages result quite directly from the selection of translations in the input graph G_0. We found that terms with translations to many languages had high chances of being included. Our approach thus successfully addresses a long-standing problem in

automatic construction of wordnets, namely that of insufficient coverage of commonly used words, which tend to be more polysemous. Using sophisticated features, it carefully benefits from cross-lingual evidence to find meanings of such terms, while previous approaches had trouble coping with the polysemy of commonly used words.

The terms in UWN have `means` links to a total of 80,620 distinct semantic nodes. Of course, when pursuing this lexical integration strategy, lexical gaps and incongruences are a problem, i.e. we do not cover language-specific concepts that are not represented in the original inventory of semantic entities. For instance, the German word '*Feierabend*' means the finishing time of the daily working hours, which is not represented in WordNet. We address this problem in Chapters 4 and 5.

The break-down by part-of-speech shows that the majority of terms are nouns. Table 3.6 provides average degrees with respect to `means` arcs for term nodes (out-degree) and semantic nodes (in-degree), revealing the level of polysemy of terms according to UWN. The middle column shows average out-degrees when term nodes with only one `means` arc are excluded.

3.5.5 Results for Semantic Relations

Cross-Lingual Transfer. We further evaluated to what extent relationships given by Princeton WordNet apply to UWN. Some might contend that using a taxonomy based on one set of languages cannot serve as a structural basis that does justice to the organization of another language's lexicon. We believe that this is mainly an issue of accounting for lexical gaps. As discussed earlier in Section 2.3.3, if in one language birds and insects are considered animals and in another they are not, then there are actually distinct concepts of animals that need to be distinguished. Thus, such issues can be addressed by adding new semantic entities and integrating them into the taxonomy, as will be discussed in Chapters 4 and 5, respectively.

Our working assumption is that relations between two semantic entities, e.g. WordNet's hypernymy, at an abstract level apply independently of the language of the terms associated with them. This is similar to the assumption made in WordNet that many relations apply at the

Table 3.6: Average degree with respect to `means` arcs of term nodes (out-degree) and semantic nodes (in-degree)

	Term Node Out-Degree	Term Node Out-Degree Excluding Monosemous	Semantic Node In-Degree (Multilingual)
Nouns	1.78	3.20	12.76
Verbs	2.52	4.24	16.12
Adjectives	1.96	3.63	15.19
Adverbs	1.37	2.53	9.97
Total	1.94	3.38	13.56

abstract synset level, independently of which particular synonyms of that synset are considered.

Assessment. For several types of relations, randomly selected links between two semantic entities were assessed, where both semantic entities have associated German language terms (linked via `means` arcs). Table 3.7 shows that the overall precision is high. Incorrect relationships resulted almost entirely from incorrect `means` arcs.

In addition to relations between synsets, WordNet also provides relations between specific *words* (with respect to meanings of those words). Such relations cannot be transferred directly to UWN, since it is not known to which pairs of involved terms of the corresponding synsets they apply. However, in some cases, we can infer from them more generic relationships between semantic entities. For instance, when WordNet tells us that the word '*scholastic*' is derivationally related to the word '*school*', we can interpret this as a generic indicator of semantic relatedness between semantic entities. Antonymy relationships between words such as '*good*' and '*bad*' are re-interpreted as a generic form of semantic opposition between semantic entities (`opposite` relation). These, too, were evaluated in Table 3.7.

Table 3.7: Quality assessment for imported relations

Relation	Precision (Wilson Interval)	Sample Size
hypernymy	$87.1\% \pm 4.8\%$	182
instance	$88.8\% \pm 4.6\%$	174
similarity	$92.0\% \pm 3.8\%$	181
category	$93.3\% \pm 4.5\%$	100
meronymy (part-of)	$94.4\% \pm 4.1\%$	102
meronymy (member-of)	$92.7\% \pm 4.0\%$	145
meronymy (substance-of)	$95.6\% \pm 3.5\%$	108
antonymy (as sense opposition)	$94.3\% \pm 3.9\%$	117
derivation (as semantic similarity)	$94.5\% \pm 4.0\%$	104

Gloss Descriptions. UWN also includes `hasGloss` links connecting semantic entities with their English language glosses from WordNet. These are textual descriptions that define or explain the meanings intended to be associated with semantic entities. The English glosses generally also apply to non-English terms attached to the semantic entities, which is an important asset also in NLP applications, as shown later on in Section 3.5.8. Obtaining non-English gloss descriptions is non-trivial. One solution we investigated is using sophisticated machine translation systems (de Melo and Weikum, 2010b). The quality varies greatly and depends on the machine translation system and the respective language pair. Fortunately, even syntactically incorrect translations can suffice as contextual information for word sense disambiguation, for example. Another solution is to rely on additional knowledge sources providing glosses in many different languages, a strategy we pursue in later chapters.

3.5.6 Alternative Inventory of Semantic Entities

In an additional set of experiments, we evaluated how generic our approach is by testing it on an alternative inventory of semantic entities. Roget's Thesaurus, first published by Peter Mark Roget in 1852, is the most well-known thesaurus in the English-speaking world (Hüllen, 2004). The thesaurus has been used as a lexical knowledge base in several different tasks, including word sense disambiguation (Yarowsky, 1992) and analysis of textual cohesion (Morris and Hirst, 1991).

We took advantage of the work by Cassidy (2000), who made the American 1911 edition of Roget's Thesaurus (Mawson, 1911) available in digital form with minor extensions. Although this version is provided as a plaintext file, parsing it in order to obtain a lexical database required considerable additional effort. We relied on a recursive top-down approach to identify the top-level divisions and various sorts of subdivisions, all the way down to the level of headwords. Under each headword, one finds one or more part-of-speech markers followed by groups of terms or phrases relating to the headword, as displayed in Figure 3.7. These groups, delimited by semicolons or full stops, can be treated as reasonably fine-grained semantic entities, reflecting much finer distinctions than the very general headwords. For instance, the three terms '*withdraw*', '*take from*', '*take away*' in Figure 3.7 would form a single semantic entity.

In our case study, we investigated attaching French term entities to these semantic entities. We used translation information derived from English-French translation dictionaries, amounting to a total of 78,000 `translation` arcs and a coverage of around 34,000 English terms and 48,000 French terms. We created a training dataset of 731 candidate arcs between term entities and semantic entities. The random test set consists of 1,012 labelled arcs of this form. For additional details, please refer to de Melo and Weikum (2008c).

With these inputs, we then built a French version of Roget's Thesaurus in a single iteration. Tables 3.8 and 3.9 give the results without and with the French OpenOffice thesaurus as background knowledge, respectively, for several choices of $w_{\min}$ and $\hat{w}_{\min}$. The results are more than satisfactory, given the difficulty of such disambiguation tasks, and demonstrate the viability of our approach despite our designation of

```
     #38. Nonaddition. Subtraction. -- N. subtraction, subduction|!;
deduction, retrenchment; removal, withdrawal; ablation, sublation[obs3];
abstraction &c. (taking) 789; garbling,, &c. v. mutilation,
detruncation[obs3]; amputation; abscission, excision, recision; curtailment
&c. 201; minuend, subtrahend; decrease &c. 36; abrasion.
     V. subduct, subtract; deduct, deduce; bate, retrench; remove,
withdraw, take from, take away; detract.
     garble, mutilate, amputate, detruncate[obs3]; cut off, cut away, cut
out; abscind[obs3], excise; pare, thin, prune, decimate; abrade, scrape,
file; geld, castrate; eliminate.
     diminish &c. 36; curtail &c. (shorten) 201; deprive of &c. (take) 789;
weaken.
     Adj. subtracted &c. v.; subtractive.
     Adv. in deduction &c. n.; less; short of; minus, without, except,
except for, excepting, with the exception of, barring, save, exclusive of,
save and except, with a reservation; not counting, if one doesn't count.
```

Figure 3.7: Excerpt from Roget's Thesaurus text file.

Table 3.8: Evaluation of Roget's Thesaurus translation for different choices of classification thresholds

$w_{\min}$	$\hat{w}_{\min *}$	Precision	Recall
0.3	0.25	84.05%	77.75%
0.35	0.3	85.80%	75.50%
0.5	0.5	89.49%	66.00%
0.6	0.5	89.38%	61.00%

semicolon groups as the semantic entities, which requires much finer distinctions than would be necessary at the level of headwords. The coverage for $w_{\min} = 0.3$, $\hat{w}_{\min} = 0.25$, with OpenOffice.org information is given in Table 3.10. Figure 3.8 shows an excerpt from the generated French thesaurus. Note how polysemy can lead to mistranslations (translating the English '*deduction*' to '*ratiocination*' may make sense in certain contexts, however in this case a different sense of '*deduction*' was intended). Similar translations have been generated in several other languages, and are freely available for download at `http://www.mpi-inf.mpg.de/~gdemelo/mtrogets/`.

Table 3.9: Evaluation of Roget's Thesaurus translation (with additional background information from the OpenOffice.org thesaurus)

$p_{\min}$	$p_{\min *}$	Precision	Recall
0.3	0.25	84.94%	81.75%
0.35	0.3	87.64%	78.00%
0.5	0.5	89.40%	67.50%
0.6	0.5	91.01%	63.25%

Table 3.10: Coverage statistics for translation of Roget's Thesaurus with additional background knowledge

	terms	lexicalized nodes	node mappings
nouns	11,161	11,628	31,376
verbs	3,624	4,861	14,666
adjectives	6,166	5,418	15,116
adverbs	705	651	1,638
total	21,232	22,560	62,798

3.5.7 Thesaurus Generation from WordNet

UWN itself can also be regarded as a multilingual thesaurus, however with very fine-grained semantic distinctions. In a separate study, we showed how parallel corpora can be used to obtain example sentences for specific meanings of words (de Melo and Weikum, 2009a). With such meaning-specific examples, users are more easily able to grasp the differences between different uses of a word.

Often, however, users are simply looking for words that are somewhat related, without any particular interest in subtle differences in meaning. To produce a more conventional associative thesaurus where words that are loosely related are listed together, we can rely on a simple recursive graph exploration. Algorithm 3 looks up all semantic entities

```
    #38. N. soustraction; prélèvement, déduction, ratiocination;
enlèvement, abaissement, élimination, déménagement, suppression, mise
à pied, retranchement, réduction; prélèvement, retraite, ablation,
retrait, claustration; inattention, idée abstraite, abstraction;
mutilation; amputation, exérèse; réduction, restriction, abaissement,
diminution, raccourcissement; réduction, abaissement; frottement,
abrasion, éraflure;
    V. retrancher, soustraire, déduire, prélever, rabattre, décompter;
déduire, diminuer, inférer, rabattre; réduire, enlever, restreindre,
prélever, tirer, ôter, retirer, éloigner, emporter; retrancher,
enlever, prélever, ôter, éloigner, emporter, emmener; enlever, amputer,
mutiler, estropier, altérer, fausser; abattre, retrancher, tailler;
découper; élaguer, tailler, éplucher, rogner, diluer, pruneau,
s'effiler, exciser; éroder, gratter, user, utiliser, décimer, racler;
castrer, châtrer; castrer, châtrer, anglaiser; évacuer; rabattre,
soustraire; retrancher, réduire, restreindre, écourter, raccoucir;
débiliter;
    Adj. soustrait, soustraites, soustraite, soustraits;
    Adv. moins; excepté, hormis, sauf, en outre, moins, hors, dénué de,
à l'exception de;
```

Figure 3.8: Excerpt from translation of Roget's Thesaurus text file.

for a term as well as certain related semantic entities, and then forms the union of all lexicalizations of these entities. Table 3.11 provides example output of the algorithm with settings $l_{\mathrm{p}} = 2$, $l_{\mathrm{c}} = 2$, $l_{\mathrm{g}} = 1$ on UWN's output graph. For the German language, the thesaurus contains a total of 67,087 terms, each entry listing 31 additional related terms on average.

3.5.8 Semantic Relatedness

Task. We studied semantic relatedness assessment as an application of UWN in conjunction with Princeton WordNet's semantic relations and descriptions. The objective is to automatically estimate the degree of relatedness between two words, producing scores that correlate well with the average ratings by human evaluators. For instance, most humans rate '*curriculum*' as much more closely related to a word like '*school*' than to a word like '*water*'. Such relatedness assessments are useful for a number of different tasks in information retrieval and text mining. Making the assessments automatically is an active research area, e.g.

Table 3.11: Sample entries from generated German thesaurus

headword: Akademiker
Absolvent, Adressat, Akademie, Akademikerin, Assistenz-Professor, Assistenz-Professorin, außerordentlicher Professor, außerordentliche Professorin, Begünstigter, Buchgelehrte, Buchgelehrter, Empfänger, Erzieher, Erzieherin, Gastprofessor, Gastprofessorin, Geist, Gymnasium, Hochschule, Hochschullehrer, Hochschullehrerin, Intellektuelle, Intellektueller, Lehrer, Lehrerin, Lehrstuhlinhaber, Lehrstuhlinhaberin, ordentlicher Professor, Professor, Professorin, Pädagoge, Pädagogin, Rezipient, Schüler, Stubengelehrte, Stubengelehrter, Student akademisch, intellektuell
headword: lernbegierig
Fleiß, Stubengelehrsamkeit achtsam, angewandt, beflissen, behutsam, eifrig, emsig, fleißig, geflissentlich, gelehrt, sorgfältig, sorgsam, wissenschaftlich

Table 3.12: Evaluation of semantic relatedness measures, using Pearson's sample correlation coefficient (r)

Dataset	GUR65		GUR350		ZG222	
	r	Cov.	r	Cov.	r	Cov.
Inter-Annotator Agreement	0.81	(65)	0.69	(350)	0.49	(222)
Wikipedia (ESA)	0.56	65	0.52	333	0.32	205
GermaNet (Lin)	0.73	60	0.50	208	0.08	88
UWN ($\mathrm{sim_n}$)	0.77	60	0.62	242	0.43	106
UWN ($\mathrm{sim_c}$)	0.77	60	0.68	242	0.52	106
UWN ($\mathrm{sim_m}$)	0.80	60	0.68	242	0.51	106

Algorithm 3: Thesaurus generation

Input: a lexical knowledge base $G = (V, A, \Sigma)$, number of parent class levels l_p, number of child class levels l_c, number of levels for other general relations l_g, set of acceptable general relations R, global set of all of term nodes V_T
Objective: generate a thesaurus that lists related terms for any given term

procedure GENERATETHESAURUS($G = (V, A, \Sigma), R, V_\mathrm{T}$)
  for each term t in $V \cap V_\mathrm{T}$ **do**
    $T \leftarrow \emptyset$ ▷ the list of related terms for t
    for each semantic node $n \in \Gamma_\mathrm{o}(t, A) \setminus V_\mathrm{T}$ **do**
      for each node $n' \in$ RELATED($G, n, l_\mathrm{p}, l_\mathrm{c}, l_\mathrm{g}, R$) **do**
        $T \leftarrow T \cup (\Gamma_\mathrm{i}(n', A) \cap V_\mathrm{T})$ ▷ add terms of n' to T
    output T as list of related terms for t

function RELATED($G = (V, A, \Sigma), n, l_\mathrm{p}, l_\mathrm{c}, l_\mathrm{g}, R$)
  $S \leftarrow \{n\}$
  for each node $n' \in \Gamma_\mathrm{o}(n, A) \setminus V_\mathrm{T}$ **do** ▷ recursively visit related nodes
    if (n `subclass` of n') $\wedge$ ($l_\mathrm{p} > 0$) **then**
      $S \leftarrow S \cup$ RELATED($G, n', l_\mathrm{p} - 1, 0, 0, \emptyset$)
    else if (n' `subclass` of n) $\wedge$ ($l_\mathrm{c} > 0$) **then**
      $S \leftarrow S \cup$ RELATED($G, n', 0, l_\mathrm{c} - 1, 0, \emptyset$)
    else if ($\exists (n, n', r, w) \in A : r \in R$) $\wedge$ ($l_\mathrm{g} > 0$) **then**
      $S \leftarrow S \cup$ RELATED($G, n', 0, 0, l_\mathrm{g} - 1, R$)
  return S

Resnik (1995) has been cited more than a thousand times. Many of the well-known techniques rely on a lexical database like WordNet.

Approach. In Section 3.4.5, we described measures of semantic relatedness between semantic entities. If we are instead given two term nodes t_1, t_2, we can estimate their relatedness as

$$\mathrm{rel}(t_1, t_2) = \max_{n_1 \in \Gamma_\mathrm{o}(t_1, A)} \max_{n_2 \in \Gamma_\mathrm{o}(t_2, A)} w(t_1, n_1) w(t_2, n_2) \mathrm{sim}(n_1, n_2)$$

using the measures from Section 3.4.5 and $w(t, n)$ denoting the `means` arc weight from t to n (or 0 if none).

Results. Three German-language datasets (Gurevych, 2005; Zesch and Gurevych, 2006) that capture the arithmetic mean of relatedness assessments made by human judges serve as our ground truth. For instance, German words for '*jewel*' and '*gem*' were assessed as highly related (98.5%), German words for '*mountain*' and '*coast*' were rated as somewhat related (42.7%), and German words for '*glass*' and '*magician*' were judged as barely related (14.6%). Such arithmetic means were compared with assessments made by our methods using Pearson's sample correlation coefficient (also known as the product-moment correlation coefficient).

In Table 3.12, the first row lists the inter-annotator agreement between different human evaluators and the number of term pairs rated for each dataset. The following rows show the results for our three semantic relatedness measures on the UWN graph, as well as scores for two alternative measures as reported by Gurevych et al. (2007): the state-of-the-art explicit semantic analysis (ESA) method by Gabrilovich and Markovitch (2007) on Wikipedia, and a more traditional method based on GermaNet, the manually compiled German wordnet.

The results suggest that UWN can be more useful than hand-crafted resources, with respect to both the correlation with human judgments and the coverage (the number of term pairs from the dataset where both terms are found in the respective lexical database). Another advantage of our approach is that it may also be applied without any further changes to the task of cross-lingually assessing the relatedness of terms in different languages.

3.5.9 Cross-Lingual Text Classification and Vector Spaces

Text Classification. Another applied task we considered was cross-lingual text classification. Text classification is the task of assigning text documents to the classes or categories considered most appropriate, thereby e.g. topically distinguishing texts about thermodynamics from others dealing with quantum mechanics. This is commonly achieved by representing each document using a vector in a high-dimensional feature space where each feature accounts for the occurrence of a particular term from the document set (a bag-of-words model), and then applying

Table 3.13: Cross-lingual text classification results

	Precision	Recall	F_1
English-Italian			
Terms only	69.90%	66.81%	68.32%
Terms and meanings	83.24%	70.49%	76.34%
English-Russian			
Terms only	57.86%	46.67%	51.66%
Terms and meanings	67.87%	74.94%	71.23%
Italian-English			
Terms only	71.97%	77.06%	74.43%
Terms and meanings	76.59%	79.67%	78.10%
Italian-Russian			
Terms only	59.65%	57.15%	58.37%
Terms and meanings	68.03%	79.26%	73.21%
Russian-English			
Terms only	68.36%	66.34%	67.34%
Terms and meanings	73.56%	80.29%	76.78%
Russian-Italian			
Terms only	67.85%	57.48%	62.24%
Terms and meanings	71.38%	72.21%	71.79%

machine learning techniques such as support vector machines. For more information, please refer to the survey by Sebastiani (2002).

Cross-Lingual Text Classification. Cross-lingual text classification is an extremely challenging variant, where the documents to be classified are in a language distinct from the language of the labelled training documents. Since documents from two different languages obviously have completely different term distributions, the conventional bag-of-words text representations perform poorly. Instead, it is necessary to induce representations that tend to give two documents from different languages similar representations when their semantic content is similar.

One means of achieving this is the use of language-independent conceptual feature vector spaces where feature dimensions represent meanings of terms rather than just the original terms. In our experiments, the set of terms in a document d is determined by tokenizing the document text. Text is preprocessed by removing stop words and performing part-of-speech tagging as well as lemmatization using the TreeTagger (Schmid, 1994). We attempt to recognize multi-word expressions by maintaining an n-gram window (limited to $n \leq 3$ for practical reasons) and performing lookups in WordNet or UWN to see whether matching multi-word expressions exist. In addition to capturing the original term frequencies for each term, the feature space is augmented by mapping each term to the respective semantic nodes listed by Princeton WordNet (for English words) or UWN (for other languages). We embrace a rather simple approach that foregoes disambiguation: For every occurrence of a term t (after preprocessing), we take all semantic nodes n_{m} with a matching part-of-speech tag, and normalize their weights by dividing by the sum of their `means` arc weights. Thus, if a term has four equally relevant semantic nodes in UWN, then each receives a local weight of $\frac{1}{4}$. Additionally, all original semantic nodes for a term pass on their weight to neighbouring nodes immediately connected via `subclass` arcs. Summing up the weights of local occurrences of a token t (either an original document term or a semantic entity identifier) within a document d, one arrives at document-level occurrence scores $n(t, d)$, from which one can then compute TF-IDF feature vectors using the following formula:

$$f(t,d) = \begin{cases} n(t,d) \log \frac{|D|}{|\{d \in D \mid n(t,d) \geq 1\}|} & \{d \in D \mid n(t,d) \geq 1\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

where D is the set of training documents.

Experiments. This approach was tested using a cross-lingual dataset derived from the Reuters RCV1 and RCV2 collections of newswire articles (Reuters, 2000a,b). The articles are mostly business related, and have topical class labels like '*accounts/earnings*', '*economic performance*', and '*funding/capital*'. For several pairs of languages, we created independent datasets by randomly selecting 10 topics covered by both lan-

guages in order to arrive at $\binom{10}{2} = 45$ separate binary classification tasks per language pair, each based on 150 training documents in one language, and 150 test documents in a second language, likewise randomly selected with balanced class distributions.

Each dataset was evaluated independently, once using only the standard bag-of-words TF-IDF representation for terms (only genuine term frequencies as $n(t, d)$ in Equation 3.18), and once with the extended representation that includes mappings to semantic entities as frequencies. Table 3.13 provides the results. The scores shown were produced with linear kernel SVMs using the SVMlight implementation in its default settings, which are known to work well for text classification (Joachims, 1999) – LIBSVM produced similar margins between the two approaches but overall slightly lower absolute scores. Since many of the Reuters topic categories are business-related, using only the original document terms, which include names of companies and people, already works surprisingly well in some cases, despite the different languages. By considering semantic entities, both precision and recall are boosted significantly. This means that the vectors of documents are more similar when their topical content is similar, despite the fact that the original documents are in different languages. Hence, these experiments show, for instance, that English terms in the training set are being mapped to the same semantic entities as the corresponding Russian terms in the test documents. The margins could be boosted even further by invoking more intelligent word sense disambiguation strategies or using more advanced semantic expansion strategies (de Melo and Siersdorfer, 2007).

Vector Space Representations. The vector space in our cross-lingual text classification experiments consists of vectors that describe documents based on the terms or semantic entities they contain. Semantic vector representations of this sort can also be used in tasks like information retrieval (Salton and McGill, 1986), text similarity assessment, and document clustering.

A different kind of vector representation can be constructed for terms themselves, where individual vector space dimensions initially represent co-occurrence with specific other terms in the text, and a singular value decomposition (Schütze, 1992) or second-order co-occurrences can be computed to create more stable representations (Schütze, 1998). As in

the case of document vectors, we can adapt these approaches to rely on co-occurrences with semantic entities, leading to a multilingual vector space where words in different languages can be compared, even if they do not themselves occur in UWN.

3.6 Discussion

In this chapter, we have presented a novel machine learning approach for building large-scale multilingual lexical knowledge bases. Statistical models are applied in multiple iterations to a graph in order to assess and disambiguate meanings of terms. The resulting resource, called UWN, contains 1.5 million `means` relationships for over 800,000 terms in over 200 languages, making it the largest multilingual version of WordNet. UWN is available at `http://www.mpi-inf.mpg.de/yago-naga/uwn/`.

Our experiments have shown that UWN is useful in applied tasks. In addition to the existing applications of WordNet, such as human consultation (de Melo and Weikum, 2010b), question answering, query expansion, text classification, semantic relatedness assessment, and so on, which are now possible for a greater range of languages, we also anticipate UWN being used for tasks that explicitly make use of multilingual connections in the network, e.g. cross-lingual information retrieval or cross-lingual text classification.

We have created a public querying Web site for UWN (Figure 5.6 on page 157) that also accepts user contributions, which in the long run may allow us to address issues like correcting inaccurate arcs. Since the confidence estimates derived from the learnt models correlate quite well with the evaluated precision on the arcs, manual efforts could be channelled to focus explicitly on arcs with borderline confidence values and terms without accepted `means` arcs. An update submitted to the Web interface or an additionally imported translation dictionary for one language can subsequently lead to a sufficient amount of accumulated evidence to sway the model towards accepting mappings in entirely different languages. Hence, it is safe to expect continued growth and refinement in the future.

One issue that can be raised with regard to the lexical integration strategy is that it may happen that the set of semantic entities taken from the input wordnets is too limited to properly reflect language-specific phenomena. This issue will be resolved in the following chapter.

CHAPTER 4

Entity Integration

4.1 Introduction

We now turn to a second aspect of establishing large-scale knowledge bases. So far, we have demonstrated how new words (*term* entities) can be integrated into a knowledge base with a given inventory of possible semantic entities. In this chapter, we address the problem of integrating new *semantic* entities into a knowledge base.

Motivation. An English lexical knowledge base like WordNet is likely to lack semantic entities for certain language-specific concepts, e.g. Chinese has words for *elder* sisters ('姊', '姐姐'), French has specific types of educational institutions like grandes écoles, and of course even English has concepts missing in WordNet, e.g. '*mockumentary*' refers to a genre of film and television. We may also want to add domain-specific concepts or individual entities like cities, movie actors, or biomedical objects to our knowledge base. Such entities can be imported from separate knowledge sources providing their own, independent inventories of entity identifiers. The challenge then is merging these separate inventories of entities from different knowledge sources to produce a consistent integrated inventory of semantic entities. Figure 4.1 schematically describes this strategy. In practice, this will require paying special attention to the `equals` relation, which expresses equivalence between entities.

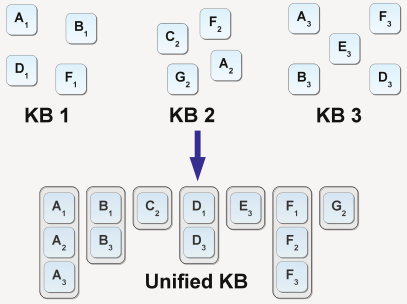


Figure 4.1: Entity integration strategy

The semantic entities could be derived from a wide range of different sources, e.g. one could take language- and domain-specific thesauri. One could also use sense induction algorithms to derive semantic entities from raw text, based on co-occurrence information (Schütze, 1992; Pereira et al., 1993). Domain-specific entities could come for example from biomedical or bibliographical datasets in the Linked Data Web (Bizer et al., 2009). In what follows, we will focus in particular on the open community-maintained encyclopedia Wikipedia. Wikipedia has not only turned the Internet into a more useful and linguistically diverse source of information, but is also increasingly being used in computational applications as a large-scale source of linguistic and encyclopedic knowledge. It is a splendid resource in this respect, because it goes beyond WordNet in describing a broad range of domain-specific concepts (e.g. `Diffeomorphism` as a concept from differential topology) as well as individual named entities (e.g. analytical philosopher `Hans Reichenbach`). Wikipedia does not cover verb, adjective, or adverb senses, but in many applications nouns and named entities are the most important items of interest. Projects like DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), WikiTaxonomy (Ponzetto and Strube, 2008), Intelligence-in-Wikipedia (Wu and Weld, 2008), and Freebase (Bollacker et al., 2008) have exploited the semi-structured nature of Wikipedia to produce valuable repositories of formal knowledge that are orders of magnitude larger than hand-crafted resources.

Additionally, while the English Wikipedia is the largest and most popular edition, the first non-English editions went online in March 2001 just two months after the English version, and the number has grown over the years. There are presently over 200 different editions of Wikipedia, even for minority languages like Faroese and Dhivehi, each providing a separate set of articles and categories.

In order to unify separate inventories of semantic entities, we assume there is some way to obtain `equals` links that connect equivalent entity identifiers. These can be mere heuristics, e.g. there has been extensive research on thesaurus and ontology mapping techniques (see Section 4.2). Later on, in Chapter 5, we will rely on heuristics of this sort to generate many `equals` links. Within Wikipedia, this process is even simpler, as Wikipedia offers cross-lingual *interwiki* links that e.g. connect the Japanese article about '*Education*' to the corresponding articles in over 100 other languages, and similarly for named entities like `UNESCO` or `Berlin University of the Arts`. In Figure 4.2, we see screenshots of an English-language article and of a corresponding Japanese-language article. The cross-lingual links are displayed as a navigational aid in a box at the side of the article text. Such links are extraordinarily valuable for cross-lingual applications (Ferrández et al., 2007; Nguyen et al., 2009; Pasternack and Roth, 2009), and for our purposes can be re-interpreted as `equals` links.

Ideally, a set of entities connected directly or indirectly via `equals` links would all describe the same entity or concept, as in Figure 4.3. Genuine equivalence, after all, is a transitive and symmetric relation. However, heuristic linking functions often produce inaccurate links. Not only is it easy to confuse similar but reasonably distinct concepts, e.g. the German *Fachhochschule*, a specific type of tertiary education institution, with regular universities. Heuristics sometimes deliver entirely inappropriate links stemming from disambiguation errors.

Even in Wikipedia, due to conceptual drift, different granularities, as well as mistakes made by editors and automated bots, we occasionally find concepts as different as `Economics` and `Manager` in the same weakly connected component in the graph of cross-lingual interwiki links. Figure 4.4 shows a larger excerpt of the connected component from Figure 4.3, where we see that it conflates the concept of television as a medium in general with the concept of TV sets as physical devices. Such issues

Figure 4.2: Wikipedia articles in English and Japanese

are unfortunately much more common than one would expect. Filtering out inaccurate links would enable us to exploit Wikipedia's multilinguality in a much safer manner and allow us to create an integrated multilingual inventory of entities.

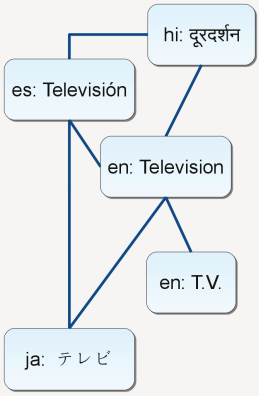


Figure 4.3: Entities connected by `equals` links

Problem Statement. We assume that we are given entity identifiers from different sources, e.g. different editions of Wikipedia. We further assume that we have some way of obtaining weighted `equals` statements that reveal equivalences between entities, as well as so-called distinctness assertions that will express weighted disequality information.

The goal will be to obtain a single integrated knowledge base, where each connected component cleanly represents a single entity or concept, and the weighted `equals` arcs have been reconciled with the weighted distinctness assertions. These notions will be made clearer later in this chapter.

Contribution. Our research contributions are:

1. We identify criteria to detect inaccurate connections in Wikipedia's cross-lingual link structure.
2. We formalize the task of integrating entities from different sources and removing inaccurate `equals` connections between them in light of distinctness information as a combinatorial optimization problem. Unlike most previous work on thesaurus and ontology

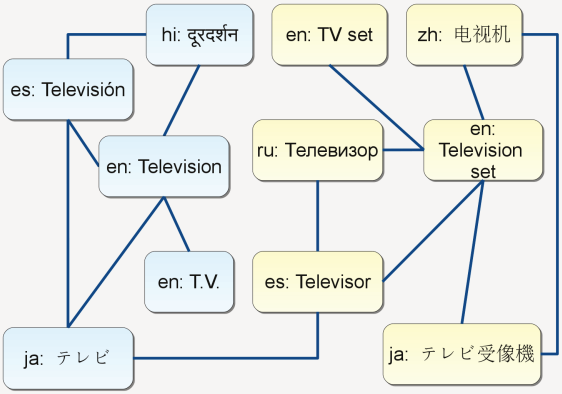


Figure 4.4: Connected component with inaccurate links (simplified)

mapping as well as record linkage, this formalization accounts for the consistency of mappings between more than just two knowledge sources and allows capturing distinctness between arbitrary subsets of entities.

3. We introduce an algorithm that aims to solve this problem in a minimally invasive way. This algorithm has an approximation guarantee with respect to optimal solutions.
4. We show how this algorithm can be applied to combine multiple entity inventories, e.g. all editions of Wikipedia, into a single large-scale multilingual register of named entities and concepts.

Overview. The rest of this chapter is organized as follows. Section 4.2 discusses related previous work. Section 4.3 specifies what the initial knowledge sources that serve as input should look like. Section 4.4 describes how we identify inaccuracies in Wikipedia's cross-lingual link structure and formalizes the task of reconciling equality information with distinctness information as an optimization problem. Section 4.5 introduces an approximation algorithm for solving this problem. Finally, Section 4.6 presents our experimental results, and Section 4.7 discusses the implications of these results.

4.2 Previous Work

Entity Linking. Over the years, there have been numerous studies on heuristics to create mappings between two repositories of entities. In the EuroWordNet project (Vossen, 1998), this strategy for building a lexical database was referred to as the *merge approach*. Rada and Martin (1987) investigated how medical thesauri like SNOMED and MeSH can be connected in order to facilitate interoperability, and similar links have been created in other domains (de Melo and Weikum, 2008a; Lauser et al., 2008). For ontology alignment, a number of linking heuristics have been proposed to align classes and individual instances in two ontologies, ranging from simple string similarity measures to more sophisticated measures that consider the ontological context (Euzenat and Shvaiko, 2007). Such heuristics can serve as input to our algorithm, which greatly boosts their value because our algorithm can make use of them to merge more than two knowledge sources into a single unified knowledge base while taking unique names assumptions into consideration.

Similarly, in the relational database world and in statistics, there have been a wide range of entity resolution, deduplication, and record linkage techniques that first compare the individual attribute fields associated with a record, and then produce similarity scores between entire records. Fields have been compared using string similarity measures (Bilenko and Mooney, 2003), term overlap scores (Cohen et al., 2003), and other heuristics. Record similarity can be assessed by combining the scores for individual fields using the Fellegi-Sunter model (Fellegi and Sunter, 1969; Winkler, 1999), weighted sums (Bilenko et al., 2005), or rules (Lee et al., 2004). See Gu et al. (2003) and Elmagarmid et al. (2007) for surveys.

Additional aspects that have been studied include how to take into account mutual dependencies between similarity scores (Dong et al., 2005; Kalashnikov and Mehrotra, 2006) and how to determine potential match candidates more efficiently (Hernández and Stolfo, 1995; Monge and Elkan, 1997; Benjelloun et al., 2009).

The entity similarity scores delivered as output by such methods can as well serve as input to our algorithm, which can then make them more consistent.

Bipartite Graphs. When two knowledge sources are connected, the similarity matrices produced by similarity heuristics of the sort just mentioned can be interpreted as weights of a bipartite graph. If one assumes that within each knowledge source all entities are mutually distinct, then a consistent mapping between the two sources corresponds to an *independent edge set* or *matching*, i.e. a set of pairwise non-adjacent edges.

The *stable marriage problem* is the problem of finding a *stable matching*, where one removes edges to obtain a matching, choosing the edges based only on preference rankings, without regard for the precise edge weights. The *assignment problem* (also called LSAP, linear sum assignment problem) considers the task of finding a maximum weight matching, where the total weight of the retained edges is maximized and the total weight of the removed edges is minimized. Solutions to this problem can be found using the Kuhn-Munkres ("Hungarian") algorithm (Munkres, 1957; Burkard et al., 2009).

The k-index assignment problem goes beyond bipartite graphs by extending the problem from 2 to k data sources, aiming at connected components of size k, each consisting of one element from each data source (Burkard et al., 2009). In our work we study the much more general case of an arbitrary number of knowledge sources with possible equivalences between arbitrary nodes in the graph, and an arbitrary number of distinctness assertions involving arbitrary sets of nodes (rather than just distinctness between all nodes within each knowledge source).

General Graphs. Our integration algorithm uses theoretical ideas put forward by researchers studying graph cuts (Leighton and Rao, 1999; Garg et al., 1996; Avidor and Langberg, 2007), as will be explained later on in further detail.

Our problem setting is related to that of correlation clustering (Bansal et al., 2004), where nodes of a graph with positively and negatively labelled similarity edges are clustered such that similar items are grouped together, however our approach is much more generic than conventional correlation clustering, e.g. it supports arbitrary edge and distinctness weights. This is important, as `equals` links are often generated by heuristics, and may be much less reliable than individual distinctness asser-

tions. Charikar et al. (2005) studied a variation of correlation clustering that is more expressive than the standard variant, but since a negative edge would have to be added between each relevant pair of entities in a distinctness assertion, the approximation guarantee would only be $O(\log n\ |V|^2)$ and the ability to merge an entity e.g. with all redirects of another entity at a fixed cost as in our framework would no longer be given.

McCallum and Wellner (2004) proposed an undirected graphical model that has similar restrictions and is solved heuristically without guarantees. Similarly, Bhattacharya and Getoor (2007) developed a greedy heuristic clustering framework where the cluster similarity can be set to zero if there is some prior knowledge of distinctness.

Cohen et al. (2000) defined the task of "hardening soft information sources" by assigning consistently used identifiers to groups of items. They consider costs for merging possibly distinct items, but their formalization cannot capture the global costs for removing equivalence edges.

Minimally invasive repair operations on graphs have also been studied for graph similarity computation (Zeng et al., 2009), where two graphs are provided as input, and need to be compared.

Wikipedia. A number of projects have used Wikipedia as a database of named entities (Silberer et al., 2008). The most well-known are probably DBpedia (Auer et al., 2007), which serves as a hub in the Linked Data Web, Freebase (Bollacker et al., 2008), which combines human input and automatic extractors, and YAGO (Suchanek et al., 2007), which adds an ontological structure on top of Wikipedia's entities. WikiTaxonomy (Ponzetto and Strube, 2007) re-organizes Wikipedia's category system as a taxonomy.

Gabrilovich and Markovitch (2007) interpreted Wikipedia's articles as concepts in order to assess semantic relatedness between texts, which Hassan and Mihalcea (2009) extended for cross-lingual text similarity. Wikipedia has further been used cross-lingually for cross-lingual IR (Su et al., 2007; Nguyen et al., 2009), question answering (Ferrández et al., 2007) as well as for learning transliterations (Pasternack and Roth, 2009), among other things. Adar et al. (2009) and Bouma et al. (2009) show how cross-lingual links can be used to propagate information from one Wikipedia's infoboxes to another edition.

Mihalcea and Csomai (2007) have studied predicting new links within a single edition of Wikipedia. Sorg and Cimiano (2008) considered the problem of suggesting new cross-lingual links, which could be used as additional inputs in our problem. Völker et al. (2007) investigated a machine learning approach for learning distinctness information in ontologies.

4.3 Knowledge Sources

Input Graph. As input, our approach takes one or more knowledge sources as well as `equals` links connecting entities from those knowledge sources. The links express likely equivalence relationships, but contain false positives. To simplify notation, we model the input here as a simple undirected graph $G = (V, E)$ with edge weights $w(e)$, in which undirected edges represent `equals` links in either direction. This is a natural choice, as equality is a symmetric relation. A knowledge base $G' = (V, A, \Sigma)$ in the sense of Chapter 2 can easily be converted into such a graph by defining E as the set of all node pairs connected via `equals` links in either direction, and edge weights straightforwardly corresponding to the sum of all weights of relevant `equals` arcs.

Definition 4.1 (Equivalence Graph) *Given a knowledge base $G' = (V, A, \Sigma)$, the corresponding undirected graph of equivalences is defined as $G = (V, E)$ with $E = \{(u, v) \mid (u, v, r, w) \in A \vee (v, u, r, w) \in A$ where $r =$ `equals`$, u \neq v\}$. Given an edge $e = (u, v) \in E$ and $r =$ `equals`, the edge weight of e is*

$$w(e) = \sum_{(u,v,r,w) \in A} w + \sum_{(v,u,r,w) \in A} w.$$

Equivalence Information. In practice, the `equals` links can be obtained using various types of heuristics. In de Melo and Weikum (2008a), we investigated mapping multilingual thesauri like AGROVOC (Leatherdale et al., 1982) with other resources. In de Melo and Weikum (2010c), we interlinked registries for ISO 639, ISO 15924, and other language- and geopolitical standards with WordNet and Wikipedia (see also Section 4.6.6). In Chapter 5, we show how entities from Wikipedia can be connected to WordNet synsets. To keep things simple for now,

we focus only on cross-lingual links between entities in Wikipedia in most of this chapter.

The union of cross-lingual links provided by all editions of Wikipedia can easily be modelled using a simple undirected graph as described above. In our experiments, we simply interpret each original cross-lingual link as an `equals` link with a weight of one. This implies $w(e) = 2$ if there are reciprocal links between the two pages, 1 if there is a single link, and 0 otherwise. However, our framework is flexible enough to deal with more advanced weighting schemes. For instance, one could easily plug in cross-lingual measures of semantic relatedness between article texts based for instance on vector dot products.

Additionally, we also consider redirect links that automatically redirect Wikipedia users from one page to another. When browsing Wikipedia on the Web, interwiki links to redirects are handled transparently, however there are many redirects with titles that do not co-refer, e.g. redirects from members of a band to the band, or from aspects of a topic to the topic in general. We only inferred `equals` from redirect links in the following cases:

1. the titles of redirect source and redirect target match after case conversion, string encoding normalization using the Unicode NFKD standard (Davis and Dürst, 2008), diacritics removal, and removal of punctuation characters
2. the redirect uses certain templates or categories that indicate co-reference with the target (alternative names, abbreviations, etc.)

We treat redirections like reciprocal interwiki links by assigning them a weight of 2.

Inaccurate Equivalence Information. It turns out that an astonishing number of connected components in this graph harbour inaccurate links between entity identifiers. For instance, the Esperanto article `Germana Imperiestro` is about German emperors and another Esperanto article `Germana Imperiestra Regno` is about the German Empire, but, as of October 2010, both are linked to the English and German articles about the German Empire. Over time, some inaccurate links may be fixed, but in this and in large numbers of other cases, the imprecise connection has

persisted for many years. In order to detect such cases, we need to have some way of specifying that two articles are likely to be distinct.

4.4 Considering Distinctness Information

4.4.1 Distinctness Assertions

Often, we are able to guess that certain entities are likely to be distinct. In Figure 4.4, we saw a connected component conflating the concept of television as a medium with the concept of TV sets as devices. Here, among other things, we would like to formally express that `Television` and `T.V.` are very likely distinct from `Television set` and `TV set`. In general, we may have several sets of entities $D_{i,1}, \ldots, D_{i,l_i}$, for which we would like to assert that any two entities u,v from different sets are pairwise distinct with some degree of confidence or weight. In our example, $D_{i,1} = \{$`Television`, `T.V.`$\}$ would be one set, and $D_{i,2} = \{$`Television set`, `TV set`$\}$ would be another set, which means that we are assuming `Television`, for example, to be distinct from both `Television set` and `TV set`. Figure 4.5 presents an extended scenario, where we have a third set $D_{i,3}$ containing `Television station` and `TV station`, which are also considered pairwise distinct from members of the other sets.

Definition 4.2 (Distinctness Assertions) *Given a set of nodes V, a* distinctness assertion *is a collection $D_i = (D_{i,1}, \ldots, D_{i,l_i})$ of pairwise disjoint (i.e. $D_{i,j} \cap D_{i,k} = \emptyset$ for $j \neq k$) subsets $D_{i,j} \subset V$ expressing that any two nodes $u \in D_{i,j}$, $v \in D_{i,k}$ from different subsets ($j \neq k$) are asserted to be distinct from each other with some weight $w(D_i) \in \mathbb{R}$.*

Our approach relies on the fact that there are reasonably good heuristics to produce such weighted distinctness assertions automatically. For example, any unique names assumption (Russell and Norvig, 2010, p. 299) for an individual knowledge source implies that distinct entity identifiers from that knowledge source refer to distinct entities. In the case of Wikipedia, we found that many components with inaccurate links can be identified automatically with the following distinctness assertions.

First of all, an important observation is that two articles from the same Wikipedia edition very likely describe distinct concepts unless

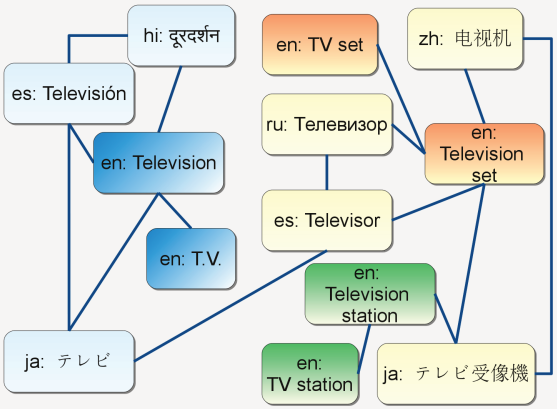


Figure 4.5: Distinctness assertion example

they are redirects of each other. For example, the entity `Educational television` is distinct from `Educational Television (Hong Kong)` – the latter is a specific TV series, and `Television` is distinct from `Television set`.

Criterion 4.1 (Distinctness of Articles). For each language-specific edition of Wikipedia, a separate assertion $(D_{i,1}, D_{i,2}, \dots)$ can be made, where each $D_{i,j}$ contains an individual *article* together with its respective redirection pages. Redirection pages that are marked by a category or template as involving topic drift are kept in a separate $D_{i,j}$, distinct from the one of their redirect targets.

The criterion additionally accounts for the fact that certain redirects are explicitly marked by a category or template as involving topic drift, e.g. redirects from songs to albums or artists, from products to companies, etc. Similar distinctness assertions can be created for categories. For instance, the category `Documentary films` is distinct from the category `Documentary filmmakers`.

Criterion 4.2 (Distinctness of Categories). For each language-specific edition of Wikipedia, a separate assertion $(D_{i,1}, D_{i,2}, \dots)$ is made, where

each $D_{i,j}$ contains a *category* page together with any (so-called "soft") redirects.

Another criterion can be used to give us more specific distinctness information when there is an interwiki link with an anchor identifier. The English article `Division by zero`, for instance, links to the German `Null#Division`. The latter is only a part of a larger article about the number zero in general, so we can add a distinctness assertion to ensure that `Division by zero` is separated from `Null`.

Criterion 4.3 (Distinctness for Anchor Identifiers). For each interwiki link or redirection with an anchor identifier, we add an assertion $(D_{i,1}, D_{i,2})$ where $D_{i,1}$,$D_{i,2}$ represent the respective articles without anchor identifiers.

These different types of distinctness assertions can automatically be instantiated for all articles and categories of different Wikipedia editions. The assertion weights are tunable; the simplest choice is using a uniform weight for all assertions (note that these weights are different from the edge weights in the graph). We will revisit this issue in our experiments.

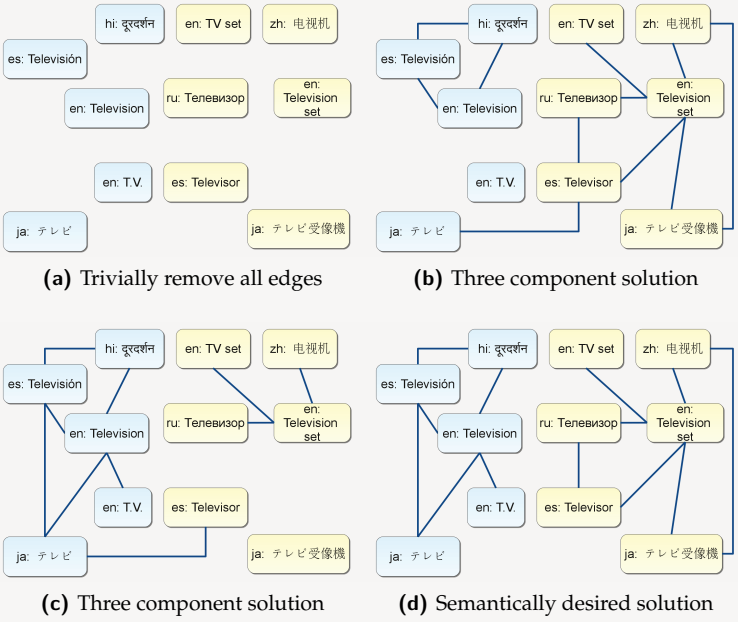
4.4.2 Enforcing Consistency

Reconciling Equivalence and Distinctness. Given a graph G representing `equals` links between entity identifiers, as well as distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$, we may find that nodes that are asserted to be distinct are in the same connected component, as in Figure 4.4. We can then attempt to perform repair operations to reconcile the graph's link structure with the distinctness assertions and obtain global consistency. There are two ways to modify the input, and for each we can think of the corresponding weights as a sort of *cost* that quantifies how much we are changing the original input:

a) **Edge cutting:** We may remove an edge $e \in E$ from the graph, paying cost $w(e)$.
b) **Distinctness assertion relaxation:** We may remove a node $v \in V$ from a distinctness assertion D_i, paying cost $w(D_i)$.

Removing edges allows us to split connected components into multiple smaller components, thereby ensuring that two nodes asserted to be

distinct are no longer connected directly or indirectly. The graph in Figure 4.4 could be reconciled with a distinctness assertion between English Wikipedia entities {`Television`, `T.V.`} and {`Television set`, `TV set`} by deleting the edge from the Spanish `Televisor` (*'TV set'*) to the Japanese *'television'* article, for instance. In contrast, removing nodes from distinctness assertions means that we decide to give up our claim of them being distinct, instead allowing them to share a connected component.



(a) Trivially remove all edges **(b)** Three component solution

(c) Three component solution **(d)** Semantically desired solution

Figure 4.6: Potential solutions

Solution Costs. Figure 4.6 shows some possible ways of modifying the input from Figure 4.4 to satisfy a distinctness assertion between {`Television`, `T.V.`} and {`Television set`, `TV set`}. Additionally, there

is also the option of just keeping the input graph unchanged by removing nodes from the distinctness assertion.

Each of these solutions is consistent, but they are certainly not all equally desirable. Our reliance on costs is based on the assumption that the link structure or topology of the graph together with the edge and distinctness weights provide the best indication of which solution to choose. In this example, we have a distinctness assertion between nodes in two densely connected clusters that are tied together only by a single spurious link. In reality, with the criteria mentioned earlier, we would even be having multiple distinctness assertions between these two clusters. Solution (d) in Figure 4.6 removes only this single incorrect edge, while the remaining solutions in the Figure remove more than one edge.

The additional option of not removing any edges and instead just removing nodes from the distinctness assertion does not make much sense in this case, as we can separate the two components by removing only a single edge. In other cases, however, we may find that separating nodes by removing edges would definitely incur high costs. If we had only a single distinctness assertion between `T.V.` and `Television` in Figure 4.7, depending on the weights, it would perhaps be wiser to just retain the original input graph and opt for relaxing the distinctness assertion by removing one of these two nodes from it.

Objective. The aim will thus be to balance the costs for removing different edges from the graph with the costs for removing nodes from distinctness assertions to produce a consistent solution with a minimal total repair cost. This allows us to accommodate our knowledge about distinctness while staying as close as possible to what Wikipedia provides as input.

This is formalized as what we call the **Weighted Distinctness-Based Graph Separation (WDGS)** problem. Let G be an undirected graph with a set of vertices V and a set of edges E weighted by $w : E \to \mathbb{R}$. We assume we have n distinctness assertions $D_1, \ldots, D_n$, each consisting of one or more sets $D_i = (D_{i,1}, \ldots, D_{i,l_i})$. If we use a set $C \subseteq V$ to specify which edges we want to cut from the original graph, and sets U_i to specify which nodes we want to remove from the corresponding

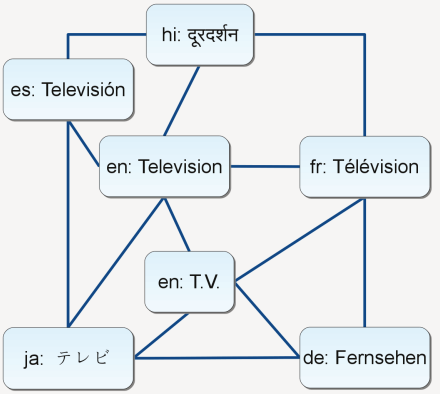


Figure 4.7: Connected component not worth breaking up

distinctness assertions D_i, we can begin by defining WDGS solutions as follows.

Definition 4.3 (WDGS Solution) *Given a graph $G = (V, E)$ and n distinctness assertions $D_1, \ldots, D_n$, a tuple $(C, U_1, \ldots, U_n)$ is a valid* WDGS *solution if and only if for all $i, j, k \neq j$ and any two nodes $u \in D_{i,j} \setminus U_i$, $v \in D_{i,k} \setminus U_i$, we have $\mathrm{P}(u, v, E \setminus C) = \emptyset$, i.e. the set of paths from u to v in the graph $(V, E \setminus C)$ is empty.*

In other words, after removing nodes in U_i from matching $D_{i,j}$, and removing edges in C from E, there are no paths left from $u \in D_{i,j}$ to $v \in D_{i,k}$ (provided $k \neq j$).

Definition 4.4 (WDGS Cost) *Let $w : E \to \mathbb{R}$ be a weight function for edges $e \in E$, and $w(D_i)$ $(i = 1 \ldots n)$ be weights for the distinctness assertions. The (total) cost of a* WDGS *solution $S = (C, U_1, \ldots, U_n)$ is then defined as*

$$\begin{aligned} c(S) &= c(C, U_1, \ldots, U_n) \\ &= \left[\sum_{e \in C} w(e)\right] + \left[\sum_{i=1}^{n} |U_i|\, w(D_i)\right] \end{aligned}$$

The WDGS problem can then straightforwardly be defined as follows.

Definition 4.5 (WDGS) *A* WDGS *problem instance P consists of a graph $G = (V, E)$ with edge weights $w(e)$ and n distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$. The objective consists in finding a solution $(C, U_1, \ldots, U_n)$ with minimal cost $c(C, U_1, \ldots, U_n)$.*

It turns out that finding optimal solutions efficiently is a computationally hard problem.

Theorem 4.6 (Hardness) WDGS *is an NP-hard problem.*

Proof. We reduce the well-known NP-complete vertex cover problem to the problem of whether a WDGS solution exists with a cost of at most k. We are given a graph $G = (V, E)$ and want to know whether a set $V' \subseteq V$ of size at most k exists such that for each edge $e = (u, v) \in E$ either $u \in V'$ or $v \in V'$ (or both).

Now construct a new star-shaped graph G^+ with an added central node v^+ that is connected to all $v \in V$, i.e. $G^+ = (V \cup \{v^+\}, \{(v^+, v) \mid v \in V\})$, with uniform edge weights $w^+(e) = 1$ for all e. Add a distinctness assertion $(\{u\}, \{v\})$ with weight $k + 1$ for each (u, v) in the original edge set E. Given a WDGS solution $(C, U_1, \ldots, U_n)$ for G^+ with $c(C, U_1, \ldots, U_n) \leq k$, we know that all $U_i = \emptyset$, i.e. all distinctness assertions are satisfied, because otherwise $c(C, U_1, \ldots, U_n) \geq k + 1$. Hence, for each $(u, v) \in E$, no paths from u to v exist in G^+ after removal of C, so either $(v^+, u) \in C$ or $(v^+, v) \in C$. Any WDGS solution $(C, U_1, \ldots, U_n)$ with cost at most k thus provides us with a vertex cover $V' = (\bigcup_{e \in C} e) \backslash \{v^+\}$. The cost for each edge in C is 1, so this vertex cover will have a size of $|V'| = |C| = \sum_{e \in C} w^+(e) = c(C, U_1, \ldots, U_n) \leq k$.

Conversely, any vertex cover V' with size k for G yields a WDGS solution $C = \{(v^+, v) \mid v \in V'\}$, $U_i = \emptyset$ with cost k for G^+. The edge set of G' after applying this solution is $\{(v^+, v) \mid v \in V\} \setminus C = \{(v^+, v) \mid v \in V \setminus V'\}$. Since every edge in E is covered by V', for any $(u, v) \in E$, either (v^+, v) or (v^+, u) will be missing from G' after removing edges. Hence, it will not provide any path from u to v. The cost of the WDGS solution is $\sum_{e \in C} w(e) = |V'| = k$. Hence, if no WDGS solution with cost k exists, then by modus tollens no vertex cover of size k exists. □

For some NP-hard problems, efficient algorithms exist that provably come extremely close to the optimal solution. An NP minimization problem is approximated within a given factor f if for every possible problem instance a solution is obtained with a cost at most f times the optimal cost, where f can be a function of the length of a problem instance. The class APX consists of all NP optimization problems that can be approximated within a *constant* f in polynomial time. Unfortunately, even when we are only interested in such approximations, WDGS turns out to be difficult.

Theorem 4.7 (Hardness of Approximation) WDGS *is APX-hard. It is NP-hard to approximate* WDGS *within a factor of 1.3606. If the Unique Games Conjecture (Khot, 2002) holds, then it is NP-hard to approximate* WDGS *within any constant factor* $\alpha > 0$.

Proof. Given an instance of the minimum vertex cover problem for a graph $G = (V, E)$, we can again construct a star-shaped graph $G^+ = (V \cup \{v^+\}, \{(v^+, v) \mid v \in V\})$ with uniform edge weights $w^+(e) = 1$, this time adding a distinctness assertion $(\{u\}, \{v\})$ with weight $|V| + 1$ for each $(u, v) \in E$. An optimal WDGS solution $(C, U_1, \ldots, U_n)$ for G^+ will have $U_i = \emptyset$ for all i, and as in the proof of Theorem 4.6 imply a vertex cover $(\bigcup_{e \in C} e) \setminus \{v^+\}$ of size $c(C, U_1, \ldots, U_n)$. Again, any optimal vertex cover implies an optimal WDGS solution with the same cost, so this reduction is gap-preserving. Hence, APX-hardness follows from the hardness results for minimum vertex cover by Clementi and Trevisan (1996). Similarly, hardness to approximate within a factor of 1.3606 follows with the minimum vertex cover result by Dinur and Safra (2005).

For showing that WDGS is not in APX given the Unique Games Conjecture unless P = NP, we refer to previous results by Chawla et al. (2005) and provide a gap-preserving reduction of the minimum multicut problem to WDGS . Given a graph $G = (V, E)$ with a positive cost $c(e)$ for each $e \in E$, and a set $D = \{(s_i, t_i) \mid i = 1 \ldots k\}$ of k demand pairs, our goal is to find a multicut M with respect to D with minimum total cost $\sum_{e \in M} c(e)$. We convert each demand pair (s_i, t_i) into a simple distinctness assertion $D_i = (\{s_i\}, \{t_i\})$ with weight $w(D_i) = 1 + \sum_{e \in E} c(e)$. An optimal WDGS solution $(C, U_1, \ldots, U_k)$ with cost c then implies a multicut C with the same weight: Since $w(D_i) > \sum_{e \in E} c(e)$, the solution can only be optimal if for all i, $U_i = \emptyset$. Hence, a multicut C will

satisfy all demand pairs. C is a minimal multicut because any multicut C' with lower cost would imply a valid WDGS solution $(C', \emptyset, \ldots, \emptyset)$ with a cost lower than the optimal solution $(C, U_1, \ldots, U_k)$, which is a contradiction. □

4.5 Approximation Algorithm

We now present an algorithm devised to tackle the WDGS problem, allowing us to integrate entities from different knowledge sources under consideration of available distinctness information about entities. Although, as we just saw, it is computationally hard to obtain optimal solutions, the algorithm not only works well in practice but also has theoretical properties that allow it to remain within certain bounds of the optimum. We are able to show that it is a polynomial-time approximation algorithm with an approximation factor of $4\ln(nq+1)$ where n is the maximal number of distinctness assertions within a connected component and $q = \max_{i,j} |D_{i,j}|$. This means that, no matter what problem instance P we have to deal with, we can guarantee

$$\frac{c(S(P))}{c(S^*(P))} \leq 4\ln(nq+1),$$

where $S(P)$ is the solution determined by our algorithm, and $S^*(P)$ is an optimal solution. Note that this approximation guarantee is independent of how long each D_i is, and that the factor merely represents an upper bound on the worst case scenario. In practice, the results tend to be much closer to the optimum, as will be shown in Section 4.6.

4.5.1 Description

Overview. Our algorithm starts out with a graph $G = (V, E)$ and distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$. Without loss of generality, we may assume that G consists of a single connected component. If there are multiple connected components, we can simply consider each respective subgraph as a separate problem.

The algorithm first solves a linear program (LP) relaxation of the original problem, which gives us hints as to which edges should most likely be cut and which nodes should most likely be removed from

distinctness assertions. Note that this is a continuous LP, not an integer linear program (ILP); the latter would not be tractable due to the large number of variables and constraints of the problem.

After solving the linear program, a new – extended – graph is constructed and the optimal LP solution is used to define a distance metric on it. The final solution is obtained by smartly selecting regions in this extended graph as the individual output components, by employing a region growing technique in the spirit of the seminal work by Leighton and Rao (1999). Edges that cross the boundaries of these regions are cut.

Definition 4.8 (WDGS Linear Program) *Given a* WDGS *instance, we define a linear program of the following form:*

minimize

$$\sum_{e \in E} d_e w(e) + \sum_{i=1}^{n} \sum_{j=1}^{l_i} \sum_{v \in D_{i,j}} u_{i,v} w(D_i)$$

subject to

$$s_{i,j,v} = u_{i,v} \qquad \forall i,\, j<l_i,\, v \in D_{i,j} \qquad (1)$$

$$s_{i,j,v} + u_{i,v} \geq 1 \qquad \forall i,\, j<l_i,\, v \in \bigcup_{k>j} D_{i,k} \qquad (2)$$

$$s_{i,j,v} \leq s_{i,j,u} + d_e \qquad \forall i,\, j<l_i,\, e \in E,\, u,v \neq u \in e \qquad (3)$$

$$d_e \geq 0 \qquad \forall e \in E \qquad (4)$$

$$u_{i,v} \geq 0 \qquad \forall i,\, v \in \bigcup_{j=1}^{l_i} D_{i,j} \qquad (5)$$

$$s_{i,j,v} \geq 0 \qquad \forall i,\, j<l_i,\, v \in V \qquad (6)$$

The LP uses decision variables d_e and $u_{i,v}$, and auxiliary variables $s_{i,j,v}$ that we refer to as *separation distance* variables. The d_e variables indicate whether (or actually, since this is a continuous LP: to what degree) an edge e should be deleted, and the $u_{i,v}$ variables indicate whether (to what degree) v should be removed from a distinctness assertion D_i. The LP objective corresponds to Definition 4.4, aiming at minimizing the total costs.

A separation distance variable $s_{i,j,v}$ reflects to what degree a node v has been separated from nodes in a set $D_{i,j}$ of a distinctness assertion. If $s_{i,j,v} = 0$, then v is still connected to nodes in $D_{i,j}$. Constraints (1) and (2) enforce separation distances between $D_{i,j}$ and all nodes in $D_{i,k}$ with $k > j$. For instance, for distinctness between `Television` and `Television set`, they might require `Television set` to have a separation distance

of 1, while `Television` has a distance of 0. The separation distances are tied to the deletion variables d_e for edges in Constraint (3) as well as to the $u_{i,v}$ in Constraints (1) and (2). This means that a separation distance $s_{i,j,v} + u_{i,v} \geq 1$ can only be obtained if edges are deleted on every path between `Television` and `Television set`, or if at least one of these two nodes is removed from the distinctness assertion (by setting the corresponding $u_{i,v}$ to non-zero values). Constraints (4), (5), (6) ensure non-negativity.

Extended Graph. Having solved the linear program, the next major step is to convert the optimal LP solution into the final – discrete – solution. We cannot rely on standard rounding methods to turn the optimal fractional values of the d_e and $u_{i,v}$ variables into a valid solution. Often, all solution variables have small values and rounding will merely produce an empty $(C, U_1, \ldots, U_n) = (\emptyset, \emptyset, \ldots, \emptyset)$.

Instead, a more sophisticated technique is necessary. We define an extended graph G' with a distance metric d between nodes derived from the optimal solution of the LP. The algorithm then operates on this graph, in each iteration selecting regions that become output components and are removed from the graph. A simple example is shown in Figure 4.8. The extended graph contains additional nodes representing distinctness assertion elements and edges representing whether a node remains in the distinctness assertion. Cutting one of these additional edges corresponds to removing the connected node from the distinctness assertion.

Definition 4.9 (Extended Graph) *Given $G = (V, E)$ and distinctness assertions $D_1, \ldots, D_n$ with weights $w(D_i)$, we define an undirected graph $G' = (V', E')$ where*

$$V' = V \cup \{\mathrm{v}_{i,v} \mid i = 1 \ldots n, v \in \bigcup_j D_{i,j}, w(D_i) > 0\},$$

$$E' = \{e \in E \mid w(e) > 0\} \cup \{(v, \mathrm{v}_{i,v}) \mid v \in \bigcup_j D_{i,j}, w(D_i) > 0\}.$$

We accordingly extend the definition of $w(e)$ to additionally cover the new edges by defining $w(e) = w(D_i)$ for $e = (v, \mathrm{v}_{i,v})$. We also extend it for sets S of edges by defining $w(S) = \sum_{e \in S} w(e)$.

Definition 4.10 (Distance Metric) *Based on the optimal linear program solution (variables d_e, $u_{i,v}$), we define a node distance metric*

$$d(u,v) = \begin{cases} 0 & u = v \\ d_e & e = (u,v) \in E \\ u_{i,v} & u = \mathrm{v}_{i,v} \\ u_{i,u} & v = \mathrm{v}_{i,u} \\ \min\limits_{p \in \mathrm{P}(u,v,E')} \sum\limits_{(u',v') \in p} d(u',v') & \textit{otherwise,} \end{cases}$$

where $\mathrm{P}(u,v,E')$ denotes the set of acyclic paths between two nodes in E'.

Definition 4.11 (Fractional Solution Cost) *We further fix*

$$\hat{c}_\mathrm{f} = \sum_{(u,v) \in E'} d(u,v)\, w(e)$$

as the weight of the fractional solution of the LP, based on E' from Definition 4.9 ($\hat{c}_\mathrm{f}$ is a constant based on the original E', irrespective of later modifications to the graph).

We will later show that this is a lower bound on the cost of the optimal solution.

Regions. In this extended graph, we consider regions with a given radius with respect to the distance metric. Regions will later essentially become the output components representing single concepts or entities, while whatever edges cross region boundaries will later be cut.

Definition 4.12 *Around a given node v in G', we consider regions $R(v,r) \subseteq V$ with radius r. The cut $C(v,r)$ of a given region is defined as the set of edges in G' with one endpoint within the region and one outside the region.*

$$R(v,r) = \{v' \in V' \mid d(v,v') \leq r\}$$
$$C(v,r) = \{e \in E' \mid |e \cap R(v,r)| = 1\}$$

Definition 4.13 *For sets of nodes $S \subseteq V$, we define*

$$R(S,r) = \bigcup_{v \in S} R(v,r),$$
$$C(S,r) = \bigcup_{v \in S} C(v,r).$$

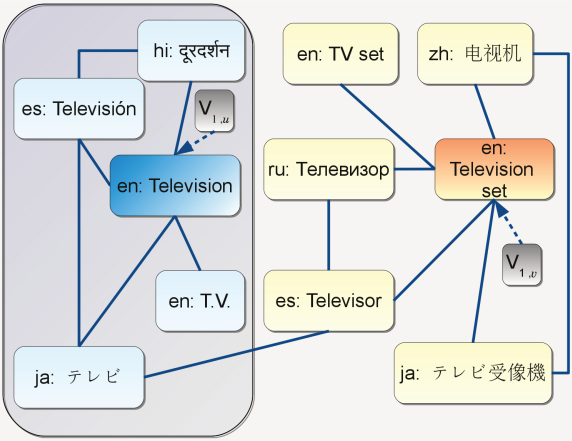


Figure 4.8: Extended graph with two added nodes $\mathrm{v}_{1,u}$, $\mathrm{v}_{1,v}$ representing distinctness between `Televisión` and `Televisor`, and a region around $v_{1,u}$ that would cut the link from the Japanese '*Television*' to `Televisor`

Choosing Regions. Our goal then becomes determining the extent of such regions in a way that minimizes the overall costs of these cuts. The good news is that the linear program solution can help us with this by revealing which choices of regions incur high cut costs with regard to our optimization objective. Given a region with radius r around a node v, we define $\hat{c}(v,r)$ based on the distance metric defined earlier. Since the distance metric is based on the optimal linear program solution, this function essentially gives us a bound on how good the optimal solution can be in a particular region. The relationship will be made more explicit later on in the proofs.

Definition 4.14 *Given* $q = \max_{i,j} |D_{i,j}|$*, we define*

$$\hat{c}(v,r) = \sum_{\substack{e=(u,u')\in E':\\ e\subseteq R(v,r)}} d(u,u')\, w(e) + \sum_{\substack{e\in C(v,r)\\ v'\in e\cap R(v,r)}} (r - d(v,v'))\, w(e) \tag{4.1}$$

$$\hat{c}(S,r) = \left[\sum_{v\in S} \hat{c}(v,r)\right] + \frac{1}{nq}\hat{c}_{\mathrm{f}} \tag{4.2}$$

The first summand for $\hat{c}(v,r)$ accounts for the edges entirely within the region, and the second one accounts for the edges in the cut $C(v,r)$ to the extent that they are within the radius. The definition of $\hat{c}(S,r)$ contains an additional slack component $\frac{1}{nq}\hat{c}_{\mathrm{f}}$ that is required for the approximation guarantee proof.

Algorithm. Based on these definitions, Algorithm 4 uses the LP solution to construct the extended graph with its distance measure. It then repeatedly, as long as there is an unsatisfied assertion D_i, chooses a set S of nodes containing a $\mathrm{v}_{i,v}$ node for one node v from each relevant $D_{i,j}$ in D_i. In each iteration, it starts out with the respective nodes in S, and simultaneously grows $|S|$ regions with the same radius around them, a technique previously suggested by Avidor and Langberg (2007). These regions roughly correspond to the connected components that serve as the final solution in the output graph.

The common radius r of the regions in each iteration could be chosen in different ways. The largest value in D is $\frac{1}{2}$, so we know that any radius chosen by our algorithm will be at most $\frac{1}{2} - \epsilon$. This is important, because a radius strictly smaller than $\frac{1}{2}$ means that two regions will never overlap (shown later on for Theorem 4.17). Repeatedly choosing a radius based on the ratio $\frac{w(C(S,r'))}{\hat{c}(S,r')}$ additionally allows us to obtain the approximation guarantee, because the distances in this extended graph are based on the solution of the LP (Theorem 4.19 below).

Algorithm 4: WDGS approximation algorithm

procedure SELECT($V, E, V', E', w, D_1, \ldots, D_n, l_1, \ldots, l_n$)
  solve LP from Definition 4.8 ▷ determine optimal fractional solution
  construct $G' = (V', E')$ ▷ create extended graph (Definition 4.9)
  $C \leftarrow \{e \in E \mid w(e) = 0\}$ ▷ cut zero-weighted edges
  $U_i \leftarrow \bigcup_{j=1}^{l_i - 1} D_{i,j} \quad \forall i : w(D_i) = 0$ ▷ remove zero-weighted D_i
  while $\exists i, j, k > j, u \in D_{i,j}, v \in D_{i,k} : \mathrm{P}(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}, E') \neq \emptyset$ **do**
    ▷ find an unsatisfied assertion
    $S \leftarrow \emptyset$ ▷ set of nodes around which regions will be grown
    for all j in $1 \ldots l_i - 1$ **do** ▷ arbitrarily choose node from each $D_{i,j}$
      if $\exists v \in D_{i,j} : \mathrm{v}_{i,v} \in V'$ **then** $S \leftarrow S \cup \mathrm{v}_{i,v}$
    $D \leftarrow \{d(u, v) \leq \frac{1}{2} \mid u \in S, v \in V'\} \cup \{\frac{1}{2}\}$ ▷ set of distances
    choose $\epsilon : \forall d, d' \neq d \in D : 0 < \epsilon \ll |d - d'|$ ▷ infinitesimally small
    $r \leftarrow \left\lceil \operatorname{argmin}_{r \in D \setminus \{0\}} \lim_{r' \to r^-} \frac{w(C(S, r'))}{\hat{c}(S, r')} \right\rceil - \epsilon$
    ▷ choose optimal radius (ties broken arbitrarily)
    $C' \leftarrow C(S, r)$ ▷ set of chosen cut edges
    $V' \leftarrow V' \setminus R(S, r)$ ▷ remove chosen regions from G'
    $E' \leftarrow \{e \in E' \mid e \subseteq V'\}$
    $C \leftarrow C \cup (C' \cap E)$ ▷ update global solution (C)
    for all i' in $1 \ldots n$ **do** ▷ update global solution (U_i)
      $U_{i'} \leftarrow U_{i'} \cup \{v \mid (\mathrm{v}_{i',v}, v) \in C'\}$
    for all i' in $1 \ldots n$ **do** ▷ prune distinctness assertions
      for all j in $1 \ldots l_{i'}$ **do**
        $D_{i',j} \leftarrow D_{i',j} \cap V'$
  return $(C, U_1, \ldots, U_n)$

4.5.2 Properties

Correctness of Algorithm

For proving the correctness, we first establish the relationship between the linear program and the WDGS objective.

Lemma 4.15 *The linear program given by Definition 4.8 enforces that for any i,j,$k \neq j$,$u \in D_{i,j}$, $v \in D_{i,k}$, and any path $v_0, \ldots, v_t$ with $v_0 = u$, $v_t = v$ we obtain*

$$u_{i,u} + \sum_{l=0}^{t-1} d_{(v_l, v_{l+1})} + u_{i,v} \geq 1.$$

Proof. Without loss of generality, let us assume that $j < k$. The LP constraints give us

$$\begin{aligned} s_{i,j,v_t} &\leq s_{i,j,v_{t-1}} + d_{(v_{t-1},v_t)} \\ \ldots &\leq \ldots \\ s_{i,j,v_1} &\leq s_{i,j,v_0} + d_{(v_0,v_1)} \end{aligned}$$

as well as $s_{i,j,v_0} = u_{i,u}$ and $s_{i,j,v_t} + u_{i,v} \geq 1$. Hence

$$1 \leq s_{i,j,v_t} + u_{i,v} \leq u_{i,u} + \sum_{l=0}^{t-1} d_{(v_l,v_{l+1})} + u_{i,v}.$$

□

Lemma 4.16 *The integer linear program obtained by augmenting Definition 4.8 with integer constraints $d_e, u_{i,v}, s_{i,j,v} \in \{0,1\}$ (for all applicable e, i, j, v) produces optimal solutions $(C, U_1, \ldots, U_k)$ for* WDGS *problems, obtained as $C = \{e \in E \mid d_e = 1\}$, $U_i = \{v \mid u_{i,v} = 1\}$.*

Proof. Lemma 4.15 implies that, with added integrality constraints, we obtain either $u \in U_i$, $v \in U_i$, or at least one edge along any path from u to v is cut, i.e. $\mathrm{P}(u, v, E \setminus C) = \emptyset$. This proves that any ILP solution induces a valid WDGS solution (Definition 4.3).

Clearly, the integer program's objective function minimizes the cost $c(C, U_1, \ldots, U_n)$ (Definition 4.4) if $C = (\{e \in E \mid d_e = 1\}, U_i = \{v \mid u_{i,v} = 1\}$. To see that the solutions are optimal, it thus suffices to observe that any optimal WDGS solution $(C^*, U_1^*, \ldots, U_n^*)$ yields a feasible ILP solution $d_e = \mathbf{1}_{C^*}(e)$, $u_{i,v} = \mathbf{1}_{U_i^*}(v)$ (where $\mathbf{1}_S$ is the indicator function for a set S). □

This means that by solving the LP, we obtain an optimal fractional solution to the LP relaxation of our actual objective.

Theorem 4.17 (Correctness) *The algorithm yields a* valid WDGS *solution $(C, U_1, \ldots, U_n)$.*

Proof. Clearly, $r < \frac{1}{2}$ holds for any radius r chosen by the algorithm, so for any region $R(v_0, r)$ grown around a node v_0, and any two nodes u, v within that region, the triangle inequality gives us $d(u, v) \leq d(u, v_0) + d(v_0, v) < \frac{1}{2} + \frac{1}{2} = 1$ (maximal distance condition).

At the same time, by Lemma 4.15 and Definition 4.9, the LP ensures that for any $u \in D_{i,j}$, $v \in D_{i,k}$ ($j \neq k$), we obtain

$$d(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}) = d(\mathrm{v}_{i,u}, u) + d(u, v) + d(v, \mathrm{v}_{i,v}) \geq 1.$$

With the maximal distance condition above, this means that $\mathrm{v}_{i,u}$ and $\mathrm{v}_{i,v}$ cannot be in the same region. Hence u, v cannot be in the same region, unless the edge from $\mathrm{v}_{i,u}$ to u is cut (in which case u will be placed in U_i) or the edge from v to $\mathrm{v}_{i,v}$ is cut (in which case v will be placed in U_i). Since each region is separated from other regions via C, we obtain that $\forall i, j, k \neq j, u, v$: $u \in D_{i,j} \setminus U_i$, $v \in D_{i,k} \setminus U_i$ implies $\mathrm{P}(u, v, E \setminus C) = \emptyset$, so a valid solution is obtained. □

Approximation Guarantee

For the approximation guarantee, we need the following lemma, which is essentially due to Avidor and Langberg (2007) and based on ideas by Garg et al. (1996):

Lemma 4.18 *For any i where $\exists j, k > j, u \in D_{i,j}, v \in D_{i,k} : \mathrm{P}(\mathrm{v}_{i,u}, \mathrm{v}_{i,v}, E') \neq \emptyset$ and $w(D_i) > 0$, there exists an r such that*

$$w(C(S, r)) \leq 2 \ln(nq + 1)\, \hat{c}(S, r)$$

and $0 \leq r < \frac{1}{2}$ for any set S consisting of $\mathrm{v}_{i,v}$ nodes from different $D_{i,j}$.

Proof. Define $w(S, r) = \sum_{v \in S} w(C(v, r))$. We will prove that there exists an appropriate r with

$$w(C(S, r)) \leq w(S, r) \leq 2 \ln(nq + 1)\, \hat{c}(S, r).$$

Assume, for reductio ad absurdum, that

$$\forall r \in [0, \frac{1}{2}) : w(S, r) > 2 \ln(nq + 1) \hat{c}(S, r).$$

As we expand the radius r, we note that

$$\hat{c}(S, r) \frac{d}{dr} = \sum_{v \in S} \sum_{e \in C(v,r)} w(e) = \sum_{v \in S} w(C(v, r)) = w(S, r)$$

wherever $\hat{c}$ is differentiable with respect to r. There are only a finite number of points $d_1,\ldots,d_{l-1}$ in $(0, \frac{1}{2})$ where this is not the case (namely, when $\exists u \in S, v \in V' : d(u,v) = d_i$). Also note that $\hat{c}$ increases monotonically for increasing values of r, and that it is universally greater than zero (since there is a path between $\mathrm{v}_{i,u}$, $\mathrm{v}_{i,v}$). Set $d_0 = 0$, $d_l = \frac{1}{2}$ and choose ϵ such that $0 < \epsilon \ll \min\{d_{j+1} - d_j \mid j < l\}$. Our assumption then implies:

$$\sum_{j=1}^{l} \int_{d_{j-1}+\epsilon}^{d_j-\epsilon} \frac{w(S,r)}{\hat{c}(S,r)}\, dr > \sum_{j=1}^{l} (d_j - d_{j-1} - 2\epsilon)\, 2\ln(nq+1).$$

This in turn entails the following:

$$\begin{aligned}
\sum_{j=1}^{l} (\ln \hat{c}(S, d_j - \epsilon) - \ln \hat{c}(S, d_{j-1} + \epsilon)) &> \left(\frac{1}{2} - 2l\epsilon\right) 2\ln(nq+1) \\
\ln \hat{c}(S, \frac{1}{2} - \epsilon) - \ln \hat{c}(S, 0) &> (1 - 4l\epsilon)\ln(nq+1) \\
\frac{\hat{c}(S, \frac{1}{2} - \epsilon)}{\hat{c}(S,0)} &> (nq+1)^{1-4l\epsilon} \\
\hat{c}(S, \frac{1}{2} - \epsilon) &> (nq+1)^{1-4l\epsilon} \hat{c}(S,0).
\end{aligned}$$

For small ϵ, the right term can get arbitrarily close to

$$(nq+1)\hat{c}(S,0) = nq\,\hat{c}(S,0) + \hat{c}(S,0) \geq \hat{c}_{\mathrm{f}} + \hat{c}(S,0),$$

which is strictly larger than $\hat{c}(S, \frac{1}{2} - \epsilon)$ no matter how small ϵ becomes. However, we cannot have $\hat{c}(S, \frac{1}{2} - \epsilon) > (nq+1)^{1-4l\epsilon}\hat{c}(S,0)$ if $(nq+1)^{1-4l\epsilon}\hat{c}(S,0)$ can come arbitrarily close to a value *strictly* larger than $\hat{c}(S, \frac{1}{2} - \epsilon)$, so the initial assumption is false. □

With this lemma, we can then prove the following theorem.

Theorem 4.19 (Approximation Guarantee) *The algorithm yields a solution $(C, U_1, \ldots, U_n)$ with an approximation factor of $4\ln(nq+1)$ with respect to the cost of the optimal* WDGS *solution $(C^*, U_1^*, \ldots, U_n^*)$, where n is the number of distinctness assertions and $q = \max_{i,j} |D_{i,j}|$. This solution can be obtained in polynomial time.*

Proof. Let S_i, r_i denote the set S and radius r chosen in particular iterations, and let c_i denote the corresponding costs incurred:

$$\begin{aligned} c_i &= w(C(S_i, r_i)) \\ &= w(C(S_i, r_i) \cap E) + \sum_{i'=1}^{n} w(D_{i'}) \, |\{v \mid (\mathrm{v}_{i',v}, v) \in C(S_i, r_i)\}| \end{aligned}$$

Note that the cut $C(S_i, r_i)$ for any radius r_i chosen by the algorithm will in fact correspond to the cut $C(S_i, r)$ for a radius r that fulfils the criterion described by Lemma 4.18. This is because $C(S_i, r)$ and $w(C(S_i, r))$ only change at points r in D, so points $r \in [0, \frac{1}{2})$ that minimize the ratio between the two terms are reached by approaching points in D from the left. Hence, we obtain $c_i \leq 2\ln(nq+1)\hat{c}(S_i, r_i)$.

For our global solution, note that there is no overlap between the regions chosen within an iteration, since regions have a radius strictly smaller than $\frac{1}{2}$, while $\mathrm{v}_{i,u}$, $\mathrm{v}_{i,v}$ for $u \in D_{i,j}$, $v \in D_{i,k}$, $j \neq k$ have a distance of at least 1. Nor is there any overlap between regions from different iterations, because in each iteration the selected regions are removed from G'. Globally, we therefore obtain (observe that $i \leq nq$):

$$\begin{aligned} c(C, U_1, \ldots, U_n) &= \sum_i c_i \\ &< 2\ln(nq+1) \sum_i \hat{c}(S_i, r_i) \\ &= 2\ln(nq+1) \sum_i \left[\left[\sum_{v \in S_i} \hat{c}(v, r_i) \right] + \frac{1}{nq} \hat{c}_\mathrm{f} \right] \\ &\leq 2\ln(nq+1) 2\hat{c}_\mathrm{f}. \end{aligned}$$

Since $\hat{c}_\mathrm{f}$ is the objective score for the fractional LP relaxation solution of the WDGS ILP (Lemma 4.16), we know that $\hat{c}_\mathrm{f} \leq c(C^*, U_1^*, \ldots, U_n^*)$, and thus

$$c(C, U_1, \ldots, U_n) < 4\ln(nq+1)\, c(C^*, U_1^*, \ldots, U_n^*).$$

To obtain a solution in polynomial time, note that the LP size is linear with respect to n,q and may be solved using a polynomial-time algorithm (Karmarkar, 1984). The subsequent steps run in no more than nq iterations in the worst case. In each iteration, we grow up to $|V|$ regions. The argmin can be computed efficiently in $O((|E| + |V|) \log |V|)$

steps by evaluating radiuses corresponding to distances of nearest neighbours with respect to the distance metric, as will be explained in Section 4.6.1. □

This guarantee is a nice property, as it shows that although our algorithm is not exact, the results will still be within certain bounds with respect to an ideal output given the input information.

4.6 Results

In addition to the theoretical properties, we also evaluate the practical behaviour of this algorithm in our particular setting, i.e. for the task of cleaning up equality links in conjunction with distinctness information about entities derived from Wikipedia.

4.6.1 System Architecture

The overall system is based on the same framework as used for Chapter 3. Equality information can come from the original data sources themselves when they are imported as graph-structured knowledge bases. This is the case for Wikipedia's cross-lingual links. Alternatively, heuristic mappers can be invoked to infer new `equals` connections.

The algorithm then operates on the union of these knowledge bases, processing one connected component at a time. Each original connected component turns into one or more connected components in the output. While the input components may conflate distinct concepts or entities, we expect each output component to cleanly represent a single concept or entity.

The linear program solving is one of the main bottlenecks of the algorithm. A fast LP solver is crucial, and making the right choice can lead to speed-ups of several orders of magnitude. In our experiments, we used the well-known commercial tool CPLEX. Even CPLEX, however, in rare cases seemed to have trouble coping with certain inputs, so for large subgraphs, we resorted to invoking CPLEX as an external process, which is automatically killed if CPLEX is unable to find a solution within a specific time frame. In Section 4.6.5 below, we explain how one can proceed if this case occurs.

During the region growing, the argmin of points r in D can be determined by iteratively visiting nearest neighbours of nodes in S with respect to the distance metric d in the extended graph. A priority queue can be used to keep track of the nearest unvisited neighbours. If this queue is initialized with nodes in S at radius 0, then a simple uniform-cost search will find the nearest neighbours that can be added to individual regions as the radius expands. At a radius r corresponding to a given neighbour, we evaluate $\lim_{r' \to r^-} \frac{w(C(S,r'))}{\hat{c}(S,r')}$. It is essential to note that this is a one-sided limit from the left, so $\lim_{r' \to r^-} w(C(S,r'))$ is not equal to $w(C(S,r))$ but rather to $w(C(S,r-\epsilon))$ for any strictly positive value ϵ that is smaller than $|d - d'|$ for $d, d' \neq d \in D$.

Since the algorithm is applied to a single connected component at a time, additional speed-ups are possible by parallelizing the processing. For each individual connected component, one can also make use of the parallel processing capabilities of recent versions of CPLEX.

4.6.2 Datasets

We downloaded XML dumps of all available editions of Wikipedia as of February 2010, in total 272 editions that amount to 86.5 GB uncompressed. From these dumps we produced two datasets. Dataset A captures cross-lingual interwiki links between pages, in total 77.07 million undirected edges (146.76 million original links). Dataset B additionally includes 2.2 million edges derived from redirects, as described in Section 4.3.

4.6.3 Application of Algorithm

The choice of distinctness assertion weights depends on how lenient we wish to be with regard to conceptual drift. Lower weights mean that the algorithm can liberally remove nodes from distinctness assertions and produce coarse-grained semantic entities, while higher weights lead to more fine-grained distinctions. Since Wikipedia editions rarely contain genuine duplicates and since we envision an output resource that reflects even subtle differences between semantic entities, we settled on a weight of 100 in the following experiments.

We analysed over 20 million connected components in each dataset, checking for distinctness assertions. For the roughly 110,000 connected components with relevant distinctness assertions, we applied our algorithm, relying on the commercial CPLEX tool to solve the linear programs. In most cases, the LP solving took less than a second, however the LP sizes quickly grow with the size of the graph. In about 300 cases per dataset, CPLEX took too long and was automatically killed or the linear program was a priori deemed too large to complete in a short amount of time. In such circumstances, we adopted an alternative strategy described later on.

Table 4.1 provides the experimental results for the two datasets. Dataset B is more connected and thus has fewer connected components with more pairs of nodes asserted to be distinct by distinctness assertions. The LP given by Definition 4.8 provides fractional solutions that constitute lower bounds on the optimal solution (as shown by Lemma 4.18), so the optimal solution cannot have a cost lower than the fractional LP solution. Table 4.1 shows that in practice, our algorithm achieves near-optimal results.

Table 4.1: Algorithm results

	Dataset A	Dataset B
Connected components	23,356,027	21,161,631
– with distinctness assertions	112,857	113,714
– algorithm applied successfully	112,580	113,387
Distinctness assertions	380,694	379,724
Node pairs considered distinct	916,554	1,047,299
Lower bound on optimal cost	1,255,111	1,245,004
Cost of our solution	1,306,747	1,294,196
Factor	1.04	1.04
Edges to be deleted (undirected)	1,209,798	1,199,181
Nodes to be merged	603	573

4.6.4 Result Quality

The near-optimal results of our algorithm apply with respect to our problem formalization, which aims at repairing the graph in a minimally invasive way. It may happen, however, that the graph's topology is misleading, and that in a specific case deleting many edges to separate two entities is more appropriate than looking for a conservative way to separate them. Figure 4.9 depicts a graph in which the Spanish `Televisión` seems to be more tightly integrated with nodes describing TV sets, but in reality, `Televisión` describes television as a medium and belongs in a separate component together with `en:Television`.

For this reason, we additionally studied the quality of the output from a semantic perspective. From Dataset A, we randomly sampled 200 pairs of nodes, consisting of an English and a German article, that were originally in the same connected component but separated into separate ones by our algorithm. The English and German Wikipedia editions are the two largest ones, so this is a particularly difficult case, as they are well-maintained and distinctness assertions often stem from very subtle semantic differences rather than from links that are completely erroneous. Examples are given in Table 4.2. The random sample was evaluated by two annotators with an inter-annotator agreement (Cohen κ) of 0.656. We obtained a precision of $87.97\% \pm 0.04\%$ against the consensus annotation, using a Wilson score interval at $\alpha = 0.05$ (Brown et al., 2001). The majority of incorrect pairs appear to have resulted from articles having large numbers of inaccurate outgoing links, often due to automated bots operating on Wikipedia. In these cases, entities may be assigned to the wrong component. In other cases, we noted duplicate articles in Wikipedia, or cases where a single Wikipedia article would actually describe two related concepts on the same page. Finally, the use of uniform edge weights in Section 4.3 means that the algorithm in some cases lacks information on which it could base its decision. This issue could be resolved by using edge weights biased to reflect entity similarities or trust scores for different knowledge sources.

4.6.5 Large Problem Instances

Partitioning. When problem instances become too large, the linear programs can become unwieldy for current linear optimization software.

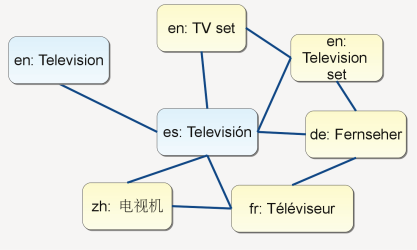


Figure 4.9: Misleading graph topology

Fortunately, in such cases, the graphs tend to be very sparsely connected, consisting of many smaller, more densely connected subgraphs. The reason for this is that a single spurious link is enough to turn two separate subgraphs into a single connected component. We thus investigated graph partitioning heuristics to decompose larger graphs into smaller parts that the LP solver could more easily cope with.

The METIS algorithms (Karypis and Kumar, 1998) can partition graphs with hundreds of thousands of nodes almost instantly, but favour equally sized clusters over lower cut costs. We obtain partitionings with costs orders of magnitude lower using the heuristic by Dhillon et al. (2007). We then run our WDGS algorithm on each individual partition. Unbalanced partitionings can in principle contain large partitions that remain too large to handle. In such cases, we can recursively apply the same partitioning heuristic to obtain even smaller partitions, and then run our WDGS algorithm on these.

Database of Named Entities. These partitioning heuristics allow us to process all entries in the complete set of Wikipedia dumps and produce a clean output set of connected components where each Wikipedia article or category belongs to a connected component consisting of pages about the same entity or concept. We can regard these connected components as equivalence classes. This means that we obtain a large-scale multilingual database of named entities and their translations.

Table 4.2: Examples of separated concepts

English concept	German concept (translated)	Explanation
Compulsory education	Right to education	duty vs. right
Associação Académica de Coimbra – O.A.F.	University of Coimbra	the former is a football organization
Nursery school	Pre-school education	the latter is more general
Mittlere Reife	GCSE	the former is a German degree
Coffee percolator	French Press	different types of brewing devices
Franz Kafka's Diaries	Franz Kafka	diaries vs. person
Baqa-Jatt	Baqa al-Gharbiyye	Baqa-Jatt is a city resulting from a merger of Baqa al-Gharbiyye and Jatt
White tiger	White tiger (Constellation)	the latter refers to the Chinese constellation symbol
Leucothoe (plant)	Leucothea (Orchamos)	the latter refers to a figure of Greek mythology
Old Belarusian language	Ruthenian language	the latter is often considered slightly broader
Sliding puzzle	Fifteen puzzle	the latter is a specific form of sliding puzzle
Grand Staircase-Escalante National Monument	Calf Creek Canyon	the latter is located in the former
Torre Sant Sebastià	Port Vell Aerial Tramway	the former is a terminal of the tramway
Multicore cable	Multicore processor	different types of objects

We are also able to more safely transfer information cross-lingually between Wikipedia editions. For example, when an article a has a category c in the French Wikipedia, we can suggest the corresponding Indonesian category for the corresponding Indonesian article.

Later on, in Chapter 5, we shall see how such a multilingual database of named entities can be used to create a multilingual taxonomy, where even entirely non-English connected components can in many cases be assigned a class in WordNet. So, the German Wikipedia article on the educational TV series '*Galileo*', despite the lack of a corresponding English article, can be assigned the WordNet synset for television and radio series.

4.6.6 Case Study: Language Information

Language Entities. Semantic entities corresponding to individual human languages are of particular interest in a multilingual knowledge base. In Figure 2.3 on page 23, we showed how one can link each term to its respective language using the `language` relation. If there is additional knowledge about these languages, we can answer queries like:

- Which words for '*student*' exist in languages spoken in Asia?
- Which words for '*mouse*' exist in Indo-European languages?

WordNet and Wikipedia already contain identifiers for several hundred languages and language families. However, in a multilingual knowledge base, it is beneficial to have a more complete register of the world's languages, based on international standards like ISO 639-3, which describes over 7,000 languages. Similar standards exist for language families, writing systems (e.g. Cyrillic, Devanagari, and Hangul) and geographical regions. We can integrate such entities into a combined knowledge base.

Knowledge Sources. The input graph's node set contained entity identifiers for languages, language families, geographical regions, and writing systems from the following sources apart from Wikipedia and WordNet:

- the ISO 639-3 specification[1], which defines codes for around 7,000 languages and lists relationships between macrolanguages and individual languages,
- the ISO 639-5 specification[2], which describes a limited number of language families (e.g. Tai languages) and other collections (e.g. sign languages),
- the ISO 15924 specification[3], which lists a number of writing systems, e.g. Cyrillic, Devanagari, and Hangul,
- the Ethnologue language codes database (Lewis, 2009), which provides additional language names, geographical regions where languages are spoken, etc.,

[1] `http://www.sil.org/iso639-3/`
[2] `http://www.loc.gov/standards/iso639-5/`
[3] `http://unicode.org/iso15924/`

- the Linguist List[4], which contributes information on extinct languages as well as constructed languages,
- the Unicode Common Locale Data Repository[5] (CLDR), which connects languages to their geographical regions and writing systems, and delivers names in many languages.

Input Edges. The nodes in the graph were connected as follows.

- The official ISO 639-3 mapping tables allowed us to connect language identifiers based on ISO 639 Part 1 or 2 to ISO 639-3 identifiers.
- Wikipedia's languages were linked to languages from ISO 639-3 by extracting the codes from the respective articles in Wikipedia.
- Wikipedia's language families were linked to corresponding language families from ISO 639-5 where possible, by extracting links from Wikipedia's '*List of ISO 639-5 codes*' article.
- The same article also provided equivalences between ISO 639-5 and ISO 639-3.
- Languages from WordNet and Wikipedia were matched using heuristic mapping scores (de Melo and Weikum, 2010c).

Result. We added distinctness assertions between ISO 639-3 codes, between WordNet synsets, and between Wikipedia articles. Our algorithm then ensures that the output components are consistent, e.g. to prevent Modern Greek and Ancient Greek from being conflated. The result is a knowledge base where information from different knowledge sources with different sets of entity identifiers has been consolidated. This leads to a domain-specific extension of WordNet describing over 7000 languages rather than just the original 600 ones listed in WordNet. Additional factual knowledge is associated with each language, e.g. where a language is spoken and what writing systems are used. More details and results are given in a separate publication (de Melo and Weikum, 2010c).

[4] `http://linguistlist.org/`
[5] `http://cldr.unicode.org/`

4.7 Discussion

In this chapter, we have presented an algorithmic framework that addresses the problem of entity integration given weighted equivalence links and distinctness assertions. Pre-existing or heuristically derived coreference information from multiple knowledge sources is represented in a weighted undirected graph, and additional weighted distinctness assertions are made. Our method reconciles conflicting information by intelligently choosing between removing edges and allowing nodes to remain connected. Our algorithm produces consistent connected components of co-referring entities. In addition to having a logarithmic approximation guarantee, the algorithm also shows excellent results in practice. It has successfully been applied to Wikipedia's cross-lingual interwiki link graph, where we identified and eliminated surprisingly large numbers of inaccurate connections, leading to a large-scale multilingual register of named entities.

Additionally, our approach is flexible enough to apply to a wide range of other scenarios. For instance, one could use heuristics to connect isolated, unconnected Wikipedia articles to likely candidates in other Wikipedias using weighted edges. One could resolve entity integration issues on the Linked Data Web, where there are well-known problems with the de facto standard of using `owl:sameAs` to indicate coreference even when mappings are not ontologically precise (Halpin and Hayes, 2010). Heuristic mappings between additional thesauri or sense clusterings induced from text could be included in the input, with the hope that the weights and link structure will then allow the algorithm to make the final disambiguation decision.

Unlike most previous work on entity integration, our algorithm can draw on distinctness information to combine more than just two knowledge sources. In the next chapter, we demonstrate, among other things, that our algorithm can be used to integrate WordNet with multiple editions of Wikipedia.

CHAPTER 5

Taxonomic Integration

5.1 Introduction

In order to put everything together into a full-fledged knowledge base with a well-structured organization, we finally turn to taxonomic integration as the third and final major building block. The techniques from Chapter 4 allow us to establish a unified repository of entities based on multiple sources, where equivalent entities are cleanly linked together by `equals` arcs. In this chapter, we explain how entities that are not equivalent can be related to one another in terms of semantic relations like `instance` and `subclass`. For example, an individual named entity like `Fersental` could be described as an `instance` of `Valleys in Italy`, and `Valleys in Italy` could be a `subclass` of `Valley`. A coherent taxonomic class hierarchy would give us a global hierarchical organization that connects all entities in the knowledge base, even if they originate from different multilingual editions of Wikipedia.

Motivation. If a user is searching for institutes of higher education in Europe, it would be helpful to have access to the fact that a Portuguese '*ensino politécnico*' or a German '*Technische Hochschule*' qualifies. Even better, an application may have such ontological information about specific institutes like the Royal College of Art, as in our example in Chapter 1.

We could try to derive taxonomic information from text corpora (Hearst, 1992), machine-readable dictionaries (Chodorow et al., 1985), or search engine query logs (Baeza-Yates and Tiberi, 2007), but once again, Wikipedia turns out to be a very useful resource that not only provides fairly reliable input signals but also richer content. For example, for each entity we can additionally obtain gloss descriptions or perhaps factual statements expressing geographical locations of places. Wikipedia has previously been exploited by projects like DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), and WikiTaxonomy (Ponzetto and Strube, 2008). To date, however, these extraction efforts have largely neglected the significant potential of Wikipedia's multilingual nature. While DBpedia and some other knowledge bases do extract abstracts and other information also from non-English versions, the coverage is still restricted to those entities that have a corresponding article in the English Wikipedia. Certainly, the English Wikipedia is by far the most comprehensive version. Yet, its articles make up only 24% among those of the 50 largest Wikipedias. This means that there are large amounts of untapped information that could be formalized in machine-readable form.

This, however, leads not only to great opportunities but also to new research challenges. In particular, it is not clear how these different information sources can be brought together to form a unified, coherent resource. In this chapter, we aggregate from multiple editions of Wikipedia as well as WordNet to construct MENTA – Multilingual Entity Taxonomy – a large-scale taxonomic knowledge base that covers a significantly greater range of entities than previous knowledge bases. Additionally, MENTA enables tasks like semantic search also in languages other than English, for which existing taxonomies are often very limited or entirely non-existent. Finally, we also hope that MENTA will facilitate decidedly multilingual applications like cross-lingual information retrieval (Etzioni et al., 2007; Bellaachia and Amor-Tijani, 2008), machine translation (Knight and Luk, 1994), or learning transliterations (Pasternack and Roth, 2009).

Problem Statement. As input we have a set of knowledge sources and a large but incomplete set of unreliable, weighted statements linking entities to parent entities (taxonomic links) or to equivalent entities

(equals arcs). For a given entity, we often have many candidate parents from different sources with different weights, and different parents may or may not be related to each other in terms of equals and subclass arcs (see Figure 5.2 for an example scenario).

The aim is to aggregate these unreliable, incomplete taxonomic links between entities from different sources into a single more reliable and coherent taxonomy. The output should be a clean, reliable knowledge base where the entities share a single upper-level core rather than having a diverse range of separate taxonomies. Schematically, this task is depicted in Figure 5.1.

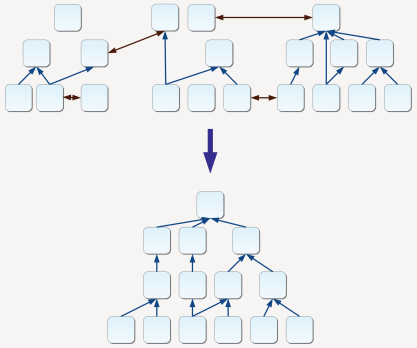


Figure 5.1: Taxonomic integration strategy

Contribution. We describe an algorithm called Markov Chain Taxonomy Induction that solves this central problem. Additionally, we present a complete framework that starts out with all editions of Wikipedia as well as WordNet and ties everything together. The input to the algorithm is supplied by a set of heuristic linking functions that connect Wikipedia articles in multiple languages, categories, so-called infobox templates, and WordNet synsets. The algorithm produces aggregated rankings of parents that take into account the dependencies between the linked entities. The output for a specific entity is given by the stationary distribution of a Markov chain, in the spirit of PageRank, but adapted

to our specific setting. Overall, this leads to MENTA having three major distinguishing properties.

1. Extended Coverage of Entities: The taxonomy draws on all existing editions of Wikipedia and hence includes large numbers of local places, people, products, etc. that are not covered by the English Wikipedia. For example, the Quechua Wikipedia has an article about the Bolivian salt lake Salar de Coipasa, and the Czech Wikipedia has an article about the French academic degree DESS.

2. Ranked Class Information: Individual entities are linked via `instance` statements to classes (e.g. `University`, `City`, `Airline company`, etc.) based on information provided by multiple Wikipedia editions, thus exploiting complementary clues from different languages. The output is a ranked list, because even when e.g. an English article provides ample information, it is useful to capture that the Colorado River being a river is more salient than it being a border of Arizona.

3. Coherent Taxonomy: While Wikipedia is an excellent source of semi-structured knowledge about entities, it lacks an ontologically organized taxonomy. The category systems of Wikipedia i) fail to distinguish classes from topic labels (the Free University of Bozen-Bolzano is a `University` but not a `Bolzano`, Jean Piaget is a `Developmental psychologist` but not a `Child development`), ii) tend to lack a clear organization especially at the abstract upper level, and iii) differ substantially between different languages. A single, more complete yet coherent taxonomic class hierarchy is obtained by aggregating information from multiple editions of Wikipedia and WordNet.

The resulting taxonomy in MENTA goes beyond what is offered by previous semantic knowledge repositories. For instance, DBpedia and YAGO do not have a multilingual upper-level ontology. None of the previous taxonomies have managed to accommodate culture-specific entities from non-English Wikipedia editions. Even for those entities that are covered, the DBpedia Ontology provides class information only for around a third. Likewise, in the field of multilingual taxonomies and hierarchically-organized lexical knowledge bases, our knowledge base surpasses all previous resources in the number of entities described. MENTA is freely available under an open-source license from `http://www.mpi-inf.mpg.de/yago-naga/menta/`.

Overview. The rest of this chapter is organized as follows. Section 5.2 begins with a description of previous knowledge bases and approaches. Section 5.3 lays out how information is extracted from the input knowledge sources and represented in a form amenable to further processing in our approach. Section 5.4 then introduces the heuristics that are used to interlink entities and provide the input for the taxonomy induction step. Section 5.5 describes the Markov Chain Taxonomy Induction algorithm that produces the output knowledge bases with unified taxonomic class hierarchies. Section 5.6 evaluates this algorithm and the resulting knowledge bases. Finally, Section 5.7 provides a concluding discussion of these results.

5.2 Previous Work

Mining Wikipedia. A number of projects have imported basic information from Wikipedia, e.g. translations and categories (Kinzler, 2008; Silberer et al., 2008), or simple facts like birth dates, e.g. in the Freebase project (Bollacker et al., 2008). Such resources lack the semantic integration of conflicting information as well as the taxonomic backbone that is the focus of our work.

Apart from such facts, DBpedia (Auer et al., 2007) also provides an ontology, based on a set of manually specified mappings from Wikipedia's infobox templates to a coarse-grained set of 260 classes. However, the majority of English articles do not have any such infobox information, and non-English articles without English counterparts are simply ignored. DBpedia additionally includes class information from YAGO (Suchanek et al., 2007), a knowledge base that links entities from Wikipedia to an upper-level ontology provided by WordNet. We adopted this idea of using WordNet as background knowledge as well as some of the heuristics for creating instance and subclass arcs. YAGO's upper ontology is entirely monolingual, while in MENTA the class hierarchy itself is also multilingual and additionally accommodates entities that are found in non-English Wikipedias. Furthermore, the class information is simultaneously computed from multiple editions of Wikipedia. Nastase et al. (2010) exploit categories not only to derive `isA` relationships, but also to uncover other types of relations, e.g. a category like '*Universities in Milan*' also reveals where a university is located.

Linking Heuristics. There are other projects that have proposed heuristics for interlinking Wikipedia editions or linking Wikipedia to WordNet. Ponzetto and Strube (2008) and Ponzetto and Navigli (2009) studied heuristics and strategies to link Wikipedia categories to parent categories and to WordNet. Their results are significant, as they lead to a taxonomy of classes based on the category system of the English Wikipedia, however they did not study how to integrate individual entities (articles) into this taxonomy.

Recently, Navigli and Ponzetto (2010) investigated matching English Wikipedia articles with WordNet synsets by comparing the respective contextual information, obtaining a precision of 81.9% at 77.5% recall. Wu and Weld (2008) use parsing and machine learning to link infobox templates to WordNet. The Named Entity WordNet project (Toral et al., 2008) attempts to link entities from Wikipedia as instances of roughly 900 WordNet synsets. Others examined heuristics to generate new cross-lingual links between different editions of Wikipedia (Oh et al., 2008; Sorg and Cimiano, 2008).

The focus in our work is on a suitable algorithmic framework to aggregate and rank information delivered by such heuristics, and many of these heuristics could in fact be used as additional inputs to our algorithm. The same holds for the large body of work on information extraction to find `isA` relationships in text corpora (Hearst, 1992; Snow et al., 2004; Etzioni et al., 2004; Garera and Yarowsky, 2008), machine-readable dictionaries (Montemagni and Vanderwende, 1992), or search engine query logs (Baeza-Yates and Tiberi, 2007). Adar et al. (2009) and Bouma et al. (2009) studied how information from one Wikipedia's infoboxes can be propagated to another edition's articles, which is distinct from the problem we are tackling.

Multilingual Knowledge Bases. Concerning multilingual knowledge bases in general, previous results have been many orders of magnitude smaller in terms of the number of entities covered (Knight and Luk, 1994; Fellbaum and Vossen, 2007), or lack an ontological class hierarchy (Mausam et al., 2009). EuroWordNet (Vossen, 1998) provides multilingual labels for many general words like '*university*', but lacks the millions of individual named entities (e.g. '*Napa Valley*' or '*San Diego Zoo*') that Wikipedia provides.

Taxonomy Induction Algorithms. Hierarchical agglomerative clustering has been used to derive monolingual taxonomies (Klapaftis and Manandhar, 2010), however clustering techniques will often merge concepts based on semantic relatedness rather than specific ontological relationships. Our work instead capitalizes on the fact that reasonably clean upper ontologies already exist, so the main challenge is integrating the information into a coherent whole. There are numerous studies on supervised learning of hierarchical classifications (Dumais and Chen, 2000), but such approaches would require reliable training data for each of the several hundred thousand classes that we need to consider. Another interesting alternative approach, proposed by Wu and Weld (2008), relies on Markov Logic Networks to jointly perform mappings between entities and derive a taxonomy. Unfortunately, such techniques do not scale to the millions of entities we deal with in our setting.

Snow et al. (2006) proposed a monolingual taxonomy induction approach that considers the evidence of coordinate terms when disambiguating. Their approach assumes that evidence for any superordinate candidates is directly given as input, while our approach addresses the question of how to produce evidence for superordinate candidates based on evidence for subordinate candidates. For instance, very weak evidence that Stratford-upon-Avon is either a village or perhaps a city may suffice to infer that it is a populated place. Talukdar et al. (2008) studied a random walk technique to propagate class labels from seed instances to other coordinate instances, but did not consider hierarchical dependencies between classes. Ponzetto and Navigli (2009) proposed a method to restructure a taxonomy based on its agreement with a more reliable taxonomy (WordNet), but do not address how to integrate multiple taxonomies.

Markov Chains. Our Markov Chain Taxonomy Induction algorithm is most similar to PageRank with personalized random jump vectors (Page et al., 1999; Haveliwala, 2002); however our transition matrix is based on statement weights, and the probability for jumping to a start node of a random walk depends on the weights of the alternative statements rather than being uniform for all nodes. Uniform weights mean that single parents are visited with very high probability even if they are only very weakly connected, while in our approach such irrelevant parents

will not obtain a high transition probability. Other studies have relied on PageRank to find important vocabulary in an ontology (Zhang et al., 2006) and to perform word sense disambiguation (Mihalcea et al., 2004). Our Markov chain model differs from these in that we aim at identifying salient parents for a specific node rather than generic random walk reachability probabilities. We are not aware of any Markov chain-based approaches for constructing class hierarchies.

5.3 Knowledge Extraction

5.3.1 Representation Model

We again rely on the knowledge representation framework from Chapter 2. The final set of entity identifiers to be used in MENTA is determined at a later step when the different knowledge sources are combined. During the initial extraction phase, while importing knowledge from existing sources, we instead start out with preliminary entity identifiers. These will again include semantic entity identifiers for Wikipedia pages (see below in Section 5.3.2), semantic entities based on the WordNet database's synsets, as well as term entities, i.e. string literals with language designators. The arc labels include:

- `equals`: identity or equivalence of entities (i.e. two entity identifiers refer to the same entity)
- `subclass`: the relation between a semantic entity and another semantic entity that is a subsuming generalization of the former one
- `instance`: the relation between an individual entity and another semantic entity it is an instance of (its class, type, or role)
- `means`: the meaning relationship between a term entity (a word or a name in a given language) and a semantic entity

A statement might express that the `University of Trento` stands in an `instance` relation to the entity `University` with confidence 1, or that the Polish name '*Trydent*' stands in a `means` relation to the city of Trento.

5.3.2 Extraction from Wikipedia

To a certain extent, the input we derive from Wikipedia will be similar to what we considered in Chapter 4. To obtain a more coherent knowledge base, we will additionally be considering equivalences between articles and WordNet synsets, categories, and infoboxes. Additionally, in order to obtain a richer knowledge base, we also extract lexical and other information from Wikipedia.

Entities. The preliminary entity identifiers used for the input graph are intended to represent the subjects of different items encountered in Wikipedia. In particular, each article page (including redirect pages), category page, or template page (including infobox templates) in an edition of Wikipedia becomes a node in our graph with a preliminary entity identifier. These are assigned while parsing the raw XML and wiki-markup-based Wikipedia dumps, extracting relevant information, and casting it into our representation model to facilitate further processing.

Unfortunately, not all information necessary for assigning canonical identifiers is available from within the dumps alone. We additionally query the Web services provided by each server to find out for instance that in the Tagalog Wikipedia, titles starting with "Kategorya:" refer to categories (in addition to the default "Kaurian:" and the English "Category:", which are also accepted). Such information is normalized, so as to obtain canonical entity identifiers. Being able to recognize categories is also helpful at a later stage when constructing the taxonomy.

Statements. Additional information about entities and meta-data about articles that may be of use later on is extracted and stored with appropriate relations. In particular, we capture template invocations, cross-lingual "interwiki" links, redirects, multimedia links, category links, and optional factual statements (`locatedIn`, `bornIn`, and so on).

Additionally, we create short description glosses for each article entity (`hasGloss`) by processing wikitext and HTML mark-up and attempting to identify the first proper paragraph in an article's wikitext mark-up (skipping infoboxes, pictures, links to disambiguation pages, etc.). If this first paragraph is too long, i.e. the length is greater than some l, a sentence boundary is identified in the vicinity of the position l.

Term Meanings. Article titles allow us to create `means` statements that link terms (words, labels, names) to the semantic entities they refer to. The original article title is modified by removing any additional qualifications in parentheses, e.g. '*School (discipline)*' becomes '*School*'. Some articles use special markup to provide the true capitalization of a title, e.g. '*iPod*' instead of '*IPod*'. If no markup is provided, we check for the most frequent capitalization variant within the article text.

5.4 Linking Functions

Given our goal of creating a single more coherent knowledge base from the different editions of Wikipedia and WordNet, our strategy will be to first expose possible connections between different nodes using several heuristics. After that, in a second step described later on in Section 5.5, we integrate these noisy inputs to induce a shared taxonomy.

For the first step, we rely on so-called linking functions to identify how different entities relate to each other. In particular, Section 5.4.1 introduces `equals` linking functions that identify identical entities, and Sections 5.4.2 and 5.4.3 present linking functions for the `subclass` and `instance` relations.

Definition 5.1 *A* linking function $l_r : V \times V \to \mathbb{R}_0^+$ *for a specific relation* $r \in \Sigma$ *is a function that yields confidence weight scores* $l_r(x, y) \in \mathbb{R}_0^+$ *and is used to produce statements* $(x, y, r, l_r(x, y))$ *for pairs of entity identifiers* x, y.

Given a set of `equals` linking functions L_{e}, a set of `subclass` linking functions L_{s}, and a set of `instance` linking functions L_{i}, Algorithm 5 shows how the input graph is extended with appropriate links. For each linking function $l \in L_{\mathrm{e}} \cup L_{\mathrm{s}} \cup L_{\mathrm{i}}$, we additionally assume we have a candidate selection function σ_l, which for a given node $x \in V$ yields a set $\sigma_l(x) \subseteq V$ containing all nodes y that are likely to have non-zero scores $l(x, y) > 0$.

Later on, we will explain how the output of somewhat unreliable linking functions can be aggregated to provide meaningful results. Which heuristics are appropriate for a given input scenario depends on the knowledge sources involved. We will now describe the specific choices

Algorithm 5: Linking function application

procedure CREATELINKS($G_0 = (V_0, A_0, \Sigma), L_\mathrm{e}, L_\mathrm{s}, L_\mathrm{i}, \{\sigma_l \mid l \in L_\mathrm{e} \cup L_\mathrm{s} \cup L_\mathrm{i}\}$)
  for all l in L_e **do** ▷ for each `equals` linking function
    $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \texttt{equals}, w = l(x, y)\}$
  for all l in L_s **do** ▷ for each `subclass` linking function
    $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \texttt{subclass}, w = l(x, y)\}$
  for all l in L_i **do** ▷ for each `instance` linking function
    $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \texttt{instance}, w = l(x, y)\}$
  return $G_0 = (V_0, A_0, \Sigma)$

of linking functions that we use to connect entities in different language-specific editions of Wikipedia as well as WordNet.

5.4.1 Equality Link Heuristics

In Chapter 4, we explored entity integration focussing only on Wikipedia. Here, we will be re-using the algorithm from Chapter 4, but will draw on a larger set of inputs. We use the following linking functions to generate `equals` arcs between two entity identifiers x, y.

Cross-Lingual Linking

Like in Chapter 4, if there is a cross-lingual interwiki link from x to y in Wikipedia, e.g. from `Trydent` in the Polish Wikipedia to `Trento` in the English one, the cross-lingual linking function yields 1, otherwise 0.

Category-Article Linking

The category-article linking function returns 1 when x, y correspond to a category and an article, respectively, known to be about the same concept, e.g. the category `Abugida writing systems` and the article `Abugida`. This is detected by checking for specific template invocations on the category page.

Supervised WordNet Disambiguation

A Wikipedia entity like `Degree (school)` could match several different WordNet entities for the word '*degree*', e.g. degree as a position on a scale,

or as the highest power of a polynomial. In Wikipedia, there are also other alternatives for each WordNet entity, e.g. degree as the number of edges incident to a vertex of a graph, or '*Degree*' as a brand name. In order to reliably assess the similarity between a Wikipedia article, category, or infobox and a WordNet synset, we relied on a supervised linking function to disambiguate possible meanings. The linking function relies on Ridge Regression (Bishop, 2007) to derive a model from a small set of manually labelled training examples (see Section 5.6.3). It uses three major signals as features.

Term Overlap. The term overlap feature quantifies the degree of similarity between the respective human language terms associated with entities. Here, the set $\mathrm{terms}(x)$ for a Wikipedia entity x is given by its title (after removing additional qualifications and detecting the correct capitalization, as mentioned earlier) and titles of its redirection articles. A set of terms for a WordNet entity is retrieved from the English, Arabic (Rodríguez et al., 2008), Catalan (Benitez et al., 1998), Estonian (Orav and Vider, 2005), Hebrew (Ordan and Wintner, 2007), and Spanish (Atserias et al., 2004a) wordnets as well as from MLSN (Cook, 2008).

For a Wikipedia entity x and a WordNet entity y, the term overlap feature is then computed as:

$$\sum_{t_\mathrm{x} \in \mathrm{terms}(x)} \max_{t_\mathrm{y} \in \mathrm{terms}(y)} \phi_\mathrm{x}(t_\mathrm{x}, x)\ \phi_\mathrm{y}(t_\mathrm{y}, y)\ \mathrm{sim}(t_\mathrm{x}, t_\mathrm{y}) \tag{5.1}$$

Here, $\mathrm{sim}(t_\mathrm{x}, t_\mathrm{y})$ is a a simple similarity measure between terms that returns 1 if the languages match and the strings are equal after lemmatizing, and 0 otherwise.

For Wikipedia, the additional term weighting ϕ_x generally yields 1, while for WordNet multiple different versions of ϕ_y are used in separate features. One option is to have ϕ_y return $1/n$ when n different meanings of t_y are listed in WordNet. Additionally, we also use WordNet's SemCor corpus frequency and synset rank weights as given in Table 3.2 in Chapter 3.

It turns out that determining the right capitalization of terms aids in avoiding incorrect matches. WordNet synsets for '*college*' will then only match articles about colleges but not articles about films or subway stops called '*College*'.

Cosine Similarity. The cosine vector similarity feature is computed as $\mathbf{v}_x^T \mathbf{v}_y (||\mathbf{v}_x||_2 \, ||\mathbf{v}_y||_2)^{-1}$ for vectors $\mathbf{v}_x$, $\mathbf{v}_y$ derived for the short description gloss extracted from the English Wikipedia in Section 5.3.2 and the gloss and terms provided by WordNet, respectively. The vectors are created using TF-IDF scores after stemming using Porter's method, as in Section 3.4.5 (page 50).

Primary Sense Heuristic. The primary sense feature is computed by taking the set of unqualified English titles for the Wikipedia entity x or any of its redirects, and then counting for how many of them the WordNet synset y is listed as the first (most frequent) noun sense in WordNet. A Wikipedia title like '*College*' is considered unqualified if it does not include an additional qualification in parentheses, unlike '*College (canon law)*'. The most frequent sense of '*college*' listed in WordNet is much more likely to correspond to Wikipedia's '*College*' article than to pages with additional qualifications like '*College (canon law)*' or '*College (1927 film)*'. Unqualified titles reflect the most important meaning of words as chosen by Wikipedia editors, and thus are more likely to correspond to the first sense of those words listed in WordNet.

Together, these three signals allow us to learn a regression model that assesses whether a Wikipedia article and a WordNet synset are likely to represent the same semantic entity.

Redirect Matching

Many projects treat redirect titles in Wikipedia as simple alias names of an entity. However, the meanings of many redirect titles differ significantly from those of their respective redirect target pages. For instance, there are redirects from `Physicist` (i.e. human beings) to `Physics` (a branch of science) and from `God does not play dice` to `Albert Einstein`. Large numbers of redirects exist from song names to album names or artist names, and so on. We decided to conservatively equate redirects with their targets only in the following two cases.

1. The titles of redirect source and redirect target match after parenthesized substring removal, Unicode NFKD normalization (Davis and Dürst, 2008), diacritics and punctuation removal, and lowercase conversion. This means that `London` would match `London`

(England), London (UK), and LONDON, but not Southwest London or Climate of London.

2. The redirect uses certain templates or categories that explicitly indicate co-reference with the target (alternative names, abbreviations, etc.).

Other redirects still have a chance of being connected to their targets later on, by the methods described in Section 5.5.1.

Infobox Matching

The infobox matching linking function returns a constant $w > 0$ when an infobox template like Infobox university is matched with an article or category having a corresponding title, in this case University, and 0.0 otherwise. We chose $w = 0.5$ because these mappings are not as reliable as interwiki links or redirect links. The function does not consider article titles with additional qualifications as matching, so University (album) would not be considered.

5.4.2 Subclass Link Heuristics

Subclass linking functions use simple heuristics to connect a class x to its potential parent classes y.

Parent Categories

The parent category linking function checks if semantic entities x for Wikipedia categories can be considered subclasses in the ontological sense of entities y for their own parent categories as listed in Wikipedia.

To accomplish this, it ensures that both x and y are likely to be categories denoting genuine classes. A genuine class like Universities can have instances as its class members (individual universities, ontologically speaking, are regarded as instances of Universities). In contrast, other categories like Education or Science education merely serve as topic labels. It would be wrong to say that the University of Trento "is an" Education. For distinguishing the two cases automatically, we found that the following heuristic generalizes the singular/plural heuristic proposed for YAGO (Suchanek et al., 2007) to the multilingual case:

- headword nouns that are countable (can have a plural form) tend to indicate genuine classes
- headword nouns that are uncountable (exist only in singular form) tend to be topic tags

Hence, we take the titles of a category as well as its cross-lingual counterparts, remove qualifications in parentheses, and, if available, rely on a parser to retain only the main headword. In practice, we exclusively use the English Link Grammar parser (Sleator and Temperley, 1993). For large numbers of non-English categories, it suffices to work with the entire string after removing qualifications, e.g. the German Wikipedia uses titles like `Hochschullehrer (Berlin)` rather than titles like `German academics`. In most other cases, the Markov Chain Taxonomy Induction algorithm will succeed at ensuring that taxonomic links are nevertheless induced. We then check that whatever term remains is given in plural (for English), or is countable (in the general case). Countability information is extracted from WordNet and Wiktionary (*wiktionary.org*), the latter using regular expressions. We also added a small list of Wikipedia-specific exceptions (words like '*articles*', '*stubs*') that are excluded from consideration as classes.

The linking function returns 1 if y is a parent category of x and both x and y are likely to be genuine classes, and 0 otherwise.

Category-WordNet Subclass Relationships

If x is a category, then the headword of its title also provides a clue as to what parent classes are likely in the input wordnets. For instance, a category like `University book publishers` has '*publishers*' as a headword. While we need the headword to be covered by the input wordnets, it suffices to use the English WordNet and perhaps a few other ones. As we will later see, even if one were to use only Princeton WordNet, the Markov Chain Taxonomy Induction algorithm could easily integrate most categories, because the majority of non-English categories will have `equals` arcs to English categories or `subclass` links ultimately leading to an article or category that is connected to WordNet.

We again relied on supervised learning to disambiguate possible meanings of a word, as earlier employing Ridge Regression (Bishop, 2007) to learn a model that recognizes likely semantic entities based on

a labelled training set (see Section 5.6.4). The main features are again of the form

$$\sum_{t_{\mathrm{x}} \in \mathrm{terms}(x)} \max_{t_{\mathrm{y}} \in \mathrm{terms}(y)} \phi_{\mathrm{x}}(t_{\mathrm{x}}, x)\ \phi_{\mathrm{y}}(t_{\mathrm{y}}, y)\ \mathrm{sim}_{\mathrm{hw}}(t_{\mathrm{x}}, t_{\mathrm{y}}) \tag{5.2}$$

This is similar to Equation 5.1, however $\mathrm{sim}_{\mathrm{hw}}(t_{\mathrm{x}}, t_{\mathrm{y}})$ matches with headwords of titles t_{x} rather than full titles t_{x} if such information is available. As for the `subclass` links, qualifications in parentheses are removed, and then the Link Grammar parser is used to retain only the headword (Sleator and Temperley, 1993) if possible. Additionally, $\phi_{\mathrm{x}}(t_{\mathrm{x}}, x)$ will be 1 if t_{x} is in plural or countable and 0 otherwise, allowing us to distinguish topic labels from genuine classes. The second weighting function $\phi_{\mathrm{y}}(t_{\mathrm{y}}, b)$ again uses the number of alternative meanings as well as synset rank and corpus frequency information. Apart from this, the linking also relies on the cosine similarity feature used earlier for `equals`. Together, these features allow the model to disambiguate between relevant WordNet synsets. A few exceptions are specified manually, e.g. '*capital*', '*single*', '*physics*', '*arts*', and Wikipedia-specific ones like '*articles*', '*pages*', '*templates*'.

WordNet Hypernymy

WordNet's notion of hypernymy between synsets is closely related to the `subclass` relation. The hypernymy linking function hence returns 1 if y is a hypernym of x in WordNet, and 0 otherwise.

5.4.3 Instance Link Heuristics

Instance linking functions link individual entities to their classes.

Infoboxes

A `University` infobox placed in a Wikipedia article is a very strong indicator of the article being about a university. The instance linking function returns a constant $w_{\mathrm{infobox}} > 0$ if y is recognized as an infobox template that occurred on the page of the article associated with x, and 0 otherwise. Since infoboxes are incorporated into Wikipedia articles by means of simple template invocations, heuristics need to be used

to distinguish them from other sorts of template invocations. For this, we rely on a list of suffixes and prefixes (like "_Infobox") for different languages. The `instance` links generated by the infobox linking function are useful later on, because we will also have `equals` links between infobox templates and articles, as described in Section 5.4.1.

Categories

Entities for articles like `Free University of Bozen-Bolzano` are made instances of certain categories, e.g. `Universities in Italy`, but not of topic categories like `Bolzano`. If y is a Wikipedia category for the article associated with x, the category linking function assesses whether a headword of y (or of its interwiki translations) is in plural or countable, and returns 1 if this is the case, and 0 otherwise, as earlier for subclass relations.

We will now explain what these linking functions give us and what needs to be done in order to obtain a more coherent output knowledge base.

5.5 Taxonomy Induction

Applying the linking functions to the input as in Algorithm 5, we obtain a graph $G_0 = (V_0, A_0, \Sigma)$ with an extended arc set A_0 connecting semantic entities from multiple knowledge sources to each other, in our case articles, categories, infoboxes (from different editions of Wikipedia), as well as WordNet entities. As shown in Figure 5.2, the connections include `equals` statements (bidirectional arrows) representing equivalence, `subclass` statements connecting categories and WordNet entities to parent classes, and `instance` statements connecting articles to categories and infoboxes (unidirectional arrows).

However, due to the noisy heuristic nature of these arcs and the fact that these entities come from different sources, it is not trivial to recognize that '*Fersental*' is a valley rather than a language. In fact, in reality, we may have more than 50 languages and many more potential parents for an entity. What is needed is a way to aggregate information and produce the final, much cleaner and more coherent knowledge

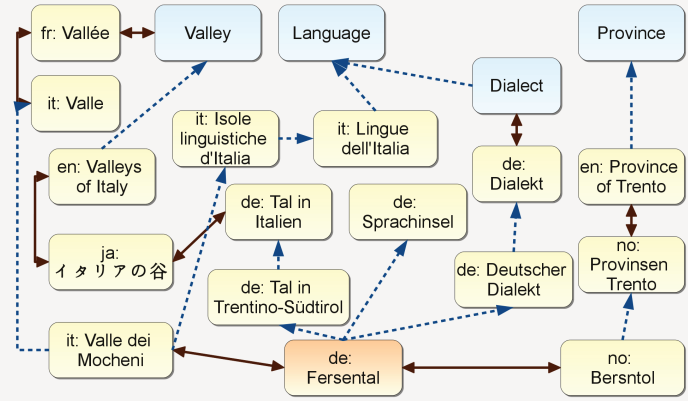


Figure 5.2: Simplified illustration of noisy input from link heuristics

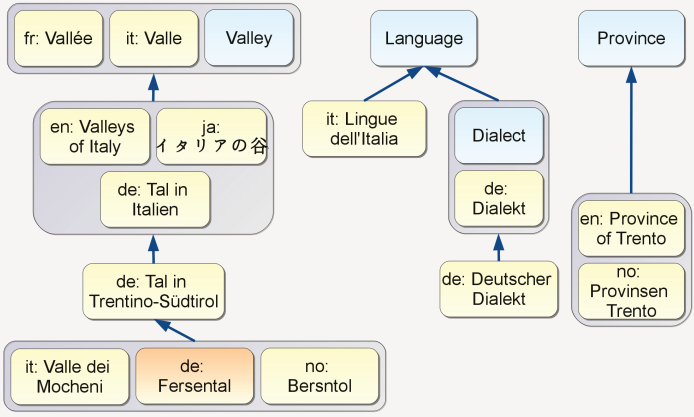


Figure 5.3: Relevant sample of the desired output

base, which would ideally include what is depicted in Figure 5.3. We proceed in two steps. The first step aggregates entity identifiers referring to the same entity by producing consistent equivalence classes. In the second step, taxonomic information from different linking functions is aggregated to produce the clean output taxonomy.

5.5.1 Consistency of Equivalence Information

In general, there will often be multiple entity identifiers that refer to the same entity and that are connected by `equals` statements. For instance, the German `Fersental` is equivalent to the corresponding Italian, Norwegian, and other articles about the valley. It will sometimes be convenient to jointly refer to all of these equivalents.

To make the knowledge base more coherent, one key ingredient is taking into account the symmetry and transitivity of equivalence. In practice, we may have an infobox in some non-English edition with an `equals` arc to an article, which has an `equals` arc to a category, which in turn has an interwiki link to an English category, and so on.

This leads us to the following definition to capture the weakly connected components corresponding to the symmetric, transitive closure of `equals`.

Definition 5.2 (e-component) *In a knowledge base $G = (V, A, \Sigma)$, an* e-component *$E \subseteq V$ for some entity $v_0 \in V$ is a minimal set of entities containing v_0 such that $v \in E$ for all $u \in E$, $v \in V$ with statements $(u, v, r, w) \in A$ or $(v, u, r, w) \in A$ (with $r =$ `equals`, $w > 0$). We use the notation $E(v_0)$ to denote the e-component containing a node v_0.*

As we saw earlier in Chapter 4, due to the heuristic nature of the equality linking functions, it often occurs that two entities u, v are transitively identified within an e-component, although we are quite sure that they should not be. For instance, we may have two different Wikipedia articles linked to the same WordNet synset. In some cases, the input from Wikipedia is imprecise, e.g. the Catalan article about the city of Bali in Rajasthan, India, as of October 2010, is linked to the Hindi article about the Indonesian island of Bali.

We will again be using our WDGS framework from Chapter 4, and most of the distinctness assertions will again come from the criteria

in Section 4.4.1. We additionally apply the following two criteria to avoid multiple WordNet synsets from being merged and to ensure that disambiguation pages are not mixed up with regular articles. In Wikipedia, disambiguation pages are special pages that provide a list of available articles for ambiguous titles.

Criterion 5.1 (Distinctness of WordNet Synsets). We assume that WordNet does not contain any duplicate synsets and add a distinctness assertion $(D_{i,1}, D_{i,2}, \dots)$, consisting of a singleton set $D_{i,j} = \{v\}$ for each semantic entity v from WordNet.

Criterion 5.2 (Distinctness from Disambiguation Pages). We add an assertion $(D_{i,1}, D_{i,2})$ where $D_{i,1}$ contains all articles recognized as disambiguation pages, and $D_{i,2}$ contains all articles not recognized as disambiguation pages.

We could also have chosen not to remain that faithful to WordNet and only enforce distinctness between different branches of entities within WordNet, e.g. $(D_{i,1}, D_{i,2})$ where $D_{i,1}$ contains all abstract entities in WordNet and $D_{i,2}$ contains all physical entities in WordNet. Since we are aiming at a more precise upper-level ontology, we decided to maintain WordNet's fine-grained sense distinctions.

Algorithm. To reconcile the `equals` arcs with the distinctness information, we first apply generic graph partitioning heuristics (Dhillon et al., 2007) to break up very large sparsely connected components into individual, much more densely connected clusters. On each of these densely connected clusters, we then apply the more accurate WDGS algorithm from Chapter 4 with its logarithmic approximation guarantee. In a few rare cases, the LP solver may time out even for small partitions, in which case we resort to computing minimal s-t cuts (Edmonds and Karp, 1972) between individual pairs of entities that should be separated. Minimal s-t cuts can be computed efficiently in $O(VE^2)$ or $O(V^2E)$ time. The statements corresponding to the cut edges are removed, and hence we obtain smaller e-components that should no longer conflate different concepts.

5.5.2 Aggregated Ranking

Requirements

Having made the `equals` arcs consistent, we then proceed to build the class hierarchy. In order to create the final output taxonomy, we will reconsider which entities to choose as superordinate taxonomic parents for a given entity. In doing so, the following considerations will need to be acknowledged.

First of all, the taxonomic arcs provided as inputs in general are not all equally reliable, as many of them originate from heuristic linking functions. The input arcs are equipped with statements weights that indicate how much we can trust them.

Property 5.1 (**Ranking).** The output should be a *ranked list* of taxonomic parents with corresponding scores rather than a simple set, based on the weights of the taxonomic arcs. All other things being equal, a taxonomic parent of an entity (that is not in the same e-component) should receive a greater parent ranking score for that entity if the weight of an incoming arc is higher.

Additionally, to obtain a clean, coherent output, it is crucial to obtain rankings that take into consideration the fact that parents are not independent, but themselves can stand in relationships to each other. For example, two different versions of Wikipedia may have what is essentially the same class (`equals` arcs) or classes that are connected by means of subclass relationships (`subclass` arcs).

This is very important in practice, because we frequently observe that the input arcs link individual articles to their categories, but these categories are language-specific local ones that are not part of a shared multilingual class hierarchy. If an article is found to be in the class `Tal in Trentino-Südtirol` in the German Wikipedia, then the possible parent class `Valley` from WordNet, which is reachable by following `equals` and `subclass` links, should gain further credibility.

The same consideration also applies to the node whose parents are currently being considered. Clearly, when evaluating parents about a Malay Wikipedia article, we may benefit from information available about an equivalent English article entity, and vice versa.

Property 5.2 (Dependencies). A taxonomic arc from a node u to a node v with weight greater than 0 should contribute to the ranking scores of nodes v' that are reachable from v via `equals` and `subclass` arcs. When evaluating parents for a node v_0, outgoing taxonomic arcs of nodes v' that are reachable from v_0 via `equals` arcs should also contribute to the ranking.

Finally, it is fairly obvious that information coming from multiple sources is likely to be more reliable and salient. For example, many Wikipedia editions describe the Colorado River as a river, but only few declare it to be a border of Arizona.

Property 5.3 (Aggregation). If a parent node v is not in the same e-component as the node v_0 whose parents are being ranked, then, all other things being equal, v should be given a higher ranking score with incoming taxonomic arcs (of weight greater than 0) from multiple nodes than if v had incoming arcs from fewer of those nodes.

Markov Chain

Taking these considerations into account, in particular Property 5.2, requires going beyond conventional rank aggregation algorithms. We use a Markov chain approach that captures dependencies between nodes.

Definition 5.3 (Parent Nodes) *Given a set of entities S and a target relation r (`subclass` or `instance`), the set of* parents *$P(S, r)$ is the set of all nodes v_m that are reachable from $v_0 \in S$ following paths of the form $(v_0, v_1, \ldots, v_m)$ with $(v_i, v_{i+1}, r_i, w_i) \in A, w_i > 0$ for all $0 \leq i < m$, and specific r_i. The path length m may be 0 (i.e. the initial entity v_0 is considered part of the parent entity set), and may be limited for practical purposes. When producing subclass arcs as output ($r =$`subclass`), all r_i must be `subclass` or `equals`. When producing instance arcs as output ($r =$`instance`), the first r_i that is not `equals` must be an `instance` relation, and any subsequent r_i must be either `equals` or `subclass`.*

Definition 5.4 (Parent e-components) *Instead of operating on original sets of parent entities $P(S, r)$, we consider the corresponding set of*

parent e-components $\{E(v) \mid v \in P(S,r)\}$ *(see Definition 5.2), which consists of the e-components for all* $v \in P(S,r)$.

For every node v_0 in the input graph, we will retrieve the set of possible parents and construct a Markov chain in which each state corresponds to a parent e-component of v_0. The Markov chain will enable us to create a ranking of those parents.

Definition 5.5 *Given a source node* v_0 *in a knowledge base* $G = (V, A, \Sigma)$*, a target relation* r*, and a corresponding set of parent e-components* $\{E_0, \ldots, E_n\}$ *(such that* $v_0 \in E_0$*), we define*

$$w_{i,j} = \sum_{u \in E_i} \sum_{v \in E_j} \sum_{(u,v,r',w) \in A} w$$

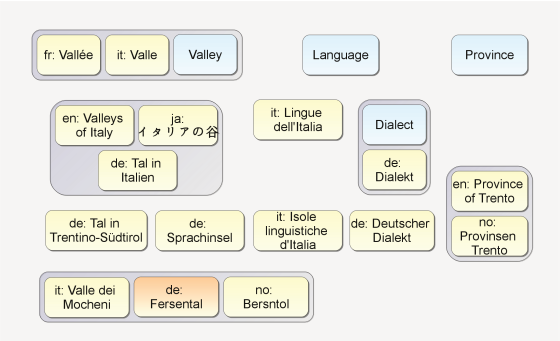
for all i*,*j *from* 0 *to* n*, where* r' *is* `instance` *if* $i = 0$ *and* r=`instance`*, and* r' *is* `subclass` *in all other cases (i.e. if* $i > 0$ *or* r =`subclass`*). We further define* $\Gamma_o(i)$ *as* $\{j \mid w_{i,j} > 0\}$.

If the target relation is `subclass`, this definition considers all `subclass` arcs between parent e-components. If the target relation is `instance`, we need to distinguish between outgoing arcs from E_0, which must be `instance` ones, and other outgoing arcs, which must be `subclass` ones.

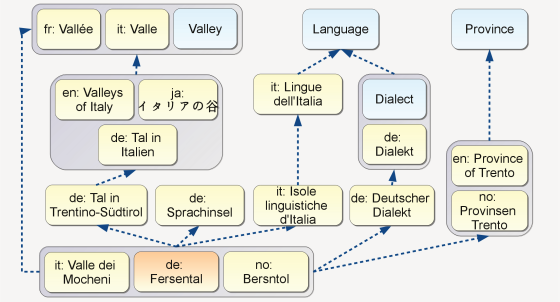
Definition 5.6 (Markov Chain) *Given an entity* v_0*, a corresponding set of parent e-components* $\{E_0, \ldots, E_n\}$ *(*$v_0 \in E_0$*), a weight matrix* $w_{i,j}$ *characterizing the links between different* E_i*, and a weight* $c \in \mathbb{R}^+$*, we define a Markov chain* $(E_{i_0}, E_{i_1}, \ldots)$ *as follows. The set* $\{E_0, \ldots, E_n\}$ *serves as a finite state space* S*, an initial state* $E_{i_0} \in S$ *is chosen arbitrarily, and the transition matrix* Q *is defined as follows.*

$$Q_{i,j} = \begin{cases} \dfrac{w_{i,j}}{c + \sum\limits_{k \in \Gamma_o(i)} w_{i,k}} & j \neq 0 \\ \dfrac{c + w_{i,j}}{c + \sum\limits_{k \in \Gamma_o(i)} w_{i,k}} & j = 0 \end{cases} \tag{5.3}$$

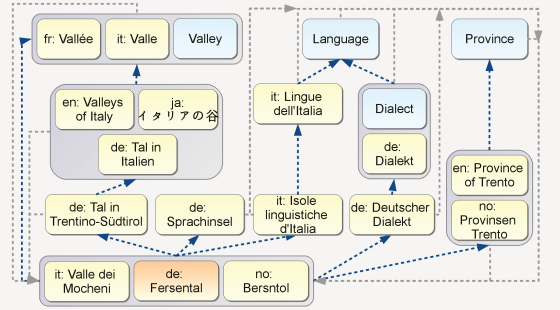
Figure 5.4 illustrates a Markov chain defined in this way: Part (a) shows parent e-components corresponding to states, Part (b) shows state transitions derived from taxonomic arcs between nodes in e-components, and Part (c) shows how one can transition back to the source node E_0, which contains `Fersental`, from any state.



(a) Parent e-components as state space



(b) State transitions based on taxonomic links



(c) Additional state transitions to source node

Figure 5.4: Markov chain setup

Theorem 5.7 *A transition matrix Q as defined in Definition 5.6 is stochastic.*

Proof. Given $c > 0$, for any $i \in \{0, \ldots, n\}$, we obtain

$$\begin{aligned}\sum_{j=0}^{n} Q_{i,j} &= \frac{c + w_{i,0}}{c + \sum\limits_{k \in \Gamma_o(i)}} + \sum_{j=1}^{n} Q_{i,j} \\ &= \frac{c + \sum\limits_{j=0}^{n} w_{i,j}}{c + \sum\limits_{k=0}^{n} w_{i,k}} \\ &= 1.\end{aligned}$$

□

The state space includes the e-component E_0 containing the source node. The probability mass received by E_0 rather than by genuine parents E_i with $i > 0$ in the stationary distribution reflects the extent of our uncertainty about the parents. For instance, if all immediate parents of the source node are linked with very low weights, then E_0 will attract a high probability mass. In the definition, c is the weight endowed to random restarts, i.e. transitions from arbitrary states back to E_0. Larger choices of c lead to a bias towards more immediate parents of E_0, while lower values work in favour of more general (and presumably more reliable) parents at a higher level. It is easy to see that the Markov chain is irreducible and aperiodic if $c > 0$, so a unique stationary distribution must exist in those cases.

Theorem 5.8 (Stationary Probability) *The Markov chain possesses a unique stationary probability distribution π with $\pi = \pi Q$.*

Proof. For any state $E \in S$, there exists some node $v_m \in E$ that is reachable from the source node v_0 by following a path of statements with non-zero weights as specified in Definition 5.3. The corresponding weights $w_{i,j}$ and state transition probabilities $Q_{i,j}$ along the path must be non-zero. Hence, every state is reachable from E_0.

Since $c > 0$, we obtain a non-zero random restart probability $Q_{i,0} > 0$ for every i, so from every state one can transition back to E_0, and thus the

chain is irreducible. Additionally, since $c > 0$, the state E_0 is aperiodic (one can remain in E_0 for any amount of steps), and hence the entire chain is aperiodic. By the Fundamental Theorem of Markov chains, a unique stationary distribution exists. □

Markov Chain Taxonomy Induction

This implies that we can use the stationary distribution of the Markov chain to rank parents of a source node with respect to their connectedness to that source node. The stationary distribution can easily be computed with the power iteration method. Algorithm 6 captures the steps taken to induce the taxonomy.

Input. As input, it takes a graph G_0 as defined in Chapter 2, containing information from the original knowledge sources as well as noisy `equals` and taxonomic statements, as produced by Algorithm 5. Additionally, one supplies the c parameter from Definition 5.6, an output selection function σ discussed below, parameters ϵ, $i_{\max}$ for the stationary probability computation, and the taxonomic root node v_{R} which is supposed to subsume all other classes (e.g. `Entity`).

Forming e-components. The algorithm begins by forming consistent e-components from the output of the WDGS framework. These become the entities of the output knowledge base. In practice, one may want to create entity identifier strings based on the entity identifiers within the e-component, perhaps preferring article titles in a specific language. Non-taxonomic statements like `means` statements that provide human-readable terms or statements capturing factual knowledge like birth dates of people are directly mapped to the e-components.

Ranking. Then, for each e-component E, the heuristics described in Section 5.4.2 are used to assess whether E is likely to be a class (checking headwords for Wikipedia and assuming yes for WordNet synsets without outgoing `instance` arcs). In accordance with the outcome of this assessment, the parents are retrieved and the transition matrix Q for the Markov chain is constructed. The fixed point $\pi = \pi Q$ can be computed using a number of different algorithms, e.g. the well-known

Algorithm 6: Markov Chain Taxonomy Induction algorithm

procedure TAXONOMY($G_0 = (V_0, A_0, \Sigma), c, \sigma, \epsilon, i_{\max}, v_{\mathrm{R}}$)
  $D_0, \ldots, D_k \leftarrow$ distinctness assertions for G_0 ▷ cf. Section 5.5.1
  apply WDGS framework to G_0, $D_0, \ldots, D_k$ ▷ cf. Section 5.5.1
  $V \leftarrow \{E(v) \mid v \in V_0\}$ ▷ consistent e-components become nodes
  $\Sigma_{\mathrm{T}} \leftarrow \{\texttt{equals}, \texttt{instance}, \texttt{subclass}\}$ ▷ set of taxonomic relations
  $A \leftarrow \{(E(u), E(v), r, w) \mid (u, v, r, w) \in A_0, r \notin \Sigma_{\mathrm{T}}\}$
    ▷ map all non-taxonomic statements
  $A_{\mathrm{T}} \leftarrow \emptyset$
  for all E in V **do** ▷ for all e-components
    $r \leftarrow \begin{cases} \texttt{subclass} & \text{if } E \text{ likely to be a class} \\ \texttt{instance} & \text{otherwise} \end{cases}$ ▷ see Section 5.4.2
    $E_0 \leftarrow E$
    $E_1, \ldots, E_n \leftarrow$ enumeration of $\{E(v) \mid v \in P(E, r)\} \setminus \{E\}$
      ▷ parent e-components as per Definition 5.4 in arbitrary order
    $Q \leftarrow$ transition matrix for E using $E_0, \ldots, E_n$ and c, r
      ▷ as per Definition 5.6
    $\pi \leftarrow$ EIGENVECTOR($Q, \epsilon, i_{\max}$)
    $A_{\mathrm{T}} \leftarrow A_{\mathrm{T}} \cup \{(E, E_i, r, \pi_i) \mid i > 0\}$ ▷ preliminary output
  $A \leftarrow A \cup \sigma(A_T)$ ▷ final output
  optionally remove entities not connected to $E(v_{\mathrm{R}})$ ▷ e.g. $v_{\mathrm{R}} = \texttt{Entity}$
  return $G = (V, A, \Sigma \cup \Sigma_{\mathrm{T}})$ ▷ taxonomic knowledge base as output
procedure EIGENVECTOR($[Q_{i,j}]_{i,j=1,\ldots,n}, \epsilon, i_{\max}$)
  choose uniform π with $\pi_i = \frac{1}{n}$ ▷ initial distribution
  $i \leftarrow 0$
  repeat ▷ Power iteration method
    $\pi' \leftarrow \pi$
    $\pi \leftarrow Q\pi$
    $i \leftarrow i + 1$
  until $||\pi - \pi'||_1 < \epsilon$ **or** $i \geq i_{\max}$
  return π

power iteration method. Although this process needs to be repeated for all e-components, these steps are nevertheless not a bottleneck (see Section 5.6).

Output. The final output is generated by some selection function σ from the preliminary output A_{T}. This can involve the following steps.

- As an optional step, filtering with respect to specific criteria can be performed, e.g. retaining only parents with Chinese labels, or only WordNet synsets as parents, and of course filtering with respect to some minimal weight threshold.
- Usually, the top-ranked k parent e-components E' will be chosen for a given E, where $k = 1$ leads to a more traditional taxonomy, while higher k lead to more comprehensive knowledge bases.
- Cycles of `subclass` relationships can optionally be removed. A cycle of formal subsumptions implies that all items in the cycle are equivalent. Since we have already merged nodes assumed to be equivalent into e-components, it makes sense to break up cycles. Cycles can be found in linear time by determining strongly connected components (Tarjan, 1972). In order to make the `subclass` arcs acyclic, one can remove the lowest-weighted `subclass` arc in each cycle.
- Redundant arcs to parent classes can be removed. Whenever there is an arc to a parent that is also a higher-order parent, we can remove the redundant direct arc to the parent.

Before completing, we can optionally prune all entities (and corresponding statements) which are not linked to the taxonomy's root node $E(v_{\mathrm{R}})$ by paths of taxonomic links in the output graph. This leads to an even more coherent knowledge base.

Analysis. Given a knowledge graph $G = (V_0, A_0, \Sigma)$ stored in a data structure that allows lookups in both directions of a directed arc, e-components can be found in linear time, i.e. $O(|V_0| + |A_0|)$, by iterating over the nodes and starting a depth-first search whenever an unseen node is encountered. Due to the overall sparsity of the graph with respect to `equals` arcs, the runtime will tend to be close to $O(|V_0|)$. Subsequently, for each $E \in V$, the same strategy can be used to retrieve the set of parent

e-components, and the weights $w_{i,j}$ can be computed on the fly while doing this. Computing the Markov chain's transition matrix Q can take $O(|V|^2)$ steps, and approximating the stationary distribution requires $O(|V|^2)$ operations if the power iteration method is used with a constant $i_{\max}$. This means that with these implementation choices, the overall worst-case complexity of the algorithm is $O(|V_0|^3)$. In practice, the set of parent e-components will be small, and additionally the transition matrices will be sparse, so the algorithm runs fairly quickly, as we show in Section 5.6.

Theorem 5.9 *The Markov Chain Taxonomy Induction algorithm possesses properties 5.1, 5.2, and 5.3, if* $c > 0$.

Proof. Definition 5.5 implies that, all other things being equal, a higher weight for a taxonomic arc from some node $u \in E_i$ to a parent $v \in E_j$ will lead to a higher weight $w_{i,j}$. We know that $c > 0$ and additionally assume $v \notin E_0$ (i.e. $j \neq 0$). Then, by Definition 5.6, $Q_{i,j}$ will increase (and at least $Q_{i,0}$ will decrease). Additionally, from the proof of Theorem 5.8, we know that Q is aperiodic and irreducible and hence regular. Due to the monotonicity of the stationary distribution of regular Markov chains (Chien et al., 2003), the e-component including v will have a greater probability mass in the new distribution, and Property 5.1 is fulfilled.

Similarly, given a node v' reachable from another node v via `equals` and `subclass` arcs, the state $E(v')$ must be reachable from $E(v)$ with non-zero probability, so any taxonomic arc from a node u to v also contributes to the ranking of v'. When evaluating parents for v_0, nodes v' that are reachable from v_0 via `equals` arcs are also in $E_0 = E(v_0)$, so outgoing taxonomic arcs of v' contribute to the ranking, and Property 5.2 is fulfilled.

Finally, Definition 5.5 implies that, all other things being equal, a parent $v \in E_j$ with input arcs from multiple children will have a higher sum of incoming weights $\sum_i w_{i,j}$ than the same parent if it had fewer of those incoming arcs. With $c > 0$ and assuming $j \neq 0$, this also implies a higher $\sum_i Q_{i,j}$. The monotonicity of the stationary distribution (Chien et al., 2003) then implies that Property 5.3 is satisfied. □

With these properties, Markov Chain Taxonomy Induction allows us to aggregate link information from heterogeneous sources, e.g. information from multiple editions of Wikipedia, including category and infobox information, and from WordNet. The output is a much more coherent taxonomic knowledge base, similar to the example excerpt in Figure 5.3, where clean e-components have been merged, and taxonomic links have been aggregated and cleaned.

5.6 Results

5.6.1 System Architecture

The system used to build MENTA is an extension of the one used in the previous chapters. As in Chapter 3, we use mappers to implement linking heuristics. The algorithm implementation from Chapter 4 ensures that the `equals` arcs are reconciled with the distinctness information.

The Markov Chain Taxonomy Induction algorithm is used to process the original noisy `subclass` and `instance` arcs that are provided as input. In order to increase the speed, we limited the maximal parent path length in Definition 5.3 to $m = 4$. This means that thousands of states that would obtain near-zero probabilities are pruned in advance. A second key to making the algorithm run quickly is relying on the fact that many entities share common parents, so the expensive lookups to determine potential parents should be cached. This allowed us to process all 19.9 million e-components in less than 3 hours on a single 3GHz CPU. Additionally, since the main loop in Algorithm 6 considers each source e-component separately, it would have been possible to parallelize the processing.

5.6.2 Dataset

We wrote a custom Web crawler that downloads the latest Wikipedia XML dumps from Wikimedia's download site, retrieving 271 different editions of Wikipedia as of April 2010. The size of the uncompressed XML dumps amounts to around 89.55 GB in total, out of which 25.4 GB stem from the English edition.

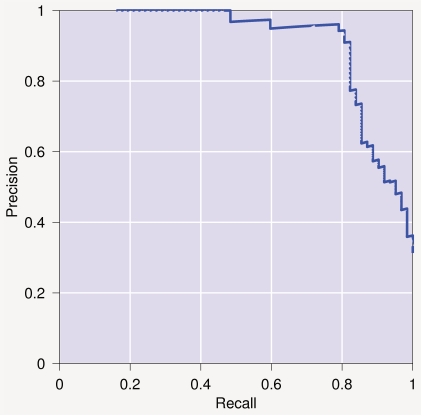


Figure 5.5: Precision-recall curve for Wikipedia-WordNet links

5.6.3 Entity Equality

Equality Information. The linking functions provided 184.3 million directed interwiki links and 7.1 million other directed `equals` arcs. The WordNet disambiguation model was obtained by training on 200 out of 407 manually labelled examples, selected randomly among all Wikipedia articles and WordNet synsets sharing a term. The precision-recall curve on the remaining 207 examples used as the test set (Fig. 5.5) shows the remarkably reliable results of the model. With a threshold of 0.5 we obtain 94.3% precision at 80.7% recall (F_1: 87.0%). The precision only drops sharply once we move towards recall levels significantly above 80%. See Section 3.5.2 for an introduction to precision and recall. The overall area under the ROC curve (ROC AUC) is 93.06%.

Distinctness Information. The equality arcs led to 19.5 million initial e-components, including templates, categories, and redirects. It turns out that roughly 150,000 of these e-components contained nodes to be separated, among them a single large e-component consisting of nearly 1.9 million nodes. Overall, more than 5.0 million individual node pairs are asserted to be distinct by the distinctness assertions.

Reconciliation. We applied the WDGS framework from Section 5.5.1 to separate the entities and obtain more consistent links. The process took several days to complete, with the expensive linear program solving by CPLEX (for the approximation algorithm) being the major bottleneck. We experimented with agglomerative clustering as an alternative, but found the WDGS solution costs to be orders of magnitude worse. Using the approximation algorithm, a total of 2.3 million undirected `equals` connections (4.6 million directed arcs) were removed, resulting in 19.9 million e-components after separation.

5.6.4 Taxonomy

Linking Functions. As additional input to the taxonomy induction algorithm, the linking functions produced what correspond to 1.2 million `subclass` arcs and 20.1 million `instance` arcs between e-components. For the `instance` arcs, we chose $w_{\text{infobox}} = 2$ because classes derived from infoboxes are more reliable than categories. The WordNet disambiguation model for `subclass` was obtained by training on 1,539 random mappings, the majority of these (1,353) being negative examples. On a test set of 234 random mappings, we obtain a precision of 81.3% at 40.0% recall, however going above 40% recall, the precision drops sharply, e.g. 60.8% precision at 47.7% recall. This task is apparently more difficult than the `equals` disambiguation, because less contextual information is directly available in the category page markup and because our heuristics for detecting classes may fail. Overall, there would be 6.1 million `subclass` arcs, but we applied a minimal threshold weight of 0.4 to filter out the very unreliable ones. The ROC AUC is only 65.8%. This shows that using the original linking functions alone can lead to a taxonomy with many incorrect links.

Algorithm. We thus relied on our Markov Chain Taxonomy Induction algorithm to choose reliable parents. In our experiments, the algorithm's c parameter was fixed at $c = \frac{1}{2}$, based on the intuition that if there is only one parent with weight 0.5, then that parent should be reached with probability $\frac{1}{2}$ from the current state. Examples of subclass and instance rankings are given in Tables 5.1 and 5.2, respectively, showing the highest-ranked parent entities from WordNet and Wikipedia. Note

Table 5.1: Ranked subclass examples

Class	WordNet Parent	Wikipedia Parent
Science museums in New Mexico	1. museum 2. science museum 3. depository	Museums Science museum Museums in New Mexico
Cathedrals in Belize	1. church building 2. cathedral (large church) 3. cathedral (diocese church)	Cathedral Churches in Belize Church buildings
Hamsters	1. rodent 2. hamster 3. mammal	Rodents Pets Domesticated animals

that in the final output, equivalent parents from WordNet and Wikipedia would in most cases form a single e-component. They are listed separately here for information purposes only.

Out of the 19.9 million e-components in the input, a large majority consist of singleton redirects that were not connected to their redirect targets, due to our careful treatment of redirect links in Section 5.4.1.

Coherence. For roughly 5.8 million e-components, we actually had outgoing `instance` links in the input. To quantify the coherence, we determine what fraction of these e-components can be connected to e-components involving WordNet synsets, as WordNet can be considered a shared upper-level core. Table 5.3 shows that this succeeds for nearly all e-components. The first column lists the number of entities for which we have outgoing `instance` arcs, while the second column is restricted to those for which we could establish `instance` arcs to WordNet (at a reachability probability threshold of 0.01). The small differences in counts between these two columns indicate that most entities for which there is any class information at all can be integrated into the upper-level

Table 5.2: Ranked instance examples

Entity	WordNet Parent	Wikipedia Parent
Fersental	1. valley 2. natural depression 3. geological formation	Valleys Valleys of Italy Valleys of Trentino / Alto Adige
Cagayan National High School	1. secondary school 2. school 3. educational institution	Secondary school School High schools in the Philippines
The Spanish Tragedy	1. book 2. publication 3. piece of work	Book British plays Plays

backbone provided by WordNet. The third column lists the number of e-components that are independent of the English Wikipedia but have successfully been integrated by our algorithm with `instance` links. While some fraction of those may correspond to entities for which cross-lingual interwiki links need to be added to Wikipedia, large numbers are entities of local interest without any matching English Wikipedia article. Additionally, we found that 338,387 e-components were connected as subclasses of WordNet synsets, out of a total of 360,476 e-components with outgoing `subclass` arcs.

Accuracy. Table 5.4 shows a manual assessment of highest-ranked WordNet-based parent classes for over 100 random entities. We rely on Wilson score intervals at $\alpha = 0.05$ (Brown et al., 2001) to generalize our findings to the entire dataset. For $k = 2, 3$, the ranked output is significantly more reliable than the $w_{i,j}$ between e-components resulting from the initial `subclass` arcs. The aggregation effect is even more noticeable for the `instance` arcs to WordNet in Table 5.5. To connect instances to WordNet, the algorithm needs to combine `instance` arcs with unreliable `subclass` arcs. Yet, the output is significantly more

Table 5.3: Coverage of individual entities by source Wikipedia

	Instances	Instances Linked to WordNet	Non-English Instances Linked to WN
English	3,109,029	3,004,137	N/A
German	911,287	882,425	361,717
French	868,864	833,626	268,693
Polish	626,798	579,702	159,505
Italian	614,524	594,403	161,922
Spanish	568,373	551,741	162,154
Japanese	544,084	519,153	241,534
Dutch	533,582	508,004	128,764
...	...	...	...
Total	13,982,432	13,405,345	2,917,999
E-components	5,790,490	5,379,832	2,375,695

Table 5.4: Accuracy of `subclass` arcs to WordNet

top-k	Sample Size	Initial Arcs	Ranked Arcs
1	104	$82.46\% \pm 7.08\%$	$83.38\% \pm 6.92\%$
2	196	$57.51\% \pm 6.85\%$	$83.03\% \pm 5.17\%$
3	264	$45.89\% \pm 5.97\%$	$79.87\% \pm 4.78\%$

accurate than the input `subclass` arcs, for $k = 1, 2$, and 3. This means that the Markov chain succeeds at aggregating evidence across different potential parents to select the most reliable ones.

We additionally asked speakers of 3 other languages to evaluate the top-ranked WordNet synset for at least 100 randomly selected entities covered in the respective language, but without corresponding English articles. We see that non-English entities are also connected to the shared upper-level ontology fairly reliably. The main sources for errors seem

to be topic categories that are interpreted as classes and word sense disambiguation errors from the subclass linking function. Fortunately, we observed that additional manually specified exceptions as in YAGO (Suchanek et al., 2007) would lead to significant accuracy improvements with very little effort. Certain categories are very frequent and account for the majority of disambiguation errors.

Table 5.5: Accuracy of `instance` arcs to WordNet

Language	top-k	Sample Size	Wilson Score Interval
English	1	116	$90.05\% \pm 5.20\%$
English	2	229	$86.72\% \pm 4.31\%$
English	3	322	$85.91\% \pm 3.75\%$
Chinese	1	176	$90.59\% \pm 4.18\%$
German	1	168	$90.15\% \pm 4.36\%$
French	1	151	$92.30\% \pm 4.06\%$

Coverage. The total number of output e-components in MENTA is roughly 5.4 million excluding redirects (Table 5.3), so with respect to both the number of entities and terms, MENTA is significantly larger than existing multilingual and monolingual taxonomies relying only on the English Wikipedia, which as of June 2010 has around 3.3 million articles. For many of these entities, MENTA contains additional supplementary information extracted from Wikipedia, including short glosses in many different languages, geographical coordinates for countries, cities, places, etc., and links to pictures, videos, and audio clips. For example, when looking up '*Mozart*', pictures as well as audio clips are available.

5.6.5 Lexical Knowledge

After forming e-components, the upper-level part of MENTA can be considered a multilingual version of WordNet. A total of 42,041 WordNet synsets have been merged with corresponding Wikipedia articles or

categories. We found that WordNet is extended with words and description glosses in 254 languages, although the coverage varies significantly between languages. The average number of Wikipedia-derived labels for these WordNet synsets is 20.

In Table 5.6, the results are compared with the results for UWN from Chapter 3, which is derived mainly from translation dictionaries. While MENTA's coverage is limited to nouns, we see that MENTA covers comparable numbers of distinct terms. The number of `means` statements is lower than for UWN, because each Wikipedia article is only merged with a single synset. The precision of MENTA's disambiguation is 94.3%, which is significantly higher than the 85-90% of UWN. This is not surprising, because an approach based on translation dictionaries has much less contextual information available for disambiguation, while MENTA can make use of Wikipedia's rich content and link structure.

Additionally, MENTA's output is richer, because we add not only words but also have over 650,000 short description glosses in many different languages as well as hundreds of thousands of links to media files and Web sites as additional information for specific WordNet synsets. Gloss descriptions are not only useful for users but are also important for word sense disambiguation (Lesk, 1986). Finally, of course, our resource adds millions of additional instances in multiple languages, as explained earlier.

UWN/MENTA Knowledge Base. The results suggest that we can obtain a more complete knowledge base by bringing together MENTA's large numbers of nouns and named entities with UWN's broad coverage of verbs, adjectives, adverbs, as well as alternative senses of nouns (e.g. '*school*' in the sense of the process of being educated).

We re-ran the UWN algorithm with a more up-to-date input graph, i.e. with larger numbers of input translations than in the experiments in Chapter 3. The new `means` statements and the corresponding terms were attached to MENTA's upper-level core, and duplicate statements were removed.

Table 5.7 gives the lexical coverage of this final, integrated UWN/MENTA knowledge base. We distinguish global values, which include large numbers of named entities and domain-specific concepts from Wikipedia, from the more restricted upper-level that consists of only

Table 5.6: Multilingual Wordnet (upper-level part of MENTA)

Language	means Statements in MENTA	Distinct Terms in MENTA	Distinct Terms in UWN
Overall	845,210	837,627	822,212
French	36,093	35,699	33,423
Spanish	31,225	30,848	32,143
Portuguese	26,672	26,465	23,499
German	25,340	25,072	67,087
Russian	23,058	22,781	26,293
Dutch	22,921	22,687	30,154

those entities that correspond to WordNet synsets. In total, there are roughly 90 languages with at least 10,000 means statements, including minority and regional languages like Welsh (48,983 means statements) and Cebuano (41,552). The coverage extends to more than 200 languages with at least 500 distinct terms and overall more than 300 languages. The large coverage and diverse range of languages show that this integrated resource comes very close to the universal multilingual knowledge base envisioned in the introduction in Chapter 1.

User Interface for Lexical Database Queries. A simple Web-based user interface has been implemented that allows users to look up words or names and browse some of the information available in the UWN/MENTA knowledge base. Figure 5.6 provides a screenshot. It is clear that the way language users search for information about words and their meanings has evolved significantly in recent years, as users are increasingly turning to electronic resources to address their lexical information needs. Traditional print media take more time to consult and are less flexible with respect to their organization. Alphabetical ordering, for instance, is not well-suited for conveying conceptual relationships between words.

Lexical databases, in contrast, can simultaneously capture multiple forms of organization and multiple facets of lexical knowledge. Especially

Table 5.7: Lexical coverage of final UWN/MENTA knowledge base

Language	`means` Statements		Distinct Terms	
	All	Upper Level	All	Upper Level
Overall	18,090,456[1]	2,280,039[1]	16,708,191[2]	1,757,616[2]
English	4,135,501[1]	66,541[1]	4,011,869[2]	55,772[2]
French	1,317,078	100,573	1,222,731	71,887
German	1,038,890	125,904	923,429	87,497
Spanish	880,254	88,798	813,605	61,911
Portuguese	732,092	68,116	667,417	48,887
Italian	722,787	74,843	662,198	53,645
Polish	678,026	47,240	612,512	38,000
Russian	649,304	80,739	578,820	57,980
Dutch	637,695	81,653	555,451	50,888
Japanese	571,797	36,799	530,535	29,859
Swedish	408,640	54,789	377,864	41,611
Finnish	286,876	59,773	260,785	44,668
Norwegian[3]	277,259	23,136	263,212	20,688
Chinese	274,361	32,179	263,253	33,920
Catalan	233,088	46,534	213,429	34,687
Czech	229,733	74,314	201,272	56,128
Turkish	225,400	62,548	198,662	46,259
Ukrainian	212,745	52,110	187,966	38,387
Romanian	212,116	28,588	200,918	23,188
Esperanto	197,303	62,812	170,882	43,467
Hungarian	196,669	47,543	180,539	38,213
Indonesian	150,773	41,691	141,513	36,201
Slovak	150,214	44,409	133,176	33,892
Danish	149,002	28,775	136,233	21,747
Korean	134,869	19,477	123,768	16,503
Vietnamese	133,967	11,312	129,103	10,170
Serbian	129,967	18,791	120,771	15,815
Arabic	120,669	15,660	114,467	13,264
Hebrew	117,600	21,081	107,992	17,366
Bulgarian	115,869	27,744	99,874	20,624
Volapük	114,152	2,968	97,264	2,289
Croatian	107,897	34,633	98,517	28,881
Thai	103,940	47,922	93,981	41,090
Slovene	101,794	17,872	94,280	13,850
…	…	…	…	…

1: counting only statements not already in Princeton WordNet
2: only terms with new `means` statements added to those already in WordNet
3: Norwegian Bokmål

with the advent of the World Wide Web, users are increasingly expecting to be able to lookup words and choose between different types of information, perhaps navigating quickly from one concept to another based on given links of interest. For example, a user wishing to find a Spanish word for the concept of persuading someone not to believe something might look up the word '*persuasion*' and then navigate to its antonym '*dissuasion*' to find the Spanish translation. A non-native speaker of English looking up the word '*tercel*' might find it helpful to see pictures available for the related terms '*hawk*' or '*falcon*'.

In our browsing interface, for a given entity, a list of relevant information is provided, sorted by category, salience and confidence. This is discussed in further detail in de Melo and Weikum (2010b), where we also explain how one can add etymological relationships, sense-specific example sentences, pronunciation information, information about misspellings or alternative spellings, and Chinese/Japanese/Korean character glyphs, among other things. A public demonstration showcasing a subset of the available information is available at `http://www.mpi-inf.mpg.de/yago-naga/menta/`.

Additionally, lexical knowledge bases like UWN/MENTA can also serve in task-specific user interfaces. For instance, the integrated English-language thesaurus of the OpenOffice.org application suite is based on WordNet. Sense-disambiguated translations as provided by UWN/MENTA could be of use in multilingual mobile communication aids for travellers (Uszkoreit et al., 2006).

5.6.6 Upper-Level Ontology

As mentioned earlier, the most generic part of an ontological taxonomy, i.e. the part at the top of the hierarchy, is known as the upper-level ontology. In the main MENTA build and in the final UWN/MENTA build, we have chosen to retain WordNet as an integral upper-level core of MENTA.

Wikipedia as Upper Level. Alternatively, we may also create a more Wikipedia-centric version where WordNet only serves as background knowledge to help us connect different articles and categories and obtain a more coherent taxonomy. To achieve this, it suffices to have

Figure 5.6: User interface

the selection function σ in the algorithm choose only e-components including Wikipedia articles or categories. This amounts to pruning all e-components that consist only of WordNet synsets without corresponding Wikipedia articles or categories. What we obtain is a taxonomy in which the root node is based on the English article `Entity` and its equivalents in other languages. At the upper-most level, the resulting taxonomy is shallower than with WordNet, as many different classes like `Organisms`, `Unit`, `Necessity`, are directly linked to `Entity`. At less abstract levels, the knowledge base becomes more complete. Tables 5.1 and 5.2 provide examples of top-ranked parent entities from Wikipedia.

Alternative Upper-Level Ontologies. In an additional experiment, we studied replacing WordNet's lexically oriented upper-level ontology with the more axiomatic one provided by SUMO (Niles and Pease, 2001). SUMO's expressive first-order (and higher-order) logic axioms enable applications to draw conclusions with some kind of common sense, capturing for example that humans cannot act before being born or that every country has a capital. Extending this with more specific

knowledge about entities from Wikipedia can give rise to a fruitful symbiosis, because such axioms can then be applied to individual entities.

We added SUMO's class hierarchy as well as the publically available mappings between WordNet and SUMO (Niles and Pease, 2003) as inputs to the instance ranking, and found that SUMO can be extended with 3,036,146 instances if we accept those linked to a SUMO class with a Markov chain stationary probability of at least 0.01. The sampled accuracy of 177 highest-ranked (top-1) arcs was $87.9\% \pm 4.7\%$. The inaccurate links often stemmed from mappings between WordNet and SUMO where the SUMO term did not appear to reflect the word sense from WordNet particularly adequately.

Since traditional theorem proving systems have difficulties coping with inconsistency and scaling to the large-scale knowledge bases produced by our work, we have collaborated with experts in the field to develop the SPASS-XDB theorem proving system, which dynamically incorporates relevant pieces of knowledge from large external databases or services on the fly (Suda et al., 2009; Sutcliffe et al., 2010).

5.6.7 Large-Scale Domain-Specific Extensions

A salient feature of our approach is that we can easily tap on additional large-scale knowledge sources in order to obtain even larger knowledge bases. For instance, we can rely on the many domain-specific knowledge bases in the Linked Data Web (Bizer et al., 2009), which describe biomedical entities, geographical objects, books and publications, music releases, etc. In order to integrate them we merely need an `equals` linking function for all entities and `equals` or `subclass` arcs for a typically very small number of classes. Our entity aggregation from Section 5.5.1 will then ensure that the links are consistent, and the Markov Chain Taxonomy Induction algorithm will choose the most appropriate classes, taking into account the weights of the `subclass` arcs.

As a case study, we investigated a simple integration of the LinkedMDB dataset, which describes movie-related entities. The `equals` links for instances were derived from the existing DBpedia links provided with the dataset, which are available for films and actors. Hence we only needed to specify two manual `equals` arcs for these two classes to allow all corresponding entities to be integrated. We obtain additional

information on 18,531 films and 11,774 actors already in our knowledge base. Additionally, up to 78,636 new films and 48,383 new actors are added. Similar extensions of MENTA are possible for many other domains.

5.6.8 Entity Search

Knowledge bases like MENTA are useful for semantic search applications. For instance, the Bing Web search engine has relied on Freebase to provide explicit lists of entities for queries like '*Pablo Picasso artwork*'.

In Table 5.8, we compare the numbers of instances obtained as results from the English Wikipedia with the numbers of instances in MENTA. The Wikipedia column lists the number of articles belonging to a given category in the English Wikipedia, while the MENTA columns list the number of e-components with outgoing `instance` arcs to the respective class e-components in MENTA's aggregated ranking (with a minimum stationary probability π_i of 0.01). Even if we consider only MENTA instances present in the English Wikipedia, i.e. e-components that include English Wikipedia pages, we often find more instances than directly given in the English Wikipedia, because our approach is able to infer new parents of instances based on evidence in non-English editions. Table 5.9 provides examples of entities from non-English Wikipedia editions integrated into the taxonomy.

Machine-readable knowledge bases allow for more advanced expert queries than standard text keyword search. For instance, one could search for philosophers who were also physicists, perhaps born in a specific time period and geographical area.

5.6.9 Non-Taxonomic Information

The taxonomic relations provide us with a global structure that connects all semantic entities in the knowledge base. Additionally, we can also include other relationships between entities. First of all, Wikipedia's category systems in different languages can be used to obtain large numbers of `hasCategory` arcs, connecting entities like `College` to topics like `Education`. Such information can be useful for word sense disambiguation (Buitelaar et al., 2006). Earlier, we already mentioned that we can extract geographical coordinates and multimedia links from

Table 5.8: Entity search query examples

Query	Wikipedia	MENTA (English Wikipedia)	MENTA (All)
cities and towns in Italy	8,156	8,509	12,992
european newspapers	13	389	1,963
people	441,710	882,456	1,778,078
video games developed in Japan	832	775	1,706

Table 5.9: Integrated non-English entities

Wikipedia edition	Entity	Top-Ranked Class in WordNet
French	Guillaume II (évêque de Meaux)	bishop
French	Hansalim	social movement
French	Tropanol	chemical compound
Chinese	王恩	person
Chinese	九巴士893	travel route
Chinese	东京梦华录	book

Wikipedia. Additionally, Wikipedia's infoboxes provide factual relationships between entities, e.g. the founding year and location of universities, the authors of books, and the genres of musicians. Such information can either be extracted from Wikipedia itself or from other databases that are derived from Wikipedia (Auer et al., 2007; Suchanek et al., 2007).

5.7 Discussion

We have presented techniques to relate entities from multiple knowledge sources to each other in terms of a coherent taxonomic hierarchy. As

a first step, this involves using linking functions to connect individual nodes that are equivalent or stand in a taxonomic relationship to each other. Subsequently, the entity integration framework from Chapter 4 cleans up the `equals` links. Finally, a Markov chain ranking algorithm is used to produce a much more coherent taxonomy while taking into account arc weights, dependencies in terms of `equals` arcs, and higher-order parents, among other things.

These methods were applied to the task of combining over 200 language-specific editions of Wikipedia as well as WordNet into a single knowledge base, where we succeeded in integrating 13.4 million out of 14.0 million possible articles from different Wikipedia editions into the upper-level ontology. The result of this work is MENTA, presumably the largest multilingual lexical knowledge base, which is freely available for download at `http://www.mpi-inf.mpg.de/yago-naga/menta/`.

We believe that MENTA can support a number of semantic applications, which leads to several opportunities for new research. For instance, all-words word sense disambiguation using WordNet is well-studied but definitely not a solved problem (Agirre et al., 2010). In particular, established systems have not been designed to support large numbers of named entities in conjunction with WordNet's fine-grained sense distinctions. Additionally, many current systems need to be adapted to operate on non-English text.

The entity search problem also needs to be studied further. Users may wish to pose natural language queries like '*What are the top-selling video games developed in Japan?*' or '*Which cities in France have mayors born in the 1930s?*'. The required factual data from Wikipedia can be incorporated into MENTA, but mapping natural language requests to knowledge base queries is non-trivial.

Further experiments could be carried out by applying our taxonomy induction in alternative settings. Apart from MENTA, we showed that our Markov Chain Taxonomy Induction algorithm is flexible enough to work with an alternative upper-level ontology like SUMO, or with additional knowledge from the Linked Data Web. Our framework could also operate on input from large-scale information extraction techniques (Tandon and de Melo, 2010), which collect named entities and clues about their classes from text. Overall, this framework paves the way

for new knowledge bases that integrate many existing large-scale data sources while offering more than the sum of the inputs.

CHAPTER 6

Conclusion

6.1 Summary

This thesis has presented graph-based methods to create large knowledge bases. Prior to our work, there had been little research on automatic approaches to produce multilingual semantic resources. In this thesis, we have presented three new techniques that induce large-scale multilingual knowledge bases by smartly integrating and reconciling input signals from existing knowledge sources and heuristics.

The lexical integration strategy attaches multilingual words to semantic entities by learning models that make use of carefully chosen features. These features reflect certain properties of the neighbourhood in the graph and allow us to disambiguate possible meanings of a word. The entity integration framework allows us to incorporate entities from different knowledge sources by reconciling equality information and distinctness information using linear programming and region growing techniques. The taxonomic integration method derives a coherent large-scale taxonomic organization from multiple knowledge sources and noisy, incomplete heuristic inputs, using a ranking based on Markov chains.

Together, these methods have been used to create the UWN/MENTA knowledge base, which is one of the largest multilingual knowledge bases available, describing over 5 million entities with over 16 million natural language words and names in over 200 different languages.

It additionally provides gloss descriptions of entities in different languages, and factual information about them.

6.2 Outlook

From a resource perspective, people looking for multilingual knowledge bases had few options available before our construction of the UWN/MENTA knowledge base. Those alternatives that are available do not offer the same level of massive multilingualism and taxonomic structuring. UWN/MENTA is freely available for download from `http://www.mpi-inf.mpg.de/yago-naga/uwn/`.

The UWN/MENTA knowledge base can serve as a catalyst for new research on text mining and language technology in different markets, as well as on new cross-lingual applications. As time passes, the knowledge sources currently used to create UWN/MENTA will expand, and additional new sources can be added, so continued growth and improvement is assured.

Additionally, the underlying algorithms and techniques themselves can play an important role in the future. The increasing number of information sources on the Web, including different editions of Wikipedia, Wiktionary, Linked Data datasets (Bizer et al., 2009), and many others, have brought us many new opportunities but also new challenges. Often, an application will need to draw on more than just one or two knowledge sources. This thesis has presented methods that allow applications to make sense of information from several knowledge sources and operate on a more coherent, unified view of the knowledge. We believe that this can be an important contribution towards the more general challenge of building bridges to tie together information from disparate origins and different perspectives.

List of Figures

1.1 Universal index of meaning 3
1.2 Hierarchical relations 4

2.1 Excerpt from Etymological WordNet 14
2.2 Monolingual lexical knowledge 22
2.3 Multilingual lexical knowledge 23

3.1 Lexical integration strategy (simplified) 27
3.2 Semantic relations 29
3.3 Excerpt of input and desired output graph 32
3.4 Candidate arc creation 40
3.5 Precision-recall curve on validation set 59
3.6 Excerpt from UWN graph 59
3.7 Excerpt from Roget's Thesaurus text file. 66
3.8 Excerpt from translation of Roget's Thesaurus text file. . . . 68

4.1 Entity integration strategy 78
4.2 Wikipedia articles in English and Japanese 80
4.3 Entities connected by `equals` links 81
4.4 Connected component with inaccurate links (simplified) . . 82
4.5 Distinctness assertion example 89
4.6 Potential solutions 91
4.7 Connected component not worth breaking up 93
4.8 Extended graph 100
4.9 Misleading graph topology 111

5.1 Taxonomic integration strategy 119

5.2 Simplified illustration of noisy input from link heuristics . . 134
5.3 Relevant sample of the desired output 134
5.4 Markov chain setup 140
5.5 Precision-recall curve for Wikipedia-WordNet links 147
5.6 User interface 157

List of Tables

3.1 Feature computation formulae 44
3.2 Arc weighting functions 46
3.3 Iterations of algorithm with validation set scores 58
3.4 Precision of UWN result graph 58
3.5 Coverage of final UWN graph 61
3.6 Average degree with respect to `means` arcs 63
3.7 Quality assessment for imported relations 64
3.8 Evaluation of Roget's Thesaurus translation 66
3.9 Evaluation of extended Roget's Thesaurus translation 67
3.10 Coverage of extended Roget's Thesaurus translation 67
3.11 Sample entries from generated German thesaurus 69
3.12 Evaluation of semantic relatedness measures 69
3.13 Cross-lingual text classification results 72

4.1 Algorithm results 109
4.2 Examples of separated concepts 112

5.1 Ranked subclass examples 149
5.2 Ranked instance examples 150
5.3 Coverage of individual entities by source Wikipedia 151
5.4 Accuracy of `subclass` arcs to WordNet 151
5.5 Accuracy of `instance` arcs to WordNet 152
5.6 Multilingual Wordnet (upper-level part of MENTA) 154
5.7 Lexical coverage of final UWN/MENTA knowledge base . . 155
5.8 Entity search query examples 160
5.9 Integrated non-English entities 160

List of Algorithms

1 Lexical category compatibility for nodes 49
2 Lexical integration 53
3 Thesaurus generation 70
4 WDGS approximation algorithm 102
5 Linking function application 127
6 Markov Chain Taxonomy Induction algorithm 143

Bibliography

Eytan Adar, Michael Skinner, and Daniel S. Weld (2009). Information arbitrage across multi-lingual Wikipedia. In *Proceedings of the 2nd International Conference on Web Search and Web Data Mining (WSDM 2009)* (Ricardo A. Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu, editors), pp. 94–103. ACM, New York, NY, USA.

Eneko Agirre, Oier López de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers (2010). SemEval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 75–80. ACL, Uppsala, Sweden.

ANSI/NIZO (2005). *ANSI/NISO Z39.19-2005. Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. NISO Press, Bethesda, MD, USA.

Jordi Atserias, Salvador Climent, Xavier Farreres, German Rigau, and Horacio Rodríguez (1997). Combining multiple methods for the automatic construction of multilingual WordNets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 1997)*.

Jordi Atserias, German Rigau, and Luís Villarejo (2004a). Spanish WordNet 1.6: Porting the Spanish WordNet across Princeton versions. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004)*.

Jordi Atserias, Luís Villarejo, German Rigau, Eneko Agirre, John Carroll, Bernardo Magnini, and Piek Vossen (2004b). The MEANING Multilingual Central Repository. In *Proceedings of the 2nd Global WordNet Conference (GWC 2004)*, pp. 80–210.

Sören Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives (2007). DBpedia: a nucleus for a web of open data. In *Proceedings*

of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC 2007), Busan, Korea, November 11–15, 2007 (Aberer et al., editor), volume 4825 of *Lecture Notes in Computer Science*. Springer.

Adi Avidor and Michael Langberg (2007). The multi-multiway cut problem. *Theoretical Computer Science*, 377(1-3):35–42.

Ricardo Baeza-Yates and Alessandro Tiberi (2007). Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, pp. 76–85. ACM, New York, NY, USA.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni (2007). Open information extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*, pp. 2670–2676. Morgan Kaufmann, San Francisco, CA, USA.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla (2004). Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Holger Bast, Alexandru Chitea, Fabian Suchanek, and Ingmar Weber (2007). Ester: Efficient search in text, entities, and relations. In *Proceedings of 30th International Conference on Research and Development in Information Retrieval (SIGIR 2007)* (Charlie Clarke, Norbert Fuhr, and Noriko Kando, editors). ACM, Amsterdam, Netherlands.

Christian Becker and Chris Bizer (2008). DBpedia Mobile: A location-enabled Linked Data browser. In *Proceedings of the Linked Data on the Web Workshop (LDOW 2008)*.

Abdelghani Bellaachia and Ghita Amor-Tijani (2008). Enhanced query expansion in English-Arabic CLIR. In *Proceedings of 19th International Conference on Database and Expert Systems Application (DEXA 2008)*. IEEE Computer Society, Washington, DC, USA.

Laura Benitez, Sergi Cervell, Gerard Escudero, Monica Lopez, German Rigau, and Mariona Taulé (1998). Methods and tools for building the Catalan WordNet. In *Proceedings of the ELRA Workshop on Language Resources for European Minority Languages at LREC 1998*.

Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom (2009). Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276.

Indrajit Bhattacharya and Lise Getoor (2007). Collective entity resolution in relational data. *TKDD*, 1(1).

Pushpak Bhattacharyya, Christiane Fellbaum, and Piek Vossen (editors) (2010). *Principles, Construction, and Applications of Multilingual Wordnets. Proceedings of the 5th Global WordNet Conference (GWC 2010)*. Narosa Publishing, India.

Daniel M. Bikel (2000). A statistical model for parsing and word-sense disambiguation. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC 2000)*, pp. 155–163. ACL, Morristown, NJ, USA.

Mikhail Bilenko, Sugato Basu, and Mehran Sahami (2005). Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pp. 58–65.

Mikhail Bilenko and Raymond J. Mooney (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003)*, pp. 39–48. ACM, New York, NY, USA.

Christopher M. Bishop (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. corr. 2nd printing edition.

Christian Bizer, Tom Heath, and Tim Berners-Lee (2009). Linked Data – the story so far. *International Journal on Semantic Web and Information Systems*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, pp. 1247–1250. ACM, New York, NY, USA.

Johan Bos and Katja Markert (2005). Recognising textual entailment with logical inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. ACL, Morristown, NJ, USA.

Gosse Bouma, Sergio Duarte, and Zahurul Islam (2009). Cross-lingual alignment and completion of Wikipedia templates. In *Proceedings of the 3rd International Workshop on Cross Lingual Information Access (CLIAWS3 2009)*, pp. 21–29. ACL, Morristown, NJ, USA.

Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta (2001). Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–133.

Paul Buitelaar, Thomas Eigner, and Thierry Declerck (2004). OntoSelect: A dynamic ontology library with support for ontology selection. In *Proceedings of the International Semantic Web Conference (ISWC 2004)*.

Paul Buitelaar, Bernardo Magnini, Carlo Strapparava, and Piek Vossen (2006). Domains in sense disambiguation. In *Word Sense Disambiguation: Algorithms and Applications* (Eneko Agirre and Phil Edmonds, editors), chapter 9, pp. 275–298. Springer.

Rainer E. Burkard, Mauro Dell'Amico, and Silvano Martello (2009). *Assignment Problems*. SIAM, Philadelphia, PA, USA.

Lou Burnard and Syd Bauman (2009). *TEI P5: Guidelines for Electronic Text Encoding and Interchange, Version 1.4.1*. TEI Consortium.

Patrick Cassidy (2000). An investigation of the semantic relations in the Roget's Thesaurus: Preliminary results. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2000)*, pp. 181–204.

Chih-Chung Chang and Chih-Jen Lin (2001). LIBSVM: a library for support vector machines. URL: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth (2005). Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383.

Niladri Chatterjee, Shailly Goyal, and Anjali Naithani (2005). Resolving pattern ambiguity for English to Hindi machine translation using WordNet. In *Proceedings of the Workshop on Modern Approaches in Translation Technologies at RANLP 2005*.

Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar (2005). On the hardness of approximating multicut and sparsest-cut. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity (CCC)*, pp. 144–153.

Steve Chien, Cynthia Dwork, Ravi Kumar, Daniel R. Simon, and D. Sivakumar (2003). Link evolution: Analysis and algorithms. *Internet Mathematics*, 1(3).

Martin S. Chodorow, Roy J. Byrd, and George E. Heidorn (1985). Extracting semantic hierarchies from a large on-line dictionary. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics (ACL 1985)*, pp. 299–304. ACL, Morristown, NJ, USA.

Andrea E. F. Clementi and Luca Trevisan (1996). Improved non-approximability results for vertex cover with density constraints. In *Proceedings of the 2nd International Conference on Computing and Combinatorics (COCOON 1996)*, pp. 333–342. Springer.

Edgar F. Codd (1970). A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387.

Edgar F. Codd (1990). *The relational model for database management: version 2*. Addison-Wesley Longman Publishing, Boston, MA, USA.

William W. Cohen, Henry Kautz, and David McAllester (2000). Hardening soft information sources. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000)*, pp. 255–259. ACM, New York, NY, USA.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI 2003 Workshop on Information Integration*, pp. 73–78.

Darren Cook (2008). MLSN: a multi-lingual semantic network. In *Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing (NLP 2008)*.

Corinna Cortes and Vladimir Vapnik (1995). Support-vector networks. *Machine Learning*, 20(3).

Jordi Daudé, Lluís Padró, and German Rigau (2000). Mapping wordnets using structural information. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, pp. 504–511. ACL, Morristown, NJ, USA.

Jordi Daudé, Lluís Padró, and German Rigau (2003). Making wordnet mappings robust. In *Proceedings of the 19th Congreso de la Sociedad Española para el Procesamiento del Lenguage Natural (SEPLN)*. Universidad de Alcalá de Henares. Madrid, Spain.

Mark Davis and Martin Dürst (2008). Unicode normalization forms, Rev. 29. Technical report, Unicode.

Gerard de Melo and Stefan Siersdorfer (2007). Multilingual text classification using ontologies. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007)*, volume 4425 of *LNCS*. Springer, Rome, Italy.

Gerard de Melo and Gerhard Weikum (2007). On the utility of automatically generated wordnets. In *Proceedings of the 4th Global WordNet Conference (GWC 2008)*, pp. 147–161. University of Szeged.

Gerard de Melo and Gerhard Weikum (2008a). Language as a foundation of the Semantic Web. In *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008)* (Christian Bizer and Anupam Joshi, editors), volume 401 of *CEUR WS*. CEUR, Karlsruhe, Germany.

Gerard de Melo and Gerhard Weikum (2008b). A machine learning approach to building aligned wordnets. In *Proceedings of the International Conference on Global Interoperability for Language Resources (ICGL 2008)*.

Gerard de Melo and Gerhard Weikum (2008c). Mapping Roget's Thesaurus and WordNet to French. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*. ELRA, Paris, France.

Gerard de Melo and Gerhard Weikum (2009a). Extracting sense-disambiguated example sentences from parallel corpora. In *Proceedings of the 1st Workshop on Definition Extraction in conjunction with RANLP 2009* (Gerardo Sierra, María Pozzi, and Juan-Manual Torres-Moreno, editors), pp. 40–46. INCOMA, Shoumen, Bulgaria.

Gerard de Melo and Gerhard Weikum (2009b). Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pp. 513–522. ACM, New York, NY, USA.

Gerard de Melo and Gerhard Weikum (2010a). MENTA: Inducing multilingual taxonomies from Wikipedia. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*. ACM, Toronto, Canada.

Gerard de Melo and Gerhard Weikum (2010b). Providing multilingual, multimodal answers to lexical database queries. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*. ELRA, Paris, France.

Gerard de Melo and Gerhard Weikum (2010c). Towards universal multilingual knowledge bases. In *Principles, Construction, and Applications of Multilingual Wordnets. Proceedings of the 5th Global WordNet Conference (GWC 2010)* (Pushpak

Bhattacharyya, Christiane Fellbaum, and Piek Vossen, editors), pp. 149–156. Narosa Publishing, India.

Gerard de Melo and Gerhard Weikum (2010d). Untangling the cross-lingual link structure of Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 844–853. ACL, Uppsala, Sweden.

Gerard de Melo and Gerhard Weikum (2011). Constructing and utilizing wordnets using statistical methods. *Journal Language Resources and Evaluation*. To appear.

Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis (2007). Weighted graph cuts without eigenvectors. a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957.

Irit Dinur and Shmuel Safra (2005). On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485.

Xin Dong, Alon Halevy, and Jayant Madhavan (2005). Reference reconciliation in complex information spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2005)*, pp. 85–96. ACM, New York, NY, USA.

Richard O. Duda, Peter E. Hart, and David G. Stork (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Susan T. Dumais and Hao Chen (2000). Hierarchical classification of Web content. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR 2000)* (Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors), pp. 256–263. ACM, Athens, Greece.

Jack Edmonds and Richard M. Karp (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios (2007). Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates (2004). Web-scale information extraction in KnowItAll: Preliminary results. In *Proceedings of the 13th International World Wide Web Conference (WWW 2004)*, pp. 100–110.

Oren Etzioni, Kobi Reiter, Stephen Soderland, and Marcus Sammer (2007). Lexical translation with application to image search on the Web. In *Proceedings of Machine Translation Summit XI, 2007*.

Jérôme Euzenat and Pavel Shvaiko (2007). *Ontology matching*. Springer-Verlag, Heidelberg, Germany.

Christiane Fellbaum (editor) (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. MIT Press.

Christiane Fellbaum and Piek Vossen (2007). Connecting the universal to the specific: Towards the Global Grid. In *Proceedings of the 1st International Workshop on Intercultural Collaboration (IWIC 2007)* (Toru Ishida, Susan R. Fussell, and Piek T. J. M. Vossen, editors), volume 4568 of *LNCS*, pp. 1–16. Springer.

Ivan P. Fellegi and Alan B. Sunter (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz (2007). Applying Wikipedia's multilingual knowledge to cross-lingual question answering. In *NLDB*, pp. 352–363.

John Rupert Firth (1957). A synopsis of linguistic theory 1930-55. In *Studies in Linguistic Analysis*, volume 1952-59, pp. 1–32. The Philological Society, Oxford.

Darja Fišer (2008). Using multilingual resources for building SloWNet faster. In *Proceedings of the 4th Global WordNet Conference (GWC 2008)*. Szeged, Hungary.

Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, and Ulrich Schäfer (2007). Question answering from structured knowledge sources. *Journal of Applied Logics, Special Issue on Questions and Answers: Theoretical and Applied Perspectives*, 5(1):20–48.

Sanae Fujita, Francis Bond, Stephan Oepen, and Takaaki Tanaka (2007). Exploiting semantic information for HPSG parse selection. In *Proceedings of the Workshop on Deep Linguistic Processing (DeepLP 2007)*, pp. 25–32. ACL, Morristown, NJ, USA.

Evgeniy Gabrilovich and Shaul Markovitch (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*, pp. 1606–1611. Morgan Kaufmann, San Francisco, CA, USA.

Nikesh Garera and David Yarowsky (2008). Minimally supervised multilingual taxonomy and translation lexicon induction. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pp. 465–472.

Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis (1996). Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing (SICOMP)*, 25:698–707.

Michael R. Genesereth and Nils J. Nilsson (1987). *Logical foundations of Artificial Intelligence*. Morgan Kaufmann, San Francisco, CA, USA.

Lise Getoor and Ben Taskar (2007). *Introduction to Statistical Relational Learning*. MIT Press.

Roxana Girju, Adriana Badulescu, and Dan Moldovan (2006). Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Zhiguo Gong, Chan Wa Cheang, and Leong Hou U (2005). Web query expansion by WordNet. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications (DEXA 2005)*, volume 3588 of *LNCS*, pp. 166–175. Springer.

Thomas R. Gruber (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford (2003). Record linkage: Current practice and future directions. Technical report, CSIRO Mathematical and Information Sciences.

Iryna Gurevych (2005). Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*. Jeju Island, Republic of Korea.

Iryna Gurevych, Christof Müller, and Torsten Zesch (2007). What to be? - Electronic career guidance based on semantic relatedness. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*. ACL, Prague, Czech Republic.

Harry Halpin and Patrick J. Hayes (2010). When owl:sameAs isn't the same: An analysis of identity links on the Semantic Web. In *Proceedings of the Workshop on Linked Data on the Web (LDOW 2010)*.

Sanda M. Harabagiu (editor) (1998). *Proceedings of the Workshop on the Usage of WordNet in Natural Language Processing Systems*. ACL, Université de Montŕeal, Montŕeal, QC, Canada.

Samer Hassan and Rada Mihalcea (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pp. 1192–1201. ACL, Morristown, NJ, USA.

Taher H. Haveliwala (2002). Topic-sensitive PageRank. In *Proceedings of the 11th International World Wide Web Conference (WWW 2002)*.

Patrick Hayes (2004). RDF semantics. W3C recommendation, World Wide Web Consortium.

Marti A. Hearst (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING 1992)*, pp. 539–545. ACL, Morristown, NJ, USA.

Mauricio A. Hernández and Salvatore J. Stolfo (1995). The merge/purge problem for large databases. *SIGMOD Rec.*, 24(2):127–138.

Werner Hüllen (2004). *A History of Roget's Thesaurus. Origins, Development, and Design*. Oxford University Press.

Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama, and Kyoko Kanzaki (2008). Development of the Japanese WordNet. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*. ELRA, Marrakech, Morocco.

Thorsten Joachims (1999). Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines* (B. Schölkopf, C. Burges, and A. Smola, editors). MIT Press, Cambridge, MA, USA.

Dmitri V. Kalashnikov and Sharad Mehrotra (2006). Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (ACM TODS)*, 31(2):716–767.

Narendra Karmarkar (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC 1984)*, pp. 302–311. ACM, New York, NY, USA.

George Karypis and Vipin Kumar (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392.

Zoubida Kedad and Elisabeth Métais (2002). Ontology-based data cleaning. In *Proceedings of the 6th International Conference on Applications of Natural Language*

to Information Systems (NLDB 2002) - Revised Papers, pp. 137–149. Springer-Verlag, London, UK.

Subhash Khot (2002). On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pp. 767–775. ACM, New York, NY, USA.

Adam Kilgarriff (1997). I don't believe in word senses. *Computers and the Humanities*, 31:91–113. 10.1023/A:1000583911091.

Daniel Kinzler (2008). *Automatischer Aufbau eines multilingualen Thesaurus durch Extraktion semantischer und lexikalischer Relationen aus der Wikipedia*. Master's thesis, Universität Leipzig.

Ioannis P. Klapaftis and Suresh Manandhar (2010). Taxonomy learning using word sense induction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*. ACL.

Kevin Knight and Steve K. Luk (1994). Building a large-scale knowledge base for machine translation. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994), vol. 1*, pp. 773–778. AAAI, Menlo Park, CA, USA.

Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee (2009). Media meets Semantic Web — how the BBC uses DBpedia and Linked Data to make connections. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009)*, pp. 723–737. Springer-Verlag, Berlin, Heidelberg.

Boris Lauser, Gudrun Johannsen, Caterina Caracciolo, Willem Robert van Hage, Johannes Keizer, and Philipp Mayr (2008). Comparing human and automatic thesaurus mapping approaches in the agricultural domain. In *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications (DCMI 2008)*, pp. 43–53. Dublin Core Metadata Initiative.

Donald Leatherdale, G. Eric Tidbury, and Roy Mack (1982). *AGROVOC : a multilingual thesaurus of agricultural terminology*. Apimondia.

Mong Li Lee, Wynne Hsu, and Vijay Kothari (2004). Cleaning the spurious links in data. *IEEE Intelligent Systems*, 19(2):28–33.

Tom Leighton and Satish Rao (1999). Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832.

Douglas B. Lenat and R. V. Guha (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing, Boston, MA, USA.

Michael Lesk (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC 1986)*. ACM.

M. Paul Lewis (2009). Ethnologue: Languages of the world, sixteenth edition (online version).

Dekang Lin (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING 1998)*, pp. 768–774. ACL, Morristown, NJ, USA.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng (2007). A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.

Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm (2001). Generic schema matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001)*, pp. 49–58. Morgan Kaufmann, San Francisco, CA, USA.

Andrea Marchetti, Maurizio Tesconi, Francesco Ronzano, Marco Rosella, and Francesca Bertagna (2006). Toward an architecture for the Global Wordnet initiative. In *Proceedings of the 3rd Italian Semantic Web Workshop (SWAP 2006)*, volume 201. CEUR-WS.org.

Marcin Marszałek and Cordelia Schmid (2007). Semantic hierarchies for visual object recognition. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition (CVPR 2007)*.

Mausam, Stephen Soderland, Oren Etzioni, Daniel Weld, Michael Skinner, and Jeff Bilmes (2009). Compiling a massive, multilingual dictionary via probabilistic inference. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009): Volume 1*, pp. 262–270. ACL, Morristown, NJ, USA.

C.O. Sylvester Mawson (editor) (1911). *Roget's Thesaurus of English Words and Phrases Classified and Arranged so as to Facilitate the Expression of Ideas and Assist in Literary Composition*. McDevitt-Wilson's, Inc., New York, NY, USA.

Andrew McCallum and Ben Wellner (2004). Conditional models of identity uncertainty with application to noun coreference. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*.

Lukas Michelbacher, Florian Laws, Beate Dorow, Ulrich Heid, and Hinrich Schütze (2010). Building a cross-lingual relatedness thesaurus using a graph similarity measure. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*. ELRA, Valletta, Malta.

Rada Mihalcea and Andras Csomai (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, pp. 233–242. ACM, New York, NY, USA.

Rada Mihalcea, Paul Tarau, and Elizabeth Figa (2004). PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, p. 1126. ACL, Morristown, NJ, USA.

David N. Milne, Ian H. Witten, and David M. Nichols (2007). A knowledge-based search engine powered by Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*. ACM, New York, NY, USA.

Alvaro Monge and Charles Elkan (1997). An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery*.

Simonetta Montemagni and Lucy Vanderwende (1992). Structural patterns vs. string patterns for extracting semantic information from dictionaries. In *Proceedings of the 14th Conference on Computational Linguistics (COLING 1992)*, pp. 546–552. ACL, Morristown, NJ, USA.

Jane Morris and Graeme Hirst (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1):21–48.

James Munkres (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

Vivi Nastase, Michael Strube, Benjamin Boerschinger, Caecilia Zirn, and Anas Elghafari (2010). WikiNet: A very large scale multi-lingual concept network. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*. ELRA.

Roberto Navigli and Simone Paolo Ponzetto (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 216–225. ACL, Uppsala, Sweden.

Dong Nguyen, Arnold Overwijk, Claudia Hauff, Dolf R. B. Trieschnigg, Djoerd Hiemstra, and Franciska De Jong (2009). WikiTranslate: query translation for cross-lingual information retrieval using only Wikipedia. In *Proceedings of the 9th Cross-Language Evaluation Forum Conference on Evaluating Systems for Multilingual and Multimodal Information Access (CLEF 2008)* (Carol Peters, Thomas Deselaers, Nicola Ferro, and Julio Gonzalo, editors), Lecture Notes in Computer Science 5706, pp. 58–65. Springer-Verlag, Berlin, Heidelberg.

Ian Niles and Adam Pease (2001). Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS 2001)* (Chris Welty and Barry Smith, editors).

Ian Niles and Adam Pease (2003). Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the IEEE International Conference on Information and Knowledge Engineering (IKE 2003)*, pp. 412–416.

Mikael Nilsson, Andy Powell, Pete Johnston, and Ambjörn Naeve (2008). Expressing Dublin Core metadata using the Resource Description Framework (RDF). URL: `http://dublincore.org/documents/2008/01/14/dc-rdf/`.

Franz Josef Och and Hermann Ney (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Jong-Hoon Oh, Daisuke Kawahara, Kiyotaka Uchimoto, Jun'ichi Kazama, and Kentaro Torisawa (2008). Enriching multilingual language resources by discovering missing cross-language links in Wikipedia. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008)*. IEEE, Washington, DC, USA.

Akitoshi Okumura and Eduard Hovy (1994). Building Japanese-English dictionary based on ontology for machine translation. In *Proceedings of the Workshop on Human Language Technology*, pp. 141–146. ACL.

Heili Orav and Kadri Vider (2005). Estonian Wordnet and lexicography. In *Proceedings of the 11th International Symposium on Lexicography*. Copenhagen.

Noam Ordan and Shuly Wintner (2007). Hebrew WordNet: a test case of aligning lexical databases across languages. *International Journal of Translation*, 19(1).

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd (1999). The PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford InfoLab.

Patrick Pantel and Marco Pennacchiotti (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*. ACL.

Jeff Pasternack and Dan Roth (2009). Learning better transliterations. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pp. 177–186. ACM, New York, NY, USA.

Fernando Pereira, Naftali Tishby, and Lillian Lee (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL 1993)*, pp. 183–190. ACL, Morristown, NJ, USA.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi (2002). MultiWordNet: Developing an aligned multilingual database. In *Proceedings of 1st Global WordNet Conference (GWC 2002), Mysore, India*, pp. 293–302.

John C. Platt (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers* (A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors), pp. 61–74. MIT Press, Cambridge, MA, USA.

Simone Paolo Ponzetto and Roberto Navigli (2009). Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI 2009)*. Morgan Kaufmann.

Simone Paolo Ponzetto and Michael Strube (2007). Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI 2007)*, pp. 1440–1445. AAAI Press.

Simone Paolo Ponzetto and Michael Strube (2008). WikiTaxonomy: A large scale knowledge resource. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*. IOS Press.

J. Ross Quinlan (1993). *C4.5: programs for machine learning*. Morgan Kaufmann, San Francisco, CA, USA.

Roy Rada and Brian K. Martin (1987). Augmenting thesauri for information systems. *ACM Trans. Inf. Syst.*, 5(4):378–392.

Philip Resnik (1995). Using Information Content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pp. 448–453. Morgan Kaufmann, San Francisco, CA, USA.

Reuters (2000a). Reuters Corpus, vol. 1: English Language, 1996-08-20 to 1997-08-19.

Reuters (2000b). Reuters Corpus, vol. 2: Multilingual Corpus, 1996-08-20 to 1997-08-19.

Horacio Rodríguez, David Farwell, Javier Farreres, Manuel Bertran, Musa Alkhalifa, Mª Antònia Martí, William J. Black, Sabri Elkateb, James Kirk, Adam Pease, Piek Vossen, and Christiane Fellbaum (2008). Arabic WordNet: Current state and future extensions. In *Proceedings of the 4th Global WordNet Conference (GWC 2008)*.

Daniel L. Rubin, Suzanna E. Lewis, Chris J. Mungall, Sima Misra, Monte Westerfield, Michael Ashburner, Ida Sim, Christopher G. Chute, Harold Solbrig, Margaret A. Storey, Barry Smith, John D. Richter, Natalya F. Noy, and Mark A. Musen (2006). National Center for Biomedical Ontology: Advancing Biomedicine through structured organization of scientific knowledge. *OMICS: A journal of integrative biology*, 10(2):185–98.

Stuart Russell and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. Pearson Education, 3 edition.

Satya S. Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda, and Ahmed Ezzat (2009). A survey of current approaches for mapping of relational databases to RDF.

Gerard Salton and Michael J. McGill (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA.

Patanakul Sathapornrungkij and Charnyote Pluempitiwiriyawej (2005). Construction of Thai WordNet lexical database from machine readable dictionaries. In *Proceedings of the 10th Machine Translation Summit, Phuket, Thailand*.

Kevin Scannell (2003). Automatic thesaurus generation for minority languages: an Irish example. In *Actes de la 10e conférence TALN, volume 2, Workshop Traitement Automatique des Langues Minoritaires et des Petites Langues*, pp. 203–212.

Nico Schlaefer, Jeongwoo Ko, Justin Betteridge, Manas Pathak, Eric Nyberg, and Guido Sautter (2007). Semantic extensions of the Ephyra QA system for TREC 2007. In *Proceedings of the 16th Text Retrieval Conference (TREC 2007)*. NIST.

Helmut Schmid (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK.

Hinrich Schütze (1992). Word space. In *Advances in Neural Information Processing Systems 5 (NIPS 1992)*, pp. 895–902. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Hinrich Schütze (1998). Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123.

Fabrizio Sebastiani (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Carina Silberer, Wolodja Wentland, Johannes Knopp, and Matthias Hartung (2008). Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*. ELRA, Marrakech, Morocco.

Daniel Sleator and Davy Temperley (1993). Parsing English with a Link Grammar. In *Proceedings of the 3rd International Workshop on Parsing Technologies*.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, and Suzanna Lewis (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng (2004). Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng (2006). Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pp. 801–808. ACL, Morristown, NJ, USA.

Philipp Sorg and Philipp Cimiano (2008). Enriching the crosslingual link structure of Wikipedia - a classification-based approach. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artifical Intelligence*.

Chen-Yu Su, Tien-Chien Lin, and Shih-Hung Wu (2007). Using Wikipedia to translate OOV term on MLIR. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access* (Noriko Kando and David Kirk Evans, editors), pp. 109–115. National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum (2007). Yago: A core of semantic knowledge. In *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*. ACM Press, New York, NY, USA.

Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum (2009). SOFIE: a self-organizing framework for information extraction. In *Proceedings of 18th International World Wide Web Conference (WWW 2009)*. ACM Press, New York, NY, USA.

Martin Suda, Geoff Sutcliffe, Patrick Wischnewski, Manuel Lamotte-Schubert, and Gerard de Melo (2009). External sources of axioms in automated theorem proving. In *Advances in Artificial Intelligence. 32nd Annual German Conference on AI (KI 2009), Paderborn, Germany, September 15-18, 2009* (Bärbel Mertsching, Marcus Hund, and Muhammad Zaheer Aziz, editors), volume 5803 of *Lecture Notes in Artificial Intelligence*, pp. 281–288. Springer.

Geoff Sutcliffe, Martin Suda, Alexandra Teyssandier, Nelson Dellis, and Gerard de Melo (2010). Progress towards effective automated reasoning with world knowledge. In *Proceedings of the 23rd International FLAIRS Conference*. AAAI Press, Menlo Park, CA, USA.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira (2008). Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. ACL, Morristown, NJ, USA.

Niket Tandon and Gerard de Melo (2010). Information extraction from Web-scale n-gram data. In *Proceedings of the Web N-gram Workshop at ACM SIGIR 2010* (Chengxiang Zhai, David Yarowsky, Evelyne Viegas, Kuansan Wang, and Stephan Vogel, editors), volume 5803, pp. 7–14. ACM.

Robert Tarjan (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

Jörg Tiedemann (2003). Combining clues for word alignment. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pp. 339–346. ACL, Morristown, NJ, USA.

Jörg Tiedemann (2004). The OPUS corpus - parallel & free. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004)*.

Jörg Tiedemann (2007). Improved sentence alignment for movie subtitles. In *Proceedings of the International Conference on Recent Advances in NLP (RANLP 1997)*.

Antonio Toral, Rafael Muñoz, and Monica Monachini (2008). Named Entity WordNet. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*. ELRA, Marrakech, Morocco.

Dan Tufiş, Dan Cristea, and Sofia Stamou (2004). BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. *Romanian Journal on Information Science and Technology*, 7(1–2):9–34.

Hans Uszkoreit, Feiyu Xu, Jörg Steffen, and Ihlan Aslan (2006). The pragmatic combination of different cross-lingual resources for multilingual information services. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC 2006)*. Genova, Italy.

Cornelis Joost van Rijsbergen (1979). *Information Retrieval (2nd Ed.)*. Butterworth, London, UK.

Vladimir N. Vapnik (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA.

Johanna Völker, Denny Vrandečić, York Sure, and Andreas Hotho (2007). Learning disjointness. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007)*, pp. 175–189. Springer-Verlag, Berlin, Heidelberg.

Piek Vossen (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Springer.

Piek Vossen, Eneko Agirre, Nicoletta Calzolari, Christiane Fellbaum, Shu kai Hsieh, Chu-Ren Huang, Hitoshi Isahara, Kyoko Kanzaki, Andrea Marchetti, Monica Monachini, Federico Neri, Remo Raffaelli, German Rigau, Maurizio Tescon, and Joop VanGent (2008). Kyoto: a system for mining, structuring

and distributing knowledge across languages and cultures. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*. ELRA, Marrakech, Morocco. Http://www.lrec-conf.org/proceedings/lrec2008/.

William E. Winkler (1999). The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau.

Fei Wu and Daniel S. Weld (2008). Automatically refining the wikipedia infobox ontology. In *Proceeding of the 17th International World Wide Web Conference (WWW 2008)*, pp. 635–644. ACM, New York, NY, USA.

David Yarowsky (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING 1992)*, pp. 454–460. ACL, Morristown, NJ, USA.

Zhiping Zeng, Anthony K. H. Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou (2009). Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1):25–36.

Torsten Zesch and Iryna Gurevych (2006). Automatically creating datasets for measures of semantic relatedness. In *COLING/ACL 2006 Workshop on Linguistic Distances*, pp. 16–24. Sydney, Australia.

Xiang Zhang, Hongda Li, and Yuzhong Qu (2006). Finding important vocabulary within ontology. In *Proceedings of the 1st Asian Semantic Web Conference (ASWC 2006)*, volume 4185 of *LNCS*. Springer.

Index

F_1 measure, 56
F_β measure, 56

accuracy, 56
ambiguity, 30
antonymy, 63
approximation algorithm, 96
approximation guarantee, 96, 104
arc, 23
arc weighting, 45
assignment problem, 83

candidate arc, 39
cost, 90
cross-lingual classification, 71

distinctness assertion, 88, 136

entity, 18
entity identifier, 18
entity integration, 77
entity linking, 83
entity reconciliation, 83
entity search, 159
equals relation, 24, 79, 127
etymology, 13

F-score, 56
fractional solution, 98

gloss, 25, 63
graph cut, 84

Hungarian algorithm, 83
hypernymy, 63

in-neighbourhood, 18
independent edge set, 83
instance relation, 25, 132
interwiki link, 79

knowledge base, 12

language, 113
lexical category, 47
lexical integration, 27
lexical knowledge, 2, 28, 152
lexical knowledge base, 12
lexicalization, 24
linear program, 96, 102, 107
linear sum assignment problem, 83
Linked Data, 16, 158
linking function, 126

Markov chain, 123, 139
matching, 83
means relation, 23
MENTA, 118

meronymy, 63
multilingual knowledge, 2
multilingual knowledge base, 34
multilingual knowledge bases, 122

negative, 55
NFC, 20
node, 19
NP-hard, 94

ontology, 14
out-neighbourhood, 18

parallel corpus, 37
parents, 138
positive, 55
precision, 55

recall, 55
record linkage, 83
region, 99
regression, 41
relational database, 16
Roget's Thesaurus, 13, 65

semantic entity, 19, 21
semantic knowledge, 1
semantic node, 19, 21, 77
semantic relatedness, 50, 68
Semantic Web, 15
stable marriage, 83
subclass relation, 24, 130
SUMO, 157

taxonomic integration, 117
taxonomy, 117
taxonomy induction, 133
term, 19, 20
term entity, 19
text classification, 71
thesaurus, 13, 36, 65, 67, 77, 83
translation, 24, 36
triangulation, 38

Unicode, 20
upper-level ontology, 156
user interface, 154
UWN, 28

WDGS, 92
Weighted Distinctness-Based Graph Separation, 92
weighting function, 45
Wikipedia, 78, 121
Wiktionary, 36
WordNet, 12, 31
wordnets, 31

Given that much of our knowledge is expressed in textual form, information systems increasingly depend on knowledge about words and the entities they represent. This book investigates novel methods for automatically building large repositories of knowledge that capture semantic relationships between words, names, and entities, in many different languages. Three major new contributions are presented, each involving graph algorithms and statistical techniques that combine evidence from multiple sources of information.

The lexical integration method involves learning models that disambiguate word meanings based on contextual information in a graph, thereby providing a means to connect words to the entities that they denote. The entity integration method combines semantic items from different sources into a single unified registry of entities by reconciling equivalence and distinctness information and solving a combinatorial optimization problem. Finally, the taxonomic integration method adds a comprehensive and coherent taxonomic hierarchy on top of this registry, capturing how different entities relate to each other.

Together, these methods can be used to produce a large-scale multilingual knowledge base semantically describing over 5 million entities and over 16 million natural language words and names in more than 200 different languages.