Knowledge Acquisition in the Information Age

The Interplay between Lexicography and Natural Language Processing

Luis Espinosa-Anke

TESI DOCTORAL UPF / ANY 2017

DIRECTOR DE LA TESI
Horacio Saggion
Departament DTIC

To Carla,
La estrella que iluminó mi camino
cuando más ciego estuve.

Acknowledgements

When I wrote the Acknowledgements section of my 2-year Master's thesis I had to cut it down a lot after I had finished. There were so many influencers I wanted to explicitly thank for helping me in that journey that I went way beyond 2 pages. Imagine how hard it is to put these acknowledgements together when it comes to a 4-year PhD program.

Let's start from the beginning, then. First and foremost, I would like to express my sincerest gratitude to professor Horacio Saggion. I always considered myself a lucky linguist who got into an NLP program thanks to Horacio being adventurous enough to agree to supervise me and provide all the support I needed since the beginning. I am right now a better researcher, better programmer, write better and have better methodological skills than I would have ever imagined, and the seed for all this was planted by prof. Saggion. It has been an honour to be his PhD student and I hope this is the beginning of many more future collaborations.

I would also like to thank all those who have co-authored a paper with me in the last years. Starting with Horacio, but also Francesco Ronzano, Claudio Delli Bovi, Mohamed Sordo, Xavier Serra, Sara Rodríguez Fernández, Aonghus Lawlor, Ahmed AbuRa'ed, José Camacho-Collados, Francesco Barbieri, the guys from Savana (Jorge, Alberto, Nacho Medrano, Nacho Salcedo) and Alberto Pardo.

I would also like to explicitly thank professor Roberto Navigli for hosting me in 2015 in the Linguistic Computing Laboratory, at the Sapienza University of Rome. My research career and publication record would not be what it is today without having had this opportunity. I find the BabelNet project a seminal work in knowledge representation, and I plan to continue contributing to it. By extension, I also would like to thank the European Network of e-Lexicography for funding my stay, specifically professors Simon Krek, Carole Tiberius, Tanneke Schooenheim and Ilan Kernerman, with whom I have had very invigorating conversations.

Very special thanks also to professor Leo Wanner, with whom I have had a lot of discussions on language, lexicography and compositional meaning. I have always felt warmly welcome at the TALN group, and Leo has made this possible with his expert feedback, and also helping me in everything I needed.

Of course, a very special mention to Francesco Barbieri, Joan Soler, Sergio Oramas, Miguel Ballesteros, Alp Öktem and Roberto Carlini. These last years have been a lot of fun, without you guys this would have probably not been the same.

And finally, I wish to thank my wife Carla and my parents Luis and Vibeke for putting up with me during all these years, for being supportive and for trying their best to motivate me when the light at the end of the tunnel could barely be seen. All the good things I have done in life are for the most part because you have given me love and taught me what really matters in life. I love you.

Abstract

Natural Language Processing (NLP) is the branch of Artificial Intelligence aimed at understanding and generating language as close as possible to a human's. Today, NLP benefits substantially of large amounts of unnanotated corpora with which it derives state-of-the-art resources for text understanding such as vectorial representations or knowledge graphs. In addition, NLP also leverages *structured* and *semi-structured* information in the form of ontologies, knowledge bases (KBs), encyclopedias or dictionaries. In this dissertation, we present several improvements in NLP tasks such as Definition and Hypernym Extraction, Hypernym Discovery, Taxonomy Learning or KB construction and completion, and in all of them we take advantage of knowledge repositories of various kinds, showing that these are essential enablers in text understanding. Conversely, we use NLP techniques to create, improve or extend existing repositories, and release them along with the associated code for the use of the community.

Resumen

El Procesamiento del Lenguaje Natural (PLN) es la rama de la Inteligencia Artificial que se ocupa de la comprensión y la generación de lenguage, tomando como referencia el lenguaje humano. Hoy, el PLN se basa en gran medida en la explotación de grandes cantidades de corpus sin anotar, a partir de los cuales se derivan representaciones de gran calidad para la comprensión automática de texto, tales como representaciones vectoriales o grafos de conocimiento. Además, el PLN también explota información *estructurada* y *parcialmente estructurada* como ontologías, bases de conocimiento (BCs), enciclopedias o diccionarios. En esta tesis presentamos varias mejoras del estado del arte en tareas de PLN tales como la extracción de definiciones e hiperónimos, descubrimiento de hiperónimos, inducción de taxonomías o construcción y enriquecimiento de BCs, y en todas ellas incorporamos repositorios de varios tipos, evaluando su contribución en diferentes áreas del PLN. Por otra parte, también usamos técnicas de PLN para crear, mejorar o extender repositorios ya existentes, y los publicamos junto con su código asociado con el fin de que sean de utilidad para la comunidad.

Contents

List of Figures

List of Tables

Chapter 1

PREAMBLE

1.1 Text is the Biggest of Data

Data is "the new oil", a kind of highly valuable raw material, which, if unrefined, cannot really be used [Palmer, 2006]. The intelligent exploitation of data has become a driving force in areas of diverse nature, such as healthcare [Raghupathi and Raghupathi, 2014], energy [Kezunovic et al., 2013] or biology [Howe et al., 2008]. This context, and the increasing rate at which information is created, published, and shared online, has led to the coinage of **big data**, a term used to describe datasets which follow the "three Vs" definition, which alludes to notions of Volume, Velocity and Variety [Ward and Barker, 2013]. Thus, we usually consider *big* datasets to be either *too big*, *too complex* and/or *too fast moving* to be easily analyzed with traditional data processing applications.

In this scenario, **text data** plays a major role. In a study conducted in 2016[1], it was shown that *every minute*, Google translates almost 70M words, SIRI answers nearly 100k requests, and more than 3.5M text messages are sent in the US alone. These figures are certainly impressive. However, how to automatically make sense of all the textual information that is generated daily remains an unsolved problem, mostly due to the fact that text is the embodiment of the challenges associated to big data. First, text data is *very big* (the English Wikipedia alone grows at a rate of 800 articles a day, with 10 edits per second[2]). It is also clearly moving very fast, with novel devices and communicative settings blossoming at a faster pace than ever. But most importantly, and what constitutes the main motivation behind this dissertation, is that text data is too complex to be processed without taking into account meaning, ambiguity or communicative context. The automatic processing of text requires a substantial effort in formalizing the *underlying semantics*

[1] www.domo.com/blog/data-never-sleeps-4-0/

[2] en.wikipedia.org/wiki/Wikipedia:Statistics

contained in language, a task which concerns a research area known as Natural Language Processing (NLP), which in turn is one of the long-lasting problems in Artificial Intelligence (AI).

NLP is concerned with automatic text understanding, a broad and complex problem, which has branched out into smaller and more tractable tasks, where specific linguistic phenomena are addressed as standalone problems, often motivated "by specific applications or by our belief that they capture something more general about natural language" [Collobert et al., 2011]. Complexity in natural language can be mostly attributed to the notion of *ambiguity*, which not only affects individual words or phrases, but rather occurs comprehensively across all levels of linguistic description. Indeed, it is widely acknowledged among NLP specialists that "most or all tasks in speech and language processing can be viewed as resolving ambiguity" at any of these levels [Jurafsky and Martin, 2000].

This means that ambiguity, as the backbone of creative language production, has to be addressed by computational models at all its layers. For instance, for resolving *syntactic ambiguity*, a system must *parse the sentence* and derive a syntactic tree from which an interpretation can be obtained in terms of words, phrases and the relations among them. To provide the reader with an illustrative example, let us refer to an extensively studied type of syntactic ambiguity: the PP-attachment[3] ambiguity, consisting in selecting, in a sentence like "he saw the woman with the telescope", who was actually using a telescope ("he" or "the woman") [Hindle and Rooth, 1993]. A different type of ambiguity, namely *lexical ambiguity*, occurs when words can be interpreted in multiple ways depending on the context in which they appear. When faced with a problem of this nature, a computational model has to perform what is known as Word Sense Disambiguation (WSD), i.e. the computational identification of meaning for words in context [Navigli, 2009].

The above are two of the many examples in which a case of ambiguity can be resolved by relying to a certain extent on predefined knowledge encoded and stored in a language resource, or *knowledge repository* (KR). For example, syntactic parsing systems are trained on large amounts of annotated corpora, whereas WSD systems may look up available *senses* (or meanings) associated to a single lemma, and then decide which is the most appropriate for a specific communicative context. These resources, with machine-readable corpora as the earliest and best known example, started to become widely available several decades ago. Their potential impact was soon foreseen in fields like lexicography, the discipline concerned with the principles and methods of writing dictionaries [Bowker, 2003]. For example, in [Church and Hanks, 1990], a corpus-based novel metric for measuring association of words was introduced (the today well-known *mutual infor-*

[3]PP stands for *prepositional phrase*.

mation), and rapidly it became a staple for evaluating multiword expressions such as collocations or compounds. Similarly, FrameNet[4], a manually annotated lexical database where meaning is associated to *semantic frames*, and which seeded most of today's NLP tasks on Semantic Role Labeling [Gildea and Jurafsky, 2002], was originally envisioned as "a lexicographic project" with applications to text understanding [Fillmore and Baker, 2001].

KRs are useful in NLP because they can store meanings of words and phrases, relations of any kind (e.g. syntactic, syntagmatic, semantic or ontologic) holding among them, and also descriptions about entities or common sense facts. This information is essential in tasks requiring any degree of text understanding, and unsurprisingly, NLP has traditionally leveraged whatever knowledge that was made available in these resources. For example, they have played a major role in tasks such as query expansion [Graupmann et al., 2005], semantic search [Auer et al., 2007], clinical decision support systems [Demner-Fushman et al., 2009], ontology learning [Lonsdale et al., 2002], automatic summarization [Reimer and Hahn, 1988], or distributional semantics [Faruqui et al., 2015, Camacho-Collados et al., 2015].

KRs may differ in coverage and depth, may be purpose-built or domain-specific, or may even be designed to capture comprehensive world knowledge in fine detail [Medelyan et al., 2013]. And while today the number and quality of these resources is certainly impressive, the fact that many of them are originally a (usually collaborative) manual effort, results in drawbacks such as scalability [Laparra et al., 2010], as these resources are not designed to evolve at par with today's information and knowledge creation. This scenario has motivated the inception of a research area in which NLP and machine learning techniques are applied for automatically creating or enriching KRs, from annotated corpora to more structured resources like lexicons, terminological databases, dictionaries, taxonomies or ontologies. This dissertation is concerned precisely with bridging the gap between today's KRs and NLP in a twofold fashion. On the one hand, by improving the state of the art in a variety of NLP tasks by leveraging information at various degrees of structuring (from lexical databases to dictionaries or simply text corpora), and on the other, by extending (and also creating from scratch) KRs via NLP techniques.

In what follows, we provide the reader with a broad picture on KRs, inspired by the original classification proposed in [Hovy et al., 2013]. We start by introducing basic notions on *how knowledge is defined and represented* (Section 1.2), and continue by fleshing out different types of KRs, namely structured (Section 1.2.1), unstructured (Section 1.2.2) and semistructured (Section 1.2.3) resources. Finally, we conclude this preamble with a conclusion and a critical view on the

[4] framenet.icsi.berkeley.edu/fndrupal

current state of knowledge representation, together with specific research goals pursued in this thesis (Section 1.3).

1.2 Defining and Representing Knowledge

Today, AI is facing challenges related to "the representation of linguistically expressible knowledge, the role of knowledge in language understanding, the use of knowledge for several sorts of commonsense reasoning, and knowledge accumulation" [Schubert, 2006]. The keyword 'knowledge' plays a crucial role in NLP and AI. In fact, the idea of feedings intelligent systems and agents with general, formalized knowledge of the world dates back to classic AI research in the 1980s [Russell and Norvig, 1995].

Thus, it is important to have a broad and clear picture of what kinds of KRs exist today, the process behind their construction (i.e., their *knowledge acquisition* pipeline), and how they are leveraged in NLP. According to [Hovy et al., 2013], we may distinguish between **structured resources** such as Knowledge Bases (KBs) or dictionaries, **unstructured resources** (such as statistical models derived from text corpora, or simply corpora), or **semi-structured resources** (Wikipedia probably being the best known example). We thus will build up on this original classification to provide a critical review of the kinds of KRs that are currently available today both from the perspective of end users as well as NLP practitioners. Specifically, we argue that lexicographic information, encoded in the form of textual definitions, (which are the only *quasi*-unstructured content in otherwise well structured resources like dictionaries or glossaries), has received little attention from the NLP community, and that it can be processed and reshaped as fully machine-readable information thanks, among others, to the algorithmic novelties we present in this thesis. In fact, it has the potential to affect dramatically current NLP applications, as it provides vast amounts of highly reliable knowledge accumulated over time.

Finally, at the end of the survey, we state the research goals of this dissertation, and anticipate our specific contributions, both in terms of experimental results and language resources and software.

1.2.1 Structured Knowledge Resources

Manually-crafted fully structured resources undoubtedly represent knowledge at the highest level of quality. They usually encode information entered by domain experts, lexicographers or ontologists, and can be further leveraged with high confidence by intelligent NLP systems. Although the terminology used to refer to some of these resources is broad (e.g. WordNet [Miller et al., 1990] has been

referred to as a lexical database, an ontology and also a lexicalized knowledge base), we distinguish three broad groups of structured resources, namely *lexicographic resources*, *lexical databases and thesauri*, and *knowledge bases*. Let dive into each of them.

1.2.1.1 Lexicographic Resources

Defining a concept by making use of expressions other than the one mentioning said concept is acknowledged to be one of the most valuable functions of language [Barnbrook, 2002]. While the interest in definitions dates back to Aristotelian times [Granger, 1984], their origin may be even more remote, and lexicographic information may be traced as far back as to all societies with writing systems, and some without [Béjoint, 1994].

Dictionaries are the paramount example of lexicographic resources, and they may serve varying goals. From monolingual to bilingual dictionaries, these may be designed for general purposes or domain-specific. Dictionaries are, by definition, human readable, and provide a listing of concepts and their associated *senses*, usually together with pronunciation, definitions, or examples or their lexical combinations (e.g. collocations).

Naturally, dictionaries have a place in history much earlier than computers ever existed, and therefore were not designed to fulfill a computational purpose. However, although they are "far from ideal for computer use, they represent an investment of resources that the computational linguistics research community is in no position to match" [Sampson, 1990]. For this reason, dictionaries play an important role in NLP, and even constitute the core of dictionary-based NLP, a subarea in NLP concerned with the exploitation of dictionaries for improving NLP tasks. These have proven useful in, among others, recognizing biomedical concepts in free text [Xu et al., 2008], part-of-speech tagging [Coughlin, 1999], or machine translation [Chowdhury, 2003]. Today, traditional dictionaries are gradually being converted to machine-readable forms to develop substantial lexicons for NLP in a resource-efficient fashion [Briscoe, 1991].

The main content of dictionaries are *definitions*, which are essential resources to consult when the meaning of term is sought [Park et al., 2002, Navigli and Velardi, 2010]. While there is a considerable amount of work on the philosophical and linguistic motivations behind certain classifications or taxonomies for definitions, we provide in what follows simply a succint survey. Most definition classification proposals are based on the *genus et differentia* model, coined by Aristotle. In this model, the structure of a definition resembles an equation, where the *definiendum* (the term that is being defined) is placed on the left, and the *definiens* (the cluster of words that differentiates the definiendum from others of its kind) is placed on the right. This definiens is made of two parts: *genus*

(the nearest general concept), and the *differentiae specificae* (the definiendum's differentiating characteristics) [Del Gaudio and Branco, 2009].

Definitions have been classified from several standpoints. For instance, (1) based on their degree of formality and informativeness [Trimble, 1985]; (2) according to their *purpose* [Robinson, 1972]; (3) according to the *method* followed to define a concept [Borsodi, 1967]; (4) according to the textual pattern used in the definitions [Westerhout and Monachesi, 2007b]; or (5) with regard to how definitional information is conveyed [Sierra et al., 2003, Aguilar et al., 2004, Sierra et al., 2006a]. An extensive overview of definition classification is provided in [Westerhout, 2010]. These definitional classification schemes, however, still have not been incorporated into standardized NLP tasks. In fact, those lexicographic repositories that are most used in NLP (and which do not follow a lexicographically motivated organization) are[5]:

- **Wiktionary**:[6] It currently constitutes the largest available collaboratively constructed lexicon for linguistic knowledge (see [Meyer and Gurevych, 2012] for a discussion on Wiktionary's quality and its role in 21st century lexicography), and has been used in several NLP applications [Etzioni et al., 2007, Müller and Gurevych, 2008, Zesch et al., 2008, Schlippe et al., 2010].

- **Urban Dictionary**:[7] Given the increasing interest in modeling the language used in social networks, where jargon and slang are frequently utilized, Urban Dictionary has been used as a reference dictionary for sentiment analysis [Wu et al., 2016].

- **Domain-Specific Dictionaries**: Similarly as in ontology engineering, domain-specific dictionaries are still frequently used for modeling specific domains of knowledge, which are usually either of interest for a specialized minority, or belong to highly specialized domains. Examples are diverse, and range from dictionaries of soccer [Dunmore, 2011] to dictionaries of epidemiology [Last et al., 2001] or human geography [Johnston, 1981]. These focused dictionaries have been taken advantage of for tasks such as semantic taxonomy enrichment [Jurgens and Pilehvar, 2016].

There are certainly other specialized electronic lexicographic resources such as dictionaries in languages others than English, bilingual dictionaries, which are designed to help users understand, produce and translate texts [Nielsen, 1994, Nielsen, 2010], or multilingual dictionaries[8], but their impact in current NLP has

[5] We intentionally leave WordNet out of this list, as it will be discussed in Section 1.2.1.2.

[6] www.wiktionary.org

[7] www.urbandictionary.com

[8] Such as `kdictionaries.com`.

yet to be measured, as many of them are only now slowly being made available to the research community. Finally, as for specialized linguistic phenomena, dictionaries may include synonyms, pronunciations, names (place names and person names), phrases and idioms, dialect terms, slang, quotations or etymologies [Cowie, 2009].

1.2.1.2 Lexical Databases and Thesauri

Two of the best known generic lexical databases and thesauri are Roget's Thesaurus [Roget, 1911] and WordNet. In what follows we describe their main features, as well as applications in NLP.

WordNet

WordNet [Miller et al., 1990, Miller, 1995, Fellbaum, 1998] represents *senses* by grouping together synonyms referring to the same idea or concept. In WordNet, there are more than 118,000 word forms and more than 90,000 word senses. Approximately 17% of the words contained in WordNet are polysemous, and roughly 40% of them have one or more synonyms. In WordNet, word forms like "back" or "right", which can have different parts of speech depending on their usage, are represented differently. Additionally, derivational and compound morphology are not considered, and hence forms like "interpretation", "interpreter" or "misinterpret" are all considered distinct words. A highly exploited feature of WordNet in NLP is the fact that it encodes *semantic relations* among synsets. Specifically, synsets are related in terms of *synonymy*, *antonymy* (opposite), *hyponymy* (subordinate), *meronymy* (part), *troponymy* (manner), and *entailment* (the last two relations being relevant only for verbs). As for its influence in NLP, it is indisputable that it has played a major role in improving a wide range of tasks where lexical knowledge is needed. In fact, the list of research papers using WordNet seems endless [Hovy et al., 2013].

Roget's Thesaurus

Roget's provides a well-constructed concept classification, and features entries written by professional lexicographers. One of its main advantages is its topical distribution. As an example, according to [Jarmasz and Szpakowicz, 2004], Roget's allows linking the word *bank*, the business that provides financial services, and the verb *invest*, i.e. to give money to a bank to get a profit, by placing in the common head *lending*. Taking advantage of these topical characteristics, a number of NLP systems have benefited from Roget's for tasks like sentiment analysis [Aman and Szpakowicz, 2008]; computing semantic similarity and lexical cohesion [McHale, 1998, Morris and Hirst, 1991]; or WSD [Yarowsky, 1992]. Finally,

let us highlight the fact that, within the area of lexical access, Roget's Thesaurus is considered to be among the few resources that may be of help for language *producers* rather than *receivers* (readers or listener). See, e.g. [Zock and Bilac, 2004, Zock and Schwab, 2008].

1.2.1.3 Knowledge Bases

KBs are "relational databases together with inference rules, with information extracted from documents and structured sources" [Ré et al., 2014]. Generally, we expect KBs to be graph-like data structures where each node represent an entity or concept (e.g. *Nintendo* or *hope*), and where edges between nodes may express WordNet-like semantic relations, but also ontologic relations such as `is-based-in` or `is-CEO-in`. KBs are essential in any knowledge-centric approach and cognitive application, e.g. disambiguation, semantic search for entities and relations in web and enterprise data, and entity-oriented analytics over unstructured contents [Suchanek and Weikum, 2013]. While a comprehensive review of KBs is out of the scope of this dissertation, we provide in what follows a summary review of the most relevant KBs in terms of coverage and/or applicability. We break them down in three separate groups according to the methodology followed to construct them, namely *fully manual* (either crowdsourced or by domain experts), *semi automatic* (usually by performing automatic mappings or alignments among manually built resources), and *fully automatic*, i.e. performing an unrestricted acquisition and ranking of facts.

Manually built KBs

Manually built KBs can be broadly classified in two groups. In the first group we find **domain-specific KBs**, usually the result of input by domain experts and knowledge engineers. Systems containing only special-purpose domain knowledge have accomplished extraordinary goals in a variety of fields [Matuszek et al., 2006]. Indeed, the number of structured manually built KBs that exist in specialized areas of knowledge is very high, and some of them have trascended their own domain by becoming core elements in research in knowledge representation. Prominent examples include the Chemistry domain (*CheBi*[9]) [Degtyarenko et al., 2008], Genetics (*GeneOntology*[10]) [Ashburner et al., 2000], Medicine (*Snomed*[11]) [Spackman et al., 1997] or Music (*MusicBrainz*[12]) [Swartz, 2002]. However, the truth is that in general, these resources are difficult to adapt to unforeseen prob-

[9] www.ebi.ac.uk/chebi/
[10] geneontology.org/
[11] browser.ihtsdotools.org
[12] musicbrainz.org/

lems or areas. They may be too brittle for robust exportation [Friedland et al., 2004], which is particularly impactful in any area involving natural language interaction such as Question Answering, where the extension of the problem space is very difficult to define a priori [Hovy et al., 2002].

The second group includes **general-purpose KBs**, which were originally envisioned as repositories which would encode vast amounts of information of any kind (a sort of comprehensive world knowledge), from common sense knowledge (e.g. "humans don't like to get hurt") to millions of facts about the world (e.g. "the capital of Spain is Madrid"). Let us describe some of the most outstanding general-purpose manually built KBS.

- **Cyc**: Cyc[13] is a large KB containing a store of formalized background knowledge [Matuszek et al., 2006]. Parts of this project are released as OpenCyc, which provides an API, RDF endpoint, and a data dump under an open source license. While it was initially designed as a fully manual enterprise, it has strong focus in enabling NLP applications, and also to recently to take advantage of NLP techniques for growing and improving its ontology[14].

- **Freebase**: Freebase is a tuple database used to structure human knowledge [Bollacker et al., 2008]. The data it contains is collaboratively created, structured and maintained. Originally, it was possible to access Freebase information via an HTTP based graph-query API. However, since August 2016, the Freebase project was discontinued and all its information was ported to Wikidata [15].

- **Wikidata**: Wikidata [Vrandečić and Krötzsch, 2014] is a document-oriented semantic database operated by the Wikimedia Foundation[16] with the goal of providing a common source of data that can be used by other Wikimedia projects.

Semi-automatically built KBs

We may define a semi-automatically built KB as any integrative project that combines knowledge derived from manual interaction with automatic modules aimed at either providing n-to-n mapping across different resources, by ontologizing unstructured knowledge, or by leveraging human input for refinement and pruning out incorrect facts. Outstanding cases include:

[13] www.opencyc.org
[14] www.cyc.com/natural-language-processing-in-cyc/
[15] plus.google.com/109936836907132434202/posts/bu3z2wVqcQc
[16] www.wikidata.org

- **ConceptNet**: ConceptNet[17] is defined as a "flexible, multilingual semantic network for common sense knowledge" [Havasi et al., 2007]. It combines automatic modules which perform pattern matching or reliability scoring with manual interaction with users of the Open Mind Common Sense project[18], the umbrella infrastructure under which ConceptNet lives. Recently, ConceptNet has also been leveraged, for instance, for learning vectorial representation of concepts and relations present in this KB [Speer et al., 2016].

- **BabelNet**[19]: BabelNet [Navigli and Ponzetto, 2012] is a large multilingual semantic network which originally was envisioned as a mapping between Wikipedia and WordNet. However, it currently integrates additional resources such as Wiktionary, OmegaWiki and Wikidata. It also features its own taxonomical organization, after taxonomizing the cyclic and dense graph formed by Wikipedia page links and categories [Flati et al., 2014]. BabelNet has been exploited for many NLP applications, such as joint WSD and Entity Linking (EL) [Moro et al., 2014], or distributed representations of disambiguated linguistic items [Camacho-Collados et al., 2015, Iacobacci et al., 2015].

- **Yago**[20]: Yago is an ontology based on Wikipedia, from which it benefits from category pages, which in turn subsume entity pages. This categorical information (rich but noisy and hardly usable) is combined with more constrained but also more precise information from WordNet. Yago is described as having "near-perfect accuracy" (97%) [Suchanek et al., 2007] , and has been enriched with *n-ary* relations, i.e. relations holding among more than two entities.

- **DBpedia**[21]: DBpedia [Auer et al., 2007] is a community effort to extract structured information from Wikipedia and to make this information available on the Web. It allows for sophisticated queries against datasets derived from Wikipedia, and also to link other datasets on the Web to Wikipedia data.

[17] `conceptnet.io`
[18] `www.media.mit.edu/research/groups/5994/open-mind-common-sense`
[19] `babelnet.org`
[20] `yago-knowledge.org/`
[21] `wiki.dbpedia.org/`

Automatically built KBs

In the third group we refer to approaches where the main breaking point from the above cases is that *all* knowledge is obtained and structured automatically. These usually fall within the so-called *Open Information Extraction* (OIE) paradigm [Banko et al., 2007], which roughly speaking can be summarized as (1) reading the web; (2) learning facts; (3) scoring them; and (4) structuring them according to some semantic criterion. While there exist minor differences in design (e.g. NELL [Carlson et al., 2010] maps all facts to a manually predefined ontology, while ReVerb [Fader et al., 2011] is purely unconstrained), the vision behind all OIE systems is to acquire knowledge in an unrestrained fashion from fully unstructured data (web documents). In the following we discuss two text-level OIE systems, namely TextRunner [Banko et al., 2007] and ReVerb; and four semantic-level (mapping facts to a reference KR) systems: NELL, PATTY [Nakashole et al., 2012], DefIE [Delli Bovi et al., 2015] and Knowledge Vault [Dong et al., 2014]. We also cover DeepDive [Niu et al., 2012, Zhang et al., 2016], which performs large-scale OIE not only on running text, but also leverages semi-structured information from Wikipedia, as well as other resources such as Freebase.

- **TextRunner**: TextRunner is the first OIE project [Etzioni et al., 2011]. It consists of three modules: A self-supervised learner which generates a classifier given a small corpus sample; a single-pass extractor that captures candidate tuples, which are further labeled as trustworthy or not by the classifier; and a redundancy-based assessor, which assigns a probability to each tuple based on its redundancy in text.

- **ReVerb**: ReVerb[22] is an OIE approach designed to, among other capabilities, reduce noise in the form of incoherent and uninformative extractions (thus improving certain weaknesses identified in its predecessor TextRunner). This is achieved thanks to a set of syntactic and lexical constraints. The former enforces an extracted relation to comply with one of the set of predefined POS-level patterns. Then, the latter is used to preserve only those relations which are general and thus informative enough to be included into the extracted learned facts. However, ReVerb's technology remains incapable to cope with certain linguistic phenomena such as *n-ary* or nested relations.

- **NELL**: NELL (Never Ending Language Learning) is an OIE system developed at Carnegie Mellon University[23]. It differs e.g. from ReVerb in that, in NELL, there is a starting ontology of 271 relations against which

[22] `reverb.cs.washington.edu/`
[23] `rtw.ml.cmu.edu/rtw/`

each extracted fact is validated and eventually introduced into the resulting KB. For example, it is possible to extract only the subset of taxonomic (is-a) relations from a NELL's output due to the fact that there is one specific relation (in NELL's own vocabulary, it is called `generalization`) that encodes such relation.

- **PATTY**: PATTY is an OIE system which incorporates a further semantic level by mapping relation arguments as well as relation types into a predefined KB (e.g. relations are mapped to Wikipedia entities). It is defined as a repository of semantically-typed relation patterns (also known as *relation synsets*)[24].

- **DefIE**: DefIE[25] is a *quasi* OIE system, in that it does not require predefined sets of relations to be extracted (like NELL), but on the other hand, it self-limits its scope on definition sentences from Wikipedia, thus overcoming one of the great challenges posed in OIE tasks, which is the degree of noise usually encountered in processes involving knowledge gathering from the web.

- **Knowledge Vault**: Knowledge Vault is a probabilistic knoweldge base construction system, based on three main architectures: Extractors, for processing large corpora for extracting ⟨subject, predicate, object⟩ triples; Graph-based priors, i.e. using predefined knowledge stored in Freebase or Wikidata as *prior knowledge*; and Knowledge fusion, for ultimately scoring related facts according to a truthfulness and trustiness probability.

- **DeepDive**[26]: DeepDive is an automatic system for KB construction which leverages various resources, e.g. Wikipedia, Freebase or the web. It is built upon modules which perform various tasks, such as syntactic parsing, relation extraction pipelines, distributed computing, markov logic for statistical inference, or EL (combining Named Entity Recognition techniques with matching against Wikipedia pages).

The above survey should provide the reader with a broad picture of the current state of *structured KRs*. As we have seen, it is not mandatory to have extensive human input for building such resources, and in fact the current trend is gradually favouring automatic approaches for improving or extending them. NLP can play a very important role in this area, as intelligent processing of text can contribute dramatically to identifying facts, infer novel truth for them, or discovering novel

[24] `www.mpi-inf.mpg.de/yago-naga/patty/`
[25] `lcl.uniroma1.it/defie/`
[26] `deepdive.stanford.edu/`

domain specific terminology (inventions, patents, drugs or named entities). In the following sections we discuss the two remaining types of KRs, namely fully unstructured and semi-structured resources.

1.2.2 Unstructured Resources

According to [Hovy et al., 2013], text collections are "the main kind of unstructured resource". Despite the fact that corpora provide some kind of organizational structure (e.g. sentences, paragraphs, sections or documents), they do not provide machine-readable knowledge because it is simply encoded as strings of text.

Since corpora are a massive resource in terms of coverage, we will narrow down this subsection by defining corpora as the means for building *distributional (vector) representations of linguistic items* (e.g. word-document frequencies and co-occurrences matrices), arguably one of the hottest areas in recent NLP.

1.2.2.1 Distributed Semantic Models

According to the review provided in [Turney and Pantel, 2010], representing words as vectors in a corpus-driven vector space model has applications in areas such as automatic thesaurus generation [Crouch, 1988, Curran and Moens, 2002], semantic similarity [Deerwester et al., 1990], word clustering [Pantel and Lin, 2002] or query expansion [Xu and Croft, 1996]. These representations, known as Vector Space Models (VSMs) are prominent approaches for representing semantics (and hence, knowledge). They "represent a linguistic item as a vector (or a point) in an n-dimensional semantic space, i.e. a mathematical space wherein each of the dimensions (hence, axes of the space) denotes a single linguistic entity, such as a word" [Camacho-Collados et al., 2016].

VSMs can be broadly categorized as co-ocurrence based or as a newer predictive branch (what we generally know today as word embeddings) [Baroni et al., 2014], the lattter having become a staple in current research in NLP, with highly impactful contributions such as *word2vec* [Mikolov et al., 2013c], *paragraph vector* [Le and Mikolov, 2014], *GloVe* [Pennington et al., 2014], *skip-thoughts* [Kiros et al., 2015], or *Word Mover's Distance* [Kusner et al., 2015]. According to [Baroni et al., 2014], "the buzz is fully justified, as the context-predicting models obtain a thorough and resounding victory against their count-based counterparts". In addition, these advances come hand in hand with the breakthrough of neural models, which have further improved the state of the art in many semantics tasks such as hypernym detection [Shwartz et al., 2016], textual entailment [Yu et al., 2014], dependency parsing [Dyer et al., 2015] or named entity recognition [Lample et al., 2016].

1.2.3 Semi-Structured Resources

[Hovy et al., 2013] suggest that, while other semi-structured resources exist (e.g. Twitter messages[27] or Yahoo! Answers[28]), Wikipedia constitutes "the largest and most popular collaborative, multilingual resource of world and linguistic knowledge containing semi-structured information". Unsurprisingly, Wikipedia is the core KR of many of the previously discussed KBs, and also the source for the result of many datasets based on OIE.

1.2.3.1 Wikipedia

Wikipedia is the largest multilingual encyclopedia in the world, and it has for many years now, been established as a reliable source for lexicographic, encyclopedic and world knowledge. Its quality has been ascertained in studies which go back as far as 2005, where a Nature article showed that Wikipedia "came close" to Encyclopedia Britannica in scientific accuracy[29].

Wikipedia articles are generally organized so that the first sentence constitutes the definition of the page's concept or entity, including etymological and phonetic information. In addition to the text in the article (which as free running text is considered unstructured information), Wikipedia pages are linked to other pages via inner hyperlinks, in addition to belong to a second organizational layer based on Wikipedia Categories. Finally, many Wikipedia pages include infoboxes, which provide domain-specific ontologic information. For example, an infobox about a movie will include information about performing actors and actresses, or if it is the sequel or prequel of another instalment of the saga.

As for its role in NLP, Wikipedia's contribution has been massive. For instance, it has been used for language normalization [Tan et al., 2015]; semantic similarity [Pilehvar et al., 2013]; text simplification [Coster and Kauchak, 2011, Štajner et al., 2015]; or machine translation [Alegria et al., 2013], among many others.

1.3 Conclusion and a critical view

1.3.1 Current limitations

From the previous survey, it stems that up to this date, research in knowledge representation and knowledge acquisition in NLP is producing a large number of high quality resources, pivoting in general around key components such as

[27] www.twitter.com

[28] answers.yahoo.com

[29] www.nature.com/nature/journal/v438/n7070/full/438900a.html

Wikipedia (and its sister projects) or the web. Advances in novel algorithmic approaches (especially in machine learning), along with an increasing awareness of the importance of sharing datasets and language resources, have set the foundations for a vibrant research area. However, certain issues seem to be slightly far reached for the current state of the art. For example, the "middle ground" which [Hovy et al., 2013] argued to be the best of both worlds (big data, and good data), can potentially be increased in size and quality. Today, many state-of-the-art approaches focus on processing larger data with existing NLP technology, e.g. POS-based pattern matching [Fader et al., 2011], dependency-based matching [Niu et al., 2012] or resource-specific heuristics [Auer et al., 2007, Suchanek et al., 2007, Flati et al., 2014], and do not seem to take full advantage of NLP at higher degrees of abstraction or discourse (e.g. semantic similarity, domain pertinence or generalizations).

As for specific resource-wise drawbacks, they can be summarized as follows. Regarding structured resources, they suffer from enormous *creation and maintenance effort*, i.e. they do not scale and are extremely time-consuming to create. Second, they show *lack of coverage*, meaning that they do not cover all knowledge in their target domain (let alone world knowledge in general-purpose endeavours). In addition, there is also the cultural bias, as these resources will inherently include more information from the cultural and historical background of the people who curated them. Third, they are *impossible to keep up-to-date*, which is especially aggravated in dynamic domains such as AI. Finally, the *language barrier* prevents multilingual text processing, as there is very little knowledge encoded in languages other than English, primarily due to the fact that manual input of knowlede implies repeating efforts according to the number of languages targeted.

On the other hand, unstructured resources also show serious problems when it comes to representing knowledge. First, they are not able to automatically acquire all the knowledge required for complex inference chains [Domingos, 2007]. For instance, *dogs have four legs* or *birds can fly* is information that almost never occurs explicitly mentioned in language data[30]. Moreover, there is the issue of the *degree and quality of ontologization*, meaning that systems that perform open-ended fact extraction (such as OIE systems) usually do not have a reference ontology against which all their potentially noisy information can be mapped to. This is an issue that has received certain attention in the recent past, for example how to deal with *unlinkable entities*, i.e. those concepts not prominent enough to have their own entry in reference KRs such as Wikipedia [Lin et al., 2012]. A third problem in unstructured resources, which also affects predictive VSMs, has to

[30]Predictive VSMs, however, seem to have greatly contributed towards this shortcoming, by grouping in vector spaces *semantically similar* concepts, from which these properties may be inferred.

do with the notion of ambiguity, which we defined earlier as the most pervasive problem in automatic language understanding. When projecting linguistic items in vector spaces, the issue of conflated meanings into one vector arises [Pilehvar and Collier, 2016] (e.g. only one vector for the polysemic word 'bank'). Another example can be found in antonyms, i.e. words with opposite semantic meanings ('big' and 'small'), which are often assigned similar vectors due to their tendency to occur in similar contexts [Schwartz et al., 2015].

1.3.2 Research goals

In this dissertation, we propose to advance the state-of-the-art in knowledge acquisition and representation by taking advantage of NLP techniques (both pre existing and those developed and presented in this thesis), and propose to combine them with KRs of varying nature. Specifically, our claim is that high quality lexicographic data such as definitions or topical grouping (knowledge domains) has had negligible impact in recent NLP. We therefore explore the extent to which dictionaries and encyclopedias provide an additional layer in Hovy et al.'s categorization, as they are not as structured as an ontology (since definitions are written in free text, sometimes very creatively), but show a more refined semantic and macrostructural organization than text corpora or Wikipedia. We thus coin the notion of the *virtuous cycle of NLP and lexicography*, which we define as the successful interplay between corpus-based statistical approaches for language representation and semi-structured high quality data encoded in lexicographic material. Based on this idea, we set two main areas of contribution. First, *NLP for lexicography*, where we develop algorithms that perform intelligent text processing tasks for improving the quality of current lexicographic resources. And second, we provide extensive experiments on *lexicography for NLP*, where we investigate how lexicographic and terminological information can be useful for downstream NLP and AI applications.

NLP for Lexicography

Throughout this preamble, we have discussed the strong influence of lexicographic and terminological resources for NLP. We propose to further improve automatically the quality of these resources via NLP. Specifically, the reader of this thesis will find experimental results on:

- **Automatic Extraction of Definitions**: We set our goal in improving the state of the art in *Definition Extraction*, a subtask of Information Extraction consisting in identifying definitional text snippets from corpora, with the ultimate goal to use them for creating or extending automatically existing dictionaries or glossaries.

- **Automatic Hypernym Discovery**: We aim at providing the research community with experimental results, resources and software designed for encoding *is-a* relations between pairs of concepts, which is a useful task, for example, when attempting topical grouping of terms in a dictionary.
- **Collocation Discovery**: We investigate a fairly unexplored area in NLP, which is the automatic acqusition of a very specific type of multiword expression, namely collocations associated by *lexical functions* [Mel'čuk, 1996].

Lexicography for NLP

Our research goals in this area are focused on improving the state-of-the-art in NLP by leveraging lexicographic information of various types, specifically:

- **Taxonomy Learning**: We propose to extensively use definitional information for creating domain-specific lexical taxonomies. We achieve state-of-the-art performance in several evaluation benchmarks in this task.
- **KB Unification**: We put forward novel approaches for improving the quality of existing KRs. Specifically, we study the potential *integration* of arbitrary KRs.
- **Domain-specific KB construction and extension**: An additional goal this dissertation also pursues is exploring difficult subtasks in this area, such as working with highly specific domains or languages other than English. We therefore investigate and evaluate use cases such as the creation of a music-specific KB from scratch, as well as experiments in the medical domain in the Spanish language.

To conclude this preamble, we provide the reader with the structural organization of this thesis.

1.3.3 Organization of the thesis

So far we have provided an extensive discussion of the current state of knowledge representation and automatic knowledge acquisition. We have touched upon how knowledge is today represented in electronic format (Section 1.1), along with the different types of KRs that are most leveraged today in NLP (Sections 1.2.1-1.2.3). Next, we have addressed some of their weaknesses and have set the main research goals that this dissertation pursues (Section 1.3). We finish this preamble by providing the structural organization of this thesis.

First, in Chapter 2, we provide a literature review of the three areas in NLP which are most related to this dissertation from the methodological point of view.

Then, we flesh out our contribution in the area of **definition extraction** (Chapter 3). In Chapter 4, we delve into **hypernym discovery**, and provide a description and evaluation of our proposed systems. We further focus, in Chapter 5, on the task of **taxonomy learning**, where we evaluate a system for jointly providing a semantic and hierarchical articulation to a flat domain terminology. We then present and evaluate different **language resources** which have been created by exploiting in different ways lexicographic knowledge (Chapter 6). We conclude the method and contributions part of the thesis with Chapter 7, where we describe the different **assets** which are published alongside this dissertation, both in terms of **datasets and software**. We finalize this dissertation with a concluding Chapter 8, where we highlight the contributions derived from our work, as well as limitations and potential avenues for future research.

1.4 Our contribution

We have quoted [Jurafsky and Martin, 2000] in that the essence of NLP is to resolve ambiguity at any level of linguistic description (from phonetics and phonology to discourse). In tasks which require high levels of semantic understanding, such as the ones covered in this literature review (definition extraction, hypernym discovery and taxonomy learning), the issue of lexical ambiguity has not been specifically addressed (albeit notable exceptions such as [Kozareva and Hovy, 2010] or [Velardi et al., 2008]). This is a notable drawback in most approaches, as they fall short when attempting, for instance, to find a proper definition in a corpus for the concept *round table* (which could be a table of a circular shape, or a debate among experts in a specific topic), or to discriminate between taxonomic relations involving the term *apple* which are only relevant for the IT industry alone (and not the food domain).

In this dissertation we discuss, in addition to linguistic and purely statistical information, the extent to which *knowledge-based* approaches can contribute to improving the state of the art in these tasks. By leveraging semantic information explicitly encoded in collaboartively-built resources, we are able to model (un)ambiguity in semantic models, which allows us to address issues such as the domain pertinence of taxonomic relations.

Since we take extensive advantage of lexical and semantic resources, we include one additional set of experiments (Chapter 6) where the focus is not to leverage existing KRs for NLP, but rather to use the combination of NLP and lexicographic information as a means to extending, enriching or creating from scratch additional knowledge resources. We will review the methodology behind the following specific use cases: (1) Creation and evaluation of a novel Knowledge Base in the music domain created entirely from scratch (Section 6.1); (2) Unification of

arbitrary outputs of OIE systems into one disambiguated and unified KB (Section 6.2); (3) Extension of the WordNet lexical database with collocational information (Section 6.3); and (4) Extension of the Spanish version of Snomed Clinical Terms, the reference medical terminology (Section 6.4).

In addition to these use cases, we also accompany this dissertation with a number of language resources and software applications (Chapter 7), and showcase their potential for replicating the experiments we report in this dissertation, in addition to allowing any user to perform semantic tasks such as the extraction of definitions from corpora, detection or discovery of hypernyms from either definitions or vector space models, or using lexical taxonomies for inspecting a domain of knowledge.

1.4.1 Publication Record

We provide below a chronologically ordered (most recent first) listing of papers and journal articles successfully published in relevant NLP venues. Most of them are either directly related with the content of this dissertation (in which case, they are marked with the relevant chapter), or indirectly related, e.g. as part of larger research projects.

1. Sergio Oramas, Luis Espinosa Anke, Mohamed Sordo, Horacio Saggion, Xavier Serra: **Information extraction for knowledge base construction in the music domain**. Data and Knowledge Engineering 106: 70-83 (2016) - Chapter 6.1.

2. Luis Espinosa Anke, Jorge Tello, Alberto Pardo, Ignacio Medrano, Alberto Ureña, Ignacio Salcedo, Horacio Saggion: **Savana: A Global Information Extraction and Terminology Expansion Framework in the Medical Domain**. Procesamiento del Lenguaje Natural 57: 23-30 (2016) - Chapter 6.4

3. Sara Rodríguez-Fernández, Luis Espinosa Anke, Roberto Carlini, Leo Wanner: **Semantics-Driven Collocation Discovery**. Procesamiento del Lenguaje Natural 57: 57-64 (2016)

4. Luis Espinosa Anke, Horacio Saggion, Francesco Ronzano, Roberto Navigli: **ExTaSem! Extending, Taxonomizing and Semantifying Domain Terminologies**. AAAI 2016: 2594-2600 - Chapter 5.1

5. Sara Rodríguez-Fernández, Luis Espinosa Anke, Roberto Carlini, Leo Wanner: **Semantics-Driven Recognition of Collocations Using Word Embeddings**. ACL (2) 2016

6. Francesco Barbieri, Luis Espinosa Anke, Horacio Saggion: **Revealing Patterns of Twitter Emoji Usage in Barcelona and Madrid**. CCIA 2016: 239-244

7. Luis Espinosa Anke, Sergio Oramas, José Camacho-Collados, Horacio Saggion: **Finding and Expanding Hypernymic Relations in the Music Domain**. CCIA 2016: 291-296

8. Luis Espinosa Anke, José Camacho-Collados, Sara Rodríguez-Fernández, Horacio Saggion, Leo Wanner: **Extending WordNet with Fine-Grained Collocational Information via Supervised Distributional Learning**. COLING 2016: 3422-3432 - Chapter 6.3

9. Luis Espinosa Anke, José Camacho-Collados, Claudio Delli Bovi, Horacio Saggion: **Supervised Distributional Hypernym Discovery via Domain Adaptation**. EMNLP 2016: 424-435 - Chapter 4.2

10. Sergio Oramas, Luis Espinosa Anke, Aonghus Lawlor, Xavier Serra, Horacio Saggion: **Exploring Customer Reviews for Music Genre Classification and Evolutionary Studies**. ISMIR 2016: 150-156

11. Sergio Oramas, Luis Espinosa Anke, Mohamed Sordo, Horacio Saggion, Xavier Serra: **ELMD: An Automatically Generated Entity Linking Gold Standard Dataset in the Music Domain**. LREC 2016

12. Sara Rodríguez-Fernández, Roberto Carlini, Luis Espinosa Anke, Leo Wanner: **Example-based Acquisition of Fine-grained Collocation Resources**. LREC 2016

13. Francesco Ronzano, Ahmed AbuRa'ed, Luis Espinosa Anke, Horacio Saggion: **TALN at SemEval-2016 Task 11: Modelling Complex Words by Contextual, Lexical and Semantic Features**. SemEval@NAACL-HLT 2016: 1011-1016

14. Luis Espinosa Anke, Francesco Ronzano, Horacio Saggion: **TALN at SemEval-2016 Task 14: Semantic Taxonomy Enrichment Via Sense-Based Embeddings**. SemEval@NAACL-HLT 2016: 1332-1336

15. Luis Espinosa Anke, Roberto Carlini, Horacio Saggion, Francesco Ronzano: **DefExt: A Semi Supervised Definition Extraction Tool**. Globalex Workshop, Co-located with LREC 2016.

16. Luis Espinosa Anke, Francesco Ronzano, Horacio Saggion: **Hypernym Extraction: Combining Machine-Learning and Dependency Grammar**. CICLing (1) 2015: 372-383 - Chapter 4.1

17. Claudio Delli Bovi, Luis Espinosa Anke, Roberto Navigli: **Knowledge Base Unification via Sense Embeddings and Disambiguation**. EMNLP 2015: 726-736 - Chapter 6.2

18. Sergio Oramas, Mohamed Sordo, Luis Espinosa Anke, Xavier Serra: **A Semantic-Based Approach for Artist Similarity**. ISMIR 2015: 100-106

19. Mohamed Sordo, Sergio Oramas, Luis Espinosa Anke: **Extracting Relations from Unstructured Text Sources for Music Recommendation**. NLDB 2015: 369-382

20. Luis Espinosa Anke, Horacio Saggion, Francesco Ronzano: **Weakly Supervised Definition Extraction**. RANLP 2015: 176-185 - Chapter 3.4

21. Luis Espinosa Anke, Horacio Saggion and Claudio Delli Bovi. **Definition Extraction Using Sense-Based Embeddings**. Proceedings of the 2015 International Workshop on Embeddings and Semantics (IWES), pages 10-15, Alicante, Spain - Chapter 3.2

22. Luis Espinosa Anke, Horacio Saggion, Francesco Ronzano: **TALN-UPF: Taxonomy Learning Exploiting CRF-Based Hypernym Extraction on Encyclopedic Definitions**. SemEval@NAACL-HLT 2015: 949-954

23. Sergio Oramas, Mohamed Sordo, Luis Espinosa Anke: **A Rule-Based Approach to Extracting Relations from Music Tidbits**. WWW (Companion Volume) 2015: 661-666

24. Luis Espinosa Anke, Horacio Saggion: **Descripción y Evaluación de un Sistema de Extracción de Definiciones para el Catalán**. Procesamiento del Lenguaje Natural 53: 69-76 (2014) - Chapter 3.3

25. Luis Espinosa Anke, Horacio Saggion: **Applying Dependency Relations to Definition Extraction** NLDB 2014: 63-74 - Chapter 3.1

Chapter 2

RELATED WORK

In order to provide a thorough, precise and relevant overview of those contributions which are most relevant to this dissertation, we will narrow down our literature review to those topics where the interaction between terminology and lexicography, on one hand, and NLP and AI, on the other, is most clear.

We start, thus, by reviewing prominent work in the area of **Definition Extraction**, an NLP task aimed at automating the process of finding definitions from corpora. Then, we focus on **Hypernym Discovery**, i.e. the task of finding, for a given concept, its most likely hypernym(s). This is a very important task in NLP due to the need of semantic search or Question Answering systems to be able to *disambiguate* and *generalize* mentions of concepts or entities. Finally, we focus on arguably the most difficult task of the three, namely **Taxonomy Learning**. It consists in deriving, from a large collection of documents, a hierarchical representation of concepts in a given domain. The resulting graph can be used as a reference knowledge resource in inference tasks, as the backbone of ontologies [Navigli et al., 2011].

2.1 Definition Extraction

Definition Extraction (DE) is the task to automatically identify definitional text fragments (definitions, in short) in corpora. Automatically extracting definitional knowledge has the potential to impact downstream applications such as automatic glossary construction [Muresan and Klavans, 2002, Park et al., 2002, Faralli and Navigli, 2013] and terminological databases [Nakamura and Nagao, 1988], Question Answering systems [Saggion and Gaizauskas, 2004a, Cui et al., 2005], as support for terminological applications [Meyer, 2001, Sierra et al., 2006a], e-learning [Westerhout and Monachesi, 2007b], ontology learning [Navigli et al., 2011, Velardi et al., 2013], hypernym detection [Flati et al., 2014], or paraphrase

detection [Yan et al., 2013]. In addition, lexicographic information (the kind of information one would expect to find in a dictionary, with definitions being the most prototypical case) is key in many semi-structured resources (e.g. Wikipedia) which are extensively exploited for KB creation, enrichment and completion. With knowledge-based systems being on the rise, it is likely that automatically creating and extending dictionaries will become a core task in intelligent applications involving natural language.

In this section, we provide the reader with a chronologically ordered survey of the most prominent methods for identifying definitions in text collections. For each relevant publication, we highlight key elements such as whether and which linguistic cues are used, whether languages other than English were covered, the corpora used, and method. For thematic clarity, we group all contributions in either **Rule-Based Approaches** or **Machine Learning Approaches**. This overview will serve as a contextualization for better understanding where our contributions to DE fit in.

2.1.1 Rule-Based approaches

Exploiting linguistic regularities in how terms are defined in naturally occurring text is a well-studied topic. In fact, these regularities have been used in more sophisticated machine learning approaches, where presence or absence of certain cues is used as indicative features in statistical models. However, before ML methods became a standard, most work reported results on pattern matching, exploiting linguistic idiosyncrasies in definitions.

An early example of rule-based methods is [Rebeyrolle and Tanguy, 2000], where a set of linguistic patterns are crafted for the French language in order to extract definitions, or *énoncés définitoires*. These patterns were grouped according to the type of definition the system is aiming to extract, e.g. designation (*NP désigner NP*) or meaning definitions (*NP signifier NP*). In their work, the authors report results based on Precision, Recall and F-score for different patterns, differentiating between verb-based patterns and NP-based patterns like "NP such as NP". Similarly, [Klavans and Muresan, 2001] also leveraged indicative cue phrases. However, these were combined with structural indicators with the purpose of automatically constructing a glossary in the medical domain, taking as input a medical corpus, where linguistic variability is scarce.

A further extension of the above appeared in [Malaisé et al., 2004], where the goal was twofold: first, identifying definitions in text corpora; and then, exploiting predefined definition typologies to detect *semantic relations* (e.g. hypernymy and synonymy) holding among terms included in a definition (e.g. the *definiendum* and the *genus*).

Following the idea of leveraging linguistic cues for DE, this is exploited in the

work by [Saggion and Gaizauskas, 2004a]. Their proposed system starts with a set of seed patterns, which are specific to "What is X" or "Who is X" questions, and mines sources like WordNet, Britannica online[1] or the web. A methodologically similar approach is described in [Sarmento et al., 2006], who propose a DE module within a knowledge engineering system (called *Corpógrafo*) which relies on 135 definitional patterns such as "`term` (is OR are) * that *". A remarkable outcome of this work is that, in their evaluation study, they collected data from the user base of *Corpógrafo* and concluded that this approach to DE could speed up the task of finding correct definitions (even with a human post-editing step) by several orders of magnitude. It contains patterns for Portuguese, English, Spanish, Italian and French.

Continuing with work in languages other than English, [Storrer and Wellinghoff, 2006] proposed to detect definitions in the German language by exploiting "frames" for 19 definitors, i.e. verbs that occur in definitional sentences. In their study, they identify a recurrent problem in rule-based DE, which is that of overly generic patterns having very high recall but low precision, due to the fact that they capture many sentences that match all linguistic requirements but are, in fact, not definitions.

Despite these concerns, rule-based approaches continued to be highly utilized in subsequent years. This is the case in another set of works for Spanish DE [Sierra et al., 2006b, Sierra et al., 2006c, Sierra, 2009, Alarcón, 2009]. The general vision of these contributions is to develop a rule-based system able to capture definitional knowledge (also known as *definitional context* or *contexto definitorio*, in Spanish), with the added value of discriminating among different *definition types*. For example, by considering cases where a term is defined by giving examples, by simply enumerating its components, where the genus is missing in the definiens, or whether the definition complies with the canonical *genus et differentia* structure.

The clearest advantages of all these rule-based approaches are based on the fact that, technically, it would be possible to craft extremely specific linguistic pattern matching rules, particularly tailored to a given domain, register and langauge. These would probably achieve high precision, at the expense, however, of low recall. To surmount this problem, machine learning methods have become the standard in DE. The latter are useful for tackling some of the main issues arising from rule-based methods, which according to [Del Gaudio et al., 2013], include language dependence and domain specificity.

[1] `http://www.britannica.com`

2.1.2 Machine Learning Approaches

In what follows, we review previous contributions to DE approached via ML techniques. These include either supervised binary classification (the most common), or other approaches like semi supervised methods, or sequence-to-sequence learning.

2.1.2.1 Supervised

The earliest work we refer to starts with standard pre-processing techniques (tokenization, part-of-speech taggin or partial parsing), and after applying lexical and punctuation indicators, extracts metalinguistic fragments (definitions) and introduces a classification module based on machine learning techniques [Rodríguez, 2004].

Next, [Cui et al., 2005] propose a *soft pattern matching* algorithm. This matching is performed on lexico-syntactic structures, and used to model textual data either as bigrams or as Profile Hidden Markov Models (PHMM), allowing in both cases approaching the task as a probabilistic process. The main intuition behind the inclusion of a more flexible probabilistic model like PHMM is to account for slight variations in semi-fixed definitional cues. For instance, for the pattern "`term`, which is known for ...", a bigram model would fail to cover cases like "`term`, which is best known for ...", or "`term`, whose xxx is known for ...".

In ML approaches there is also work in languages other than English. For instance, the work by [Fahmi and Bouma, 2006] describes a system that operates on fully parsed Dutch text from Wikipedia. Pattern annotation consisted in definitional cue phrases such as *NP* + *Copula* + *V*, and a sentence level manual classification considered three classes: "definition", "non-definition", and "undecided". The process of training a model able to distinguish between definitional and non definitional sentences is based on identifying features that account for properties such as: (1) text properties, combining bag of words and bigrams with punctuation, a feature also used in [Muresan and Klavans, 2002]; (2) document properties, such as the position of a sentence in the document, the intuition being that definitions may occur more frequently at the beginning of a document; (3) syntactic properties, namely position of each subject and its complement; and (4) named entity tags, using the regular typology of "location", "person" and "organization".

Following this line of combining recurrent linguistic patterns with ML, [Westerhout and Monachesi, 2007a] describe a system where a grammar that matches the syntactic structure of a definition sentence is applied in a first run in order to obtain a set of candidates. Then, a machine learning component is incorporated. It uses manually annotated data for training, and models each sentence taking ad-

vantage of similar properties as [Fahmi and Bouma, 2006]. Another contribution with similarities from the methodological standpoint was developed for the Polish language, and showed that combining a simple grammar with a ML classification step (using a bag of *ngrams* as the only feature set) yielded better results than applying highly sophisticated linguistic grammar alone, or ML alone [Degórski et al., 2008]. All of the above contributions are among the many outcomes resulting from the LT4eL[2] project, which aimed at providing easier access to e-learning technologies in several languages. In the DE area, notable work includes, *inter alia*, [Degórski et al., 2008, Del Gaudio and Branco, 2007, Borg and Rosner, 2007].

Among all the contributions framed within the LT4eL project, let us discuss with deeper detail the innovative work by [Borg et al., 2009]. Their proposed method combines genetic programming for learning typical linguistic forms in definitional sentences in corpora, together with genetic algorithms for learning the weights for these forms. Still, while the algorithmic choice is novel in comparison with previous work, which used more traditional Naïve Bayes, SVM, or Random Forests classifiers, the conceptual methodological approach based on revealing linguistic regularities in definitions as opposed to distractors still remained the core task. As such, the feature vector learned for each sentence contains information on, for instance, whether the sentence contains the verb "to be" or the "is a" sequence, whether there exists a foreign word in the sentence followed by "is", if there are possessive pronouns, or paralinguistic features (italics, bold, etc.)

Let us refer now to one of the most impactful systems (and its associated dataset) of our survey, namely the WCL (Word-Class Lattices) system [Navigli and Velardi, 2010]. The main idea is to model each sentence as directed acyclic graphs, with the purpose of retaining salient differences among different sequences, while at the same time eliminating redundant information. The authors suggest that, in DE, "the variability of patterns is higher than for *traditional* applications of lattices, such as translation and speech, however not as high as in unconstrained sentences", which makes modeling linguistic regularities in this task viable. The approach consists in extracting WCLs from a training corpus, and for classification, several configurations are tested, e.g. classifying as definition a candidate sentence where a WCL matches, or selecting the combination of lattices that best fit the sentence. The dataset used, which has become a *de-facto* standard for DE, is described in [Navigli et al., 2010]. It consists of 1908 definitions (first sentences from Wikipedia articles), along with 2711 distractors, or as the authors call them, "syntactially plausible false definitions", in order to account for the fact that there may exist sentences that *look like definitions* in terms of their linguistic structure, but are in fact non definitions.

[2] `http://www.lt4el.eu/`

An improvement over the results published in [Navigli and Velardi, 2010] in the same dataset were reported by [Boella et al., 2014]. This is another example in which, in a supervised classification setting, linguistic patterns are used as features to inform a classifier for DE. However, the novely lies in that, parting ways from shallower syntactic approaches (e.g. the grammars in the LT4eL papers, or the WCL approach), the authors propose to exploit *syntactic dependencies*, a linguistic paradigm in which there exist head-dependent relations between words in a sentence which define the role of each word towards its head, and where unlike in constituency grammar, these relations are not phrasal [Nivre, 2005]. Specifically, this system is an SVM-based classifier which takes as input features related to head-dependent relations over the nouns in the sentence, under the intuition that syntactic dependencies between nouns may reveal hypernymic (is-a) relations, and classifies a sentence as being definitional or not.

In [Del Gaudio et al., 2013], a pervasive problem in DE, which is only marginally addressed in previous publications, is brought up. Due to the nature of definitions and their low frequency in free text, all training and evaluation datasets suffer from the *imbalanced dataset issue*: The *definition* class being much more sparsely distributed than the *non-definition* class. This contribution describes experiments in a fully supervised ML setting, along with an extensive discussion on the effect of different resampling techniques (via either oversampling training definitions, or undersampling negative instances). Moreover, another alternative discussed is the possibility to adjusting the costs of misclassifying one or the other class, giving the classifier a higher penalty if, during training, misclassifies a definition (or, more generally, an instance belonging to the minority class) instead of a non definition sentence. The authors present results in the context of DE, where data from the LT4eL project is resampled using various techniques, and conclude that Naïve Bayes outperforms other more sophisticated algorithms in most experiments.

So far, all the systems we have reviewed defined the DE task as a binary classification problem. However, this has inherent design flaws, which may be aggravated depending on the intended application. For example, assuming an automatic glossary generation scenario, only extracting definitions from corpora would not suffice. A posterior step would have to detect definiendum and definiens, and this may be another substantial research problem, as definienda may appear in arbitrary positions in a sentence. An approach that, by nature, would be able to cope with these limitations is described in [Jin et al., 2013]. Here, DE is modeled as a *sequentail classification* problem, where instead of discretely tagging sentences as definitions or non-definitions, the problem is reformulated in a similar way as POS tagging or chunking. The authors present a system based on Conditional Random Fields (CRF) [Lafferty et al., 2001]. It learns the probability of sequentially tagging a given sequence (i.e. a list of words) using a set of predefined labels. These labels are the classic BIO tagset (Beginning, Inside or Outside), and

refer to whether a word is at the beginning, included in, or not in a definition. At training time, the individual probability of an element to have a certain label is modeled, along with *transition probabilities*, i.e. how likely it is for the next word to have the same or a different label as the current one. In this paper, evaluation is carried out over a manually annotated subset of the ACL anthology [Bird et al., 2008] corpus[3].

Having reviewed the most prominent work in supervised DE, in the following we provide the reader with a survey with the (fewer) works that attempted the automatic extraction of definitional knowledge from text corpora in an unsupervised or semi-supervised manner.

2.1.2.2 Unsupervised

In [Reiplinger et al., 2012], a comparison is provided between bootstrapping versus leveraging linguistic analysis (in other words, exploiting linguistic regularities). Building up on the seminal paper by [Yarowsky, 1995], which introduced an unsupervised bootstrapping algorithm for WSD, subsequent bootstrapping methods are based on the same core fundamentals: Bootstrapping algorithms are semi or unsupervised classification methods which rely only on a few seeds, i.e. correctly labeled instances, for training a first version of the model. A self-training algorithm *bootstraps* a development set extracting confidently classified instances, and transfers these instances to the original training data. The training seeds are expected to be of very good quality, as they constitute the core of the learning algorithm, and the training data gradually increases in size as the bootstrapping process advances. In the specific case of this contribution, the authors, after preprocessing, obtain *seed* terms and definition patterns. After matching definition patterns with one of the predefined terms, term-term pairs are ranked with the Pointwise Mutual Information metric, and used iteratively to discover novel patterns, and so on. This approach, much like in [Jin et al., 2013], is run and evaluated on the ACL Anthology. In this specific case, however, the quality of extracted definitions is rated by domain experts and thus direct comparison with [Jin et al., 2013] is not possible.

The last publication we review is *GlossBoot* [Faralli and Navigli, 2013], a minimally supervised approach for the acquisition of multilingual domain glossaries from the web. The system requires a potentially small set of term-hypernym pairs and starts by collecting web pages with these terms, from which snippets starting from the term and ending in the hypernym are extracted, broken down in term and gloss, and added to a set of candidates. Then, glosses are ranked and filtered by

[3]Code and datasets are available for download at: `https://github.com/YipingNUS/DefMiner`.

domain pertinence and, finally, seeds are selected for the next iteration. This last module is based on a re-ranking of newly acquired term-hypernym pairs.

2.1.3 Conclusion

It clearly stems from the works reviewed in this section that the DE task has traditionally benefited from two main sources of information. On the one hand, linguistic information such as part of speech or syntax has proven to play a crucial role, mostly due to certain regularities exhibited by definitional sentences. On the other hand, annotated corpora (if available), can also contribute to improvement in performance of DE systems, as statistical models are capable to identify definitions which not necessarily comply with classic predefined definition patterns such as the genus et differentia model. However, three main criticisms can be made to the above contributions. First, syntactic information only exploits a kind of shallow parsing in the case of [Navigli and Velardi, 2010], or head-modifier relations between pairs of words in a dependency syntactic tree [Boella et al., 2014]. This leaves much room for improvement for encoding deeper syntactic information in supervised models. Second, no semantic information is considered in any of the above cases, which leaves out a highly informative feature in that definiendum and definiens usually are semantically similar[4] (e.g. the pair (*mosque*, *building*) in a definition such as "a mosque is a building where muslims go to pray"). And third, while a small number of semi-supervised approaches have been described, none of them attempted *domain adaptation*, i.e. to design a DE model able to cope with linguistic idiosyncrasies of a specific target domain, which arguably, in a real world scenario would be highly desirable, as one of the most clear applications of DE systems is the automatic creation of domain glossaries. These issues are specificaly addressed throughout our contributions to DE in Chapter 3.

After having reviewed rule-based and ML learning approaches to DE, we move to another closely related task in which this dissertation presents novel contributions. This task is Hypernym Discovery. We will provide an overview of both pattern-based and distributional methods, again aiming at providing the context of the state of the art in this area, so that our contributions can be put in perspective.

2.2 Hypernym Discovery

Hypernymy, i.e. the capability for generalization, lies at the core of human cognition. Unsurprisingly, identifying hypernymic relations has been pursued in NLP for approximately the last two decades [Shwartz et al., 2016], as successfully identifying this lexical relation not only improves the quality of Question Answering,

[4]This can be measured by looking at how often they occur with similar contexts in corpora.

Textual Entailment or Semantic Search systems [Roller and Erk, 2016], but also is the backbone of almost any taxonomy, ontology and semantic network [Yu et al., 2015].

The terminology is unclear in terms of how this task is to be named, especially since it seems to have branched out towards different but strongly related subtasks. For example, in [Snow et al., 2004] the task receives the name of corpus-based *hypernym pair identification*. Additionally, in [Navigli and Velardi, 2010, Flati et al., 2014], the task is called *hypernym extraction*, since the focus is the *extraction* of a hypernym for a given term in an already extracted snippet of text where both co-occur (a definition). Moreover, in e.g. [Fu et al., 2014], they name the task *automatic discovery of hypernym-hyponym relations*, framed within the broader problem of constructing a lexical taxonomy. Alternatively, [Santus et al., 2014] narrow down the task in (1) *directionality identification*, i.e. detecting the broader term in a given hyponym-hypernym pair; and (2) *hypernym detection*, where the task consists in, given a word pair among which there exists a semantic relation, identify whether it is hypernymy or not. The term *hypernym detection* is also used in [Shwartz et al., 2016], where the task also consists in deciding, for a given word pair, whether there is a hypernymic relation holding between them, or not. This discussion on terminology is important because in one of our key contributions, we address the task of generalization as a more difficult *hypernym discovery* task than the above methods. The task, specifically, consists in, given an input word, finding its most likely hypernym in a large vocabulary, instead of predicting the label of a relation existing in a predefined pair of concepts. For this reason, and despite terminological nuances, we will henceforth refer to any task concerned with the generalization of a concept as hypernym discovery (HD), although in most previous work the task is usually reduced to sequence labeling or even binary classification.

There is a fair agreement in that most contributions to the HD task have either relied on pattern-based methods or distributional approaches, with recent notable exceptions reporting very promising results by combining both into a neural model [Shwartz et al., 2016]. In what follows, we review, first, contributions in the **pattern-based** line of work, and second, **distributional** approaches. Let us mention that, although theoretically the seminal work by Hearst [Hearst, 1992] should fall into the pattern-based category, recent work has showed that these patterns actually emerge in distributional methods as well, when inspecting common contexts in vector space models [Roller and Erk, 2016].

2.2.1 Pattern-based approaches

The first pattern-based method we review is described in [Snow et al., 2004]. Here, the authors first identify sentences in corpora where two terms co-occur

in the WordNet lexical taxonomy. The next step consists in obtaining the dependency parsed representation of these sentences, and automatically extract syntactic patterns from each parse trees. Finally, using patterns based on syntactic dependencies, a hypernym detection classifier is trained.

Next, we refer again to [Navigli and Velardi, 2010], as in their paper they jointly proposed the DE method we reviewed previously, along with an HD module. Without providing any further detail of the WCL method, let us simply highlight that, at training time, those terms corresponding to the hypernym label manually introduced become special nodes which become part of the trained lattice with a hypernym attribute, which later on is used by the classifier to detect the hypernym in an unseen sentence.

Similarly (joint DE and HD), in [Boella et al., 2014], they exploit a system for DE for also performing experiments in the extraction of hypernymic relations from definitions. As explained in Section 2.1, they model each sentence as a set of syntactic subtrees covering nodes performing any of a predefined set of definitional functions (e.g. definiendum and definiens syntactic heads), and use these as features to train a classifier. For HD, if the classifier identifies both term and hypernym as key nodes, they are directly connected and the relation is extracted, with the only constraint that both nodes must be connected in the same parse tree. Their method obtains better results than the one in [Navigli and Velardi, 2010] in the WCL hypernym-extraction portion of the WCL dataset.

Finally, in [Seitner et al., 2016], where the focus is to gather millions of hypernymic relations from a the CommonCrawl[5] web corpus, there is strong reliance on lexical but also shallow syntactic patterns. The authors combine Hearst Patterns with others coming from different sources, constructing a set of 44 patterns. Some of these are "NP_t kinds of NP_h", "compare NP_t with NP_h", or "NP_t forms of NP_h". This contribution comes alongside an online web application, where a user can input terms and categories, in addition with pre and post modifiers[6].

All these approaches, while successful, have major disadvantages when compared with distributional methods. This stems from a core limitation: they require both candidate hyponym and hypernym to occur *nearby* in text corpora, in order for a predefined lexico-syntactic pattern to be able to capture any kind of semantic relation between them [Shwartz et al., 2016]. In what follows we cover distributinonal approaches to HD, which are inherently capable to infer a hypernymic relation between two concepts that were never seen together or nearby in text.

[5] www.commoncrawl.org

[6] http://webisadb.webdatacommons.org/webisadb/

2.2.2 Distributional approaches

Distributional approaches to semantics may be defined as unsupervised methods to build lexical semantic representations from corpus-derived co-occurrences encoded as distributional vectors [Santus et al., 2014]. Distributional approaches stem, in general, from the Distributional Inclusion Hypothesis (DIH) [Zhitomirsky-Geffet and Dagan, 2009], which states that more specific terms appear in a subset of the distributional contexts in which more general terms appear. For instance, the word *animal* may share all its contexts with *dog*, and may occur in additional contexts in which dog will be unlikely to appear.

As for specific contributions, let us start with [Santus et al., 2014], where a novel metric called SLQS is introduced in order to measure the semantic generality of a word by the entropy of its statistically most prominent contexts. This metric is based on the observation that more specific (hyponyms) terms may have linguistic contexts more informative than their corresponding hypernyms (e.g. "bark" or "have fur" for dog, versus "eat" or "run" for dog). It is evaluated on a randomly selected subset of the BLESS dataset [Baroni and Lenci, 2011].

The DIH is further explored in [Roller et al., 2014], who propose a simple supervised distributional model to weight the importance of different context features. Specifically, two classifiers are introduced. The first one is an SVM-based classifier with concatenation of vectors as input features. The second one is a Logistic Regression model trained on difference vectors. The authors mention that, despite usage of difference vectors as features has been reported unsuccessful in previous approaches, they seem to be highly competitive given three modifications, namely: (1) Using a linear classifier, (2) normalizing vectors to magnitude 1, and (3) squared difference vectors should also be included as features. Evaluation is also carried out in the BLESS dataset, as well as in a dataset for evaluating textual entailment systems [Baroni et al., 2012].

Next, in [Fu et al., 2014], the task is to construct a "semantic hierarchy" (a taxonomy) in the Chinese language. We include this work in the hypernym detection section, however, because the authors acknowledge that, while "it is an interesting problem how to construct a globally optimal semantic hierarchy conforming to the form of a DAG" (directed acyclic graph), this was not the ultimate focus of their paper. The main idea is to exploit the semantic properties inherent to word embeddings models [Mikolov et al., 2013c, Mikolov et al., 2013a]. Specifically, the observation that semantically similar embeddings in analogous spaces have a linear relation between them. In [Mikolov et al., 2013b], experiments were conducted in word-level machine translation, and it was shown that "one" (in English), and "uno" (in Spanish) were linearly related. Then, this idea is adapted to the HD task. One of their first findings is that, while linear relations do hold between hyponyms and hypernyms, this is only true if there is some

kind of semantic relation among them. For instance, the *offset* between the vector pair ("carpenter","man") is different than the one for the pair ("dragonfly", "insect"). Their method consists in learning a *linear projection* specific for different semantic clusters, which are obtained by K-Means clustering.

Another relevant contribution in the HD task learns *term embeddings* [Yu et al., 2015]. The main idea is to learn an embeddings space with a distance-margin neural network, learning hypernymic relations identified beforehand. Then, a supervised method based on SVM is applied, using as novel features the concatenation of an input vector pair and their 1-norm distance. Their distance-margin embeddings model leverages two separate spaces, one for hyponyms and one for hypernyms, and the main objective is to minimize the margin between targetted (hyponym, hypernym) pairs, while at the same time maximizing distances with distractors (randomly selected non-hypernymic pairs). Overall, their newly learned embeddings space is expected to have the following three properties: (1) Hyponym-hypernym similarity, e.g. "dog" and "animal" are similar; (2) co-hyponymy similarity, e.g. "dog" and "cat" are similar; and (3) co-hypernymy similarity, e.g. "car" and "auto" are similar. In their experiments, they compare their embedddings space with a vanilla space constructed using *word2vec* default parameters, and with different ways of modeling vectors as features, including the one proposed in [Roller et al., 2014].

Finally, let us review the recent work of [Roller and Erk, 2016], which explores intrinsic properties in the *concat* classifier for HD, described in [Roller et al., 2014]. It consists in using as input features the concatenation of candidate (hyponym, hypernym) pairs. The authors unveil the fact that the model learns to identify this kind of semantic relations by strongly relying on Hearst-like patterns as they appear in the vector space. Their model exploits this observation, in addition to well established observations in the HD task like the DHI and overall word similarity.

2.2.3 Combined approaches

Recently, a novel line of work has opened up [Shwartz et al., 2016], which proposes to combine the advantages found in pattern-based methods (high reliability of hypernymic evidence found in corpora), with those inherent to distributional approaches (less constrained and capable to infer hypernymic relations between pairs of concepts not co-occurring). The proposed method, *HypeNet*, encodes dependency paths (whose contribution to semantics has been extensively discussed in previous chapters) into a Long-Short Term Memory neural network [Hochreiter and Schmidhuber, 1997], a particular type of recurrent neural network architecture well suited for sequence classification. Then, distributional signals are incorporated into the network, showing an overall increase in performance.

2.2.4 Conclusion

In this section, we have provided a survey on the most notable (and thematically relevant to this dissertation) contributions to the HD task. These have been divided into pattern-based, distributional, and combined approaches. One clear criticism that may be derived from the above survey is that in all experiments, the search space is restricted to the evaluation data, which usually consists of hyponym-hypernym pairs mixed up with distractors (e.g. the BLESS dataset [Baroni and Lenci, 2011]). This evaluation setting does not account for the possible scenario in which the task is not to classify a pair of words as having a hypernymic relation, but rather to provide a hypernym out of a very large vocabulary. This and other avenues for improvement are addressed in this thesis, in Chapter 4.

As for downstream tasks, one of the most straightforward application of an HD system is its application to learning lexical taxonomies (hencefort, taxonomies) or ontologies. This is important because taxonomies are of utmost importance for many AI and NLP tasks involving any kind of reasoning, and constitute the cornerstone of the so-called *knowledge-based systems*. Thus, in what follows, we close the literature review chapter by covering prominent work in *taxonomy learning*.

2.3 Taxonomy Learning

A taxonomy is a hierarchy of concepts that expresses parent-child or broader-narrower relationships, and has applications in, among others, search, retrieval, website navigation and records management [Bordea et al., 2015]. Taxonomy Learning is the task of extracting hierarchical relations from text, and subsequently, the construction of a taxonomy. Research in taxonomy learning can be grouped in two main directions, namely *corpus-based taxonomy learning* and *knowledge-based taxonomy learning*. In the former, evidence gathered from large corpora or the web is used to assess the hypernymic relationship between concepts, and usually includes a *graph induction module*, where the final graph (either tree-shaped or as a DAG) is built. Analogously, the latter research area mostly concerns taxonomization of components of KBs, such as Wikipedia pages and categories. In this taxonomy learning literature review, we provide the reader with contributions in both areas.

2.3.1 Corpus-based Taxonomy Learning

In this section, we review prominent research work on corpus-based taxonomy learning, as well as providing a review of two shared tasks which ran in 2015

and 2016. Their impact was threefold. First, they constituted a valuable testbed where distributional, pattern based and machine learning approaches competed in creating high quality lexical taxonomies. Second, multilinguality was specifically addressed in [Bordea et al., 2016], as evaluation considered languages other than English. Finally, the evaluation results of these tasks also unveiled issues related to the controversial topic of taxonomy evaluation, which we further discuss in Section 7.6.

In one of the earliest works in taxonomy learning, by [Snow et al., 2006], taxonomies are incrementally constructed via a probabilistic approach. Evidence from multiple classifiers over heterogeneous relationships is incorporated to optimize the entire structure of the taxonomy. Experimentally, this approach is not strictly speaking concerned with learning a new taxonomy from scratch, but rather with extending WordNet by attaching new concepts to it.

Further advances in taxonomy learning were later proposed in [Yang and Callan, 2009], who introduce a semi-supervised taxonomy induction system that factors in elements such as contextual and co-occurrence evidence, lexico-syntactic patterns, or syntactic dependencies. An ontology metric is learned in order to estimate the *semantic distance* existing between a pair of terms in a taxonomy. This work hols certain similarities with [Snow et al., 2006], as it assumes that the terminology is known, and hence the task is narrowed down to discovering relations between pairs in this terminology. Evaluation is carried out in WordNet sub-hierarchies, such as `People`, `Building`, `Place` or `Meal`.

Lexical patterns are further utilized in [Kozareva and Hovy, 2010]. Starting from an initial seed set of root concepts, basic level terms, and Hearst patterns, they mine the web with a *doubly-anchored* method (combining a general as well as a domain-descriptive term in the query in order to implicitly disambiguate both query terms dynamically). Moreover, their system includes several modules for graph induction, such as removing nodes with low out-degree, or pruning cyclic edges. Additionally, if several paths are available between two concepts in a taxonomy, they keep the longest one, which is standard practice as this is in general preferred in taxonomy learning [Navigli et al., 2011]. Finally, this paper is accompanied with three datasets extracted from WordNet, which have been extensively used for evaluation in subsequent publications. These are sub-hierarchies in the Animals, Plants and Vehicles domains (henceforth, the APV dataset).

Continuing with the trend of developing taxonomy learning systems with little or no supervision, in [Fountain and Lapata, 2012] the idea is to learn a taxonomy approaching the task as inferring a hierarchy from a network or graph. This graph is initially constructed via clustering, which is initiated simply by encoding hypernymic relations between semantically similar terms. Interestingly, this work explicitly acknowledges that evaluating taxonomies is notoriously hard, and hence there may occur that one single domain of knowledge may be equally well

represented by different variations of the same taxonomy. Therefore, this work does not seek to find a single correct taxonomical representation of a domain, but rather valid approximations. Several experiments are conducted, comparing different configurations of the same systems over a semi-automatically constructed validation dataset adapted from [McRae et al., 2005]. Their approach is also evaluated in terms of how well it constructs a taxonomy from large corpora (e.g. the BNC), or how well automatic taxonomies would compare with a manually constructed ones.

Another relevant unsupervised system is ONTOLEARN RELOADED [Velardi et al., 2013][7], a graph-based algorithm which constructs a taxonomy from scratch. The first modules perform DE from corpora, using the WCL algorithm [Navigli and Velardi, 2010]. Then, first, noisy definitions are pruned out using a domain-pertinence statistical filter (called *domain weight*), and then, a dense hypernym graph is constructed by including terms and hypernyms surviving this filtering stage. The pruning step (removing redundant or cyclic edges) leverages various observations, e.g. a node's weight, an edge's weight, or an ideal *optimal* branching of the taxonomy (i.e. moving from a dense graph to a tree-like structure). It is evaluated on the APV dataset, along with manual evaluation of newly constructed taxonomies in the AI domain.

Corpus-based hypernymic evidence is also leveraged as the backbone of full-fledged taxonomic graphs in [Bansal et al., 2014]. In their work, they combine statistical and pattern-based information, along with *heterogeneous relational evidence* of synonymy and siblinghood. Hypernymic relations are obtained with Hearst-like patterns, while at the same time they factor in cues such as coordination (which may indicate that two concepts are siblings). Evaluation is carried out by comparing different configurations of their system, in addition to comparison with previous approaches in a subset of the APV dataset.

Statistical and pattern-based (linguistic) information are further extended with additional taxonomic evidence in [Luu Anh et al., 2014], as the core of their system leverages syntactic contexts of identified taxonomic relations. This essentially introduces syntactic information to the DIH in the form of two measures, namely *web-based evidence*, and *contextual set inclusion*. The graph-construction step, in addition, exploits evidence scores method, as well as topological properties of the graph. Evaluation is also performed on the AVP dataset, in addition to expert judgement on the quality of automatically generated taxonomies. Novel comparative evaluation is introduced versus the automatically generated taxonomy in the AI domain by [Velardi et al., 2013].

The next contribution, proposed by [Luu Anh et al., 2015], takes [Luu Anh et al., 2014] as a baseline, which is improved by incorporating "trustiness and col-

[7] ontolearn.org

lective synonym/contrastive evidence" to the taxonomy learning task. The main idea is the following: taxonomy learning systems, which are in general strongly reliant of obtaining taxonomic relations from the web, may benefit from knowing how trustworthy the textual source from which a hypernymic candidate relation was obtained. In other words, if the web page from which textual evidence was obtained complies with certain quality standards, the likelihood of this evidence of being correct increases. The authors also incorporate synonymic reinforcement and *contrastive* weakening of evidence, essentially assuming that if two synonymous terms occur in the same taxonomic relation, this is a strong indicator that the relation is valid. Their evaluation, also on the AVP dataset, shows a consistent improvement over several baselines, and includes experiments in other datasets released by Velardi et al., namely Virus, Finance and AI.

Bringing back clustering based approaches (as in [Fountain and Lapata, 2012]), in [Alfarone and Davis, 2015] a system called TAXIFY is introduced. Their method consists in a clustering-based inference strategy for improving a taxonomy's coverage. Then, a novel graph-based algorithm is used to prune out incorrect edges in the taxonomy. In this paper, the authors challenge the established intuition that those edges covered by multiple paths are more likely to be correct, by showing empirically that a taxonomy's precision may increase if very popular edges in a taxonomy are deleted.

Finally, in [Luu Anh et al., 2016], the idea is to learn term embeddings (similarly as in [Yu et al., 2015]) via a dynamic weighting neural network. Then, these embeddings are used in a supervised setting for identifying taxonomical relations. Specifically, the concatenation of candidate hyponym-hypernym pair vectors is used as features for an SVM classifier. They complete the feature set by introducing an additional feature, namely the offset vector (subtraction) that contains the information of all the contextual words shared in the candidate term pair. The authors not only evaluate on the AVP taxonomies, but also on the BLESS dataset as well as the ENTAILMENT corpus described in [Baroni et al., 2012].

2.3.1.1 SemEval Taxonomy Learning Tasks: 2015-16

In 2015 and 2016, the interest in taxonomy learning derived in two tasks on lexical taxonomy learning, TexEval [Bordea et al., 2015] and TexEval-2 [Bordea et al., 2016], and one on semantic taxonomy enrichment [Jurgens and Pilehvar, 2016]. We proceed to describe these tasks and to briefly describe the best performing systems.

In both TexEval tasks, systems were asked to perform a hierarchical organization of a domain terminology. In [Bordea et al., 2015], domains were `food`, `equipment`, `science` and `chemical`, and all terminologies were provided in the English language. Evaluation was conducted in terms of precision, recall and

f-score at the edge level, as well as a manual evaluation of a random sample of 100 novel edges (in those cases where the submitted systems included novel nodes or relations). A third evaluation criterion in these tasks concerned structural characteristics of the graph itself, e.g. whether there were cycles, its average depth or the number of connected components it included. The best system for this task was [Grefenstette, 2015], which interestingly was entirely unsupervised, i.e. it did not take advantage of any of the training data that was released nor any pre existing taxonomical resource. The method was based on co-occurence statistics together with substring inclusion counts. In one of our contributions, described in Chapter 5, we report an improvement in average performance with respect to this system.

In [Bordea et al., 2016], the best submitted run, called TAXI [Panchenko et al., 2016] performed substring-based hypernymic relation extraction, which was supported with large domain-specific corpora evidence bootstrapped from the input terminology. Evaluation was similar as in the previous task, but the domains considered were different, i.e. the domains were `environment`, `food` and `science`, and a multilingual challenge was included, by incorporating terminologies in the Dutch, French and Italian languages.

Finally, in [Jurgens and Pilehvar, 2016], the task was to find the *best point of attachment* for a novel lemma in the WordNet taxonomy. Terms came from highly specific glossaries or terminologies, and were accompanied by a descriptive definition and the source `url`. For each novel term, a system had to decide whether the novel term had to be inserted as a hyponym of an existing WordNet synset (if there was no existing synonym in the taxonomy), or as a synonym of an already existing term. Evaluation was performed both in terms of lemma match and via the Wu&Palmer similarity metric [Wu and Palmer, 1994] between predicted WordNet synset and its gold standard. In this task, the best performing system [Schlichtkrull and Alonso, 2016] disambiguated the words in each gloss, and then used a supervised SVM classifier for predicting the goodness of fit for a candidate attachment synset.

2.3.2 Knowledge-based Taxonomy Learning

Unlike corpus-based taxonomy learning systems, in this section we review algorithms (and their associated outcome) which do not attempt to model a set of concepts at the lexical level, but rather are aimed at providing a better (or new) taxonomization of existing semantic resources. The clearest example can be drawn from the case of Wikipedia categories, which provide a certain taxonomization of the pages they subsume, but usually only consist of lists of anchors to other articles, which can be however useful for "capturing categorical information that roughly contains a mixture of hyponymy and meronymy relations between articles" [Yeh et al., 2009]. In the following listing we describe several knowledge

based taxonomies.

1. **WikiTaxonomy**: WikiTaxonomy [Ponzetto and Strube, 2008][8] constitutes the first approach that attempted to taxonomize Wikipedia categories [Flati et al., 2014]. It leverages lightweight yet effective heuristics to define whether hypernymic relations hold between a category and its subcategories, generating about 100k category-wise is-a relations.

2. **WikiNet**: Wikinet [Nastase et al., 2010] is also based on heuristically leveraging different components of Wikipedia, i.e. not only categories but also Wikipedia pages, in order to derive a semantic hierarchy including relations beyond hyepernymy.

3. **MENTA**: In MENTA [de Melo and Weikum, 2010], a multilingual lexical KB is created by linking together more than 10M articles in 271 languages. An interesting feature is how the mapping between Wikipedia pages and WordNet synsets is obtained. Specifically, they used a supervised ridge regression system trained with manually labeled examples. It uses as features signals such as term overlap, semantic similarity via cosine distance, and presence or absence of what they call *qualifications*, i.e. the "explanation" in brackets in ambiguous Wikipedia pages, e.g. *(novel)* in the page *House (novel)*.

4. **WiBi**: WiBi (which stands for Wikipedia Bitaxonomy[9]) [Flati et al., 2014] is an approach to reconcile in one resource an automatically learned taxonomic structure of Wikipedia pages and categories. It performs several NLP-intensive steps, e.g. hypernym extraction, disambiguation and linking. Roughly, WiBi first performs a Wikipedia page taxonomization followed by its integration into a page and category taxonomy. According to their experiments, WiBi results in a wide-coverage and self-contained resource, rivaling the granularity and average depth of WordNet, the reference lexical resource.

2.3.3 Conclusion

We have provided the reader with a review on both corpus-based and knowledge-based taxonomy learning systems and their associated assets. Both research directions seem to be fairly disconnected with one another in terms of methodological design and evaluation (although interestingly, in both areas evaluation almost always concerns WordNet in one way or another). This disconnection provokes, for

[8] www.h-its.org/en/research/nlp/wikitaxonomy

[9] wibitaxonomy.org/

instance, that semantically aware systems like the ones described in this section are by design unable to capture novel terminology (as all nodes in the taxonomy must exist a priori in the KB being taxonomized). The implicit overlap between these two research areas is explicitly exploited in our novel system EXTASEM! (cf. Chapter 5), where we leverage corpus-based evidence and the Wikipedia graph structure to generate domain-specific lexical taxonomies, with many additional terms derived from the extraction and parsing of hypernyms present in definitions.

Chapter 3

DEFINITION EXTRACTION

In this chapter, we describe our methodological contributions along with experimental results in the field of DE. As we discussed in Chapter 2, current approaches look mostly at linguistic and statistical evidence to model the DE problem, without considering explicit semantic representations. In this chapter, we provide experimental results that incorporating syntactic dependencies for modeling candidate definition sentences, on one side, and semantics (e.g. WSD, Entity Linking and distributional semantics), on the other, contributes decisively to the DE task, outperforming competing approaches solely based on shallower linguistic and statistical features.

Our first contribution leverages syntactic regularities, specifically, syntactic dependencies, and uses them to model each individual sentence as a vector over syntactic features such as presence or absence of prototypical definitional patterns, or the syntactic neighbours of certain words acting as subject, predicative complement or direct object. Henceforth, we refer to this method as **DependencyDE**. Second, we evalute the extent to which state-of-the-art WSD and entity linking techniques, on one hand, and sense-based vector representations, on the other, may contribute to the improvement in performance in supervised DE. We will refer to this approach as **SemanticDE**. Next, we describe one additional experiment on supervised DE, this time for a language other than English and with a different methodological approach, which we denote as **SequentialDE**. Finally, we delve into a more unexplored area, which is the one concerning unsupervised DE, and we propose a system called **WeakDE**, with experiments carried out in English, but easily portable to any other language.

Before focusing on the details of each proposed system, let us mention that in both **DependencyDE** and **SemanticDE**, the **dataset** used was the WCL dataset (cf. Section 2.1.2.1). Following the terminology generalization approach from [Navigli and Velardi, 2010], all the definienda are generalized to a wildcard `target` token, which allows its inclusion as a feature for the learning algorithm.

3.1 DependencyDE: Applying Dependency Relations to Definition Extraction

3.1.1 Data Modeling

Our motivation stems from the observation that syntactic dependencies provide a framework for parsing and analyzing deep linguistic structures in natural language. These structures are described by the distribution of lexical elements linked by asymmetrical relations called dependencies. One of its main characteristics is that, unlike constituent structures, a dependency tree has no phrasal nodes. Moreover, dependency representations provide a direct encoding of predicate-argument structures (see the dotted dependencies in Figure 3.1). Finally, the relations between units in a dependency tree are bilexical, which makes them beneficial for disambiguation [Nivre, 2005]. Modeling each sentence as a set of syntactic relations between words, where the most important ones act as *heads* and those adding certain information act as *modifiers*, has proven useful in several applications, e.g. Information Extraction [Sudo et al., 2003] [Stevenson and Greenwood, 2006] [Afzal et al., 2011], paraphrase identification [Szpektor et al., 2004] or KB construction [Delli Bovi et al., 2015].

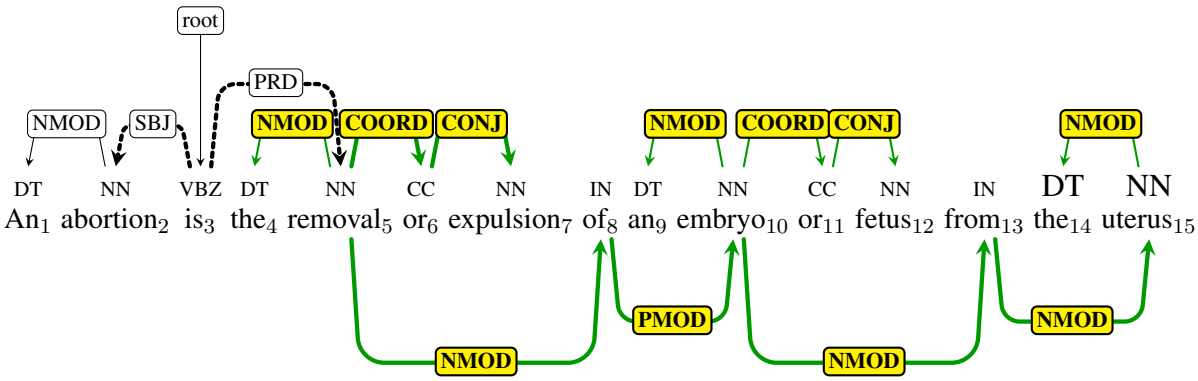


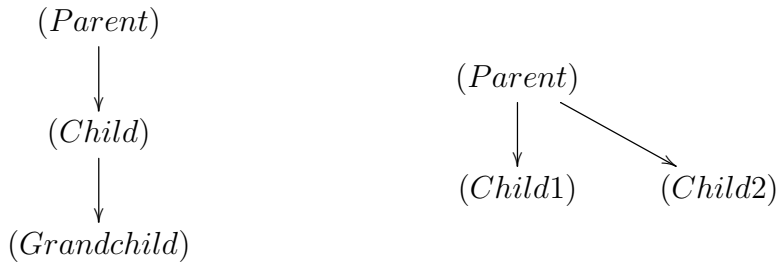
Figure 3.1: Example definition parsed with syntactic dependencies

Specifically, and in the context of DE, the dependency structure of a sentence may reveal syntactic regularities, indicating high likelihood of such sentence being a definition. For example, sentences S1 and S2 below share the same surface structure (`target` *is * which was **). However, only S1 is a definition.

> S1: `target` *is the independent school which was opened.*
> S2: `target` *is secure against CCA2 which was proved to be a security hazard.*

The main difference is that the dependency relation between the verb and the phrase "the independent school" is of **object noun phrase**, while in S2, the relation with the adjective "secure" is of **adjectival phrase**. This is the kind of information that (in S1) an *ngram*-based approach[1] would be unable to tackle due to the non-adjacent distance between the components.

In our approach, we model textual data as a "bag of subtrees", i.e. we extract all parent-child-grandchild (PCG) (left) and parent-child-child (PCC) (right) subtrees from dependency parsed training data, and use them as features for training. In what follows, we distill the intuition of definitional information that may be captured by any of these predefined syntactic structures.



Syntactic dependency relations may reveal the *domain* or *discipline* governing a definition, and can be expressed by a locative at the beginning of a sentence. Consider the following sample sentence: ***In law**, an abstract **is** a brief statement that contains the most important points of a long legal document or of several related legal papers*. The highlighted words `is` $\xrightarrow{root}$ `in` $\xrightarrow{loc}$ `law` form a PCG subtree, where the locative preposition `in` connects the definition's domain or topic with the verb. In the definitional split of the WCL dataset, almost 20% of all the definitions include this syntactic pattern at the beginning of the sentence.

Additionally, while *term*-is-a-*hypernym* patterns constitute potential candidates for any feature space exploiting syntactic dependencies, attempting an ngram-based approach would present drawbacks. The task would be tackled as surface pattern matching, perhaps including the Part-of-Speech of the candidate hypernym (to disregard, for example, noun phrases whose first word is an adjective). However, by looking at the syntactic function of the noun phrases involved (`sbj` for *term*, and `prd` for the *hypernym*), it is possible to filter out some of the noisy candidates that would be retrieved.

We discuss now on the potential of the PCC subtree. We argue that it can be useful for identifying *SVO* relations [Stevenson and Greenwood, 2006], as well as extracting multiword terminology. This can be further illustrated with the following definition:

*An abugida is a **segmental writing system** which is based on conso-*

[1]For instance, using a sliding window of *n* words for matching definitional cue phrases.

nants but in which vowel notation is obligatory.

The highlighted pattern has the following syntactic structure:

$$\texttt{segmental} \xleftarrow{nmod} \texttt{system} \xrightarrow{nmod} \texttt{writing}$$

Since we know that `system` is the predicative complement of the sentence root node (thanks to the syntactic parsing), we are highly confident that it may constitute the hypernym of the term being defined. In addition to these examples, additional informative instances of PCG and PCC can be indicative of definitional knowledge. For example, (1) non-adjacent subject-predicate patterns; (2) description of the definiens' head; and (3) synonymy relation among an enumeration of heads, all acting as potential genus.

3.1.2 Features

After having provided a linguistically motivated description of our data modeling process, we proceed to describe the **features** we designed, as well as **experimental results**. Syntactic information is obtained after running a graph-based parser [Bohnet, 2010] over the corpus.

1. **Subtrees**: From all the available PCC and PCG subtree combinations, we extract the following information for each node: Surface form (sf), Part-of-Speech (pos), and Dependency Relation (dr). Each sentence is transformed into a feature vector of the 15 most frequent subtrees of each type. Features are binary, i.e. 1 for presence of the subtree, and 0 for absence. The six combinations of linguistic information used in this feature set as well as examples are shown in Table 3.1[2]. Additionally, let us highlight that the number of subtrees used as features is not arbitrary. It comes from a manual analysis of the frequency distribution of each type across the dataset. The definitional sentences in the Wikipedia corpus tend to have recurrent syntactic patterns, which produce a long-tailed frequency distribution and thus a remarkable gap between systematic and idiosyncratic features, and the rest. By keeping only the 15 most frequent subtrees we design a balanced feature set across the types while at the same time disregarding the long tail in each type (Figure 3.2).

[2]For a comprehensive list of the tagset, refer to the CoNLL 2000 shared task. `dkpro.github.io/dkpro-core/releases/1.8.0/docs/tagset-reference.html#tagset-en-conll2000-chunk`

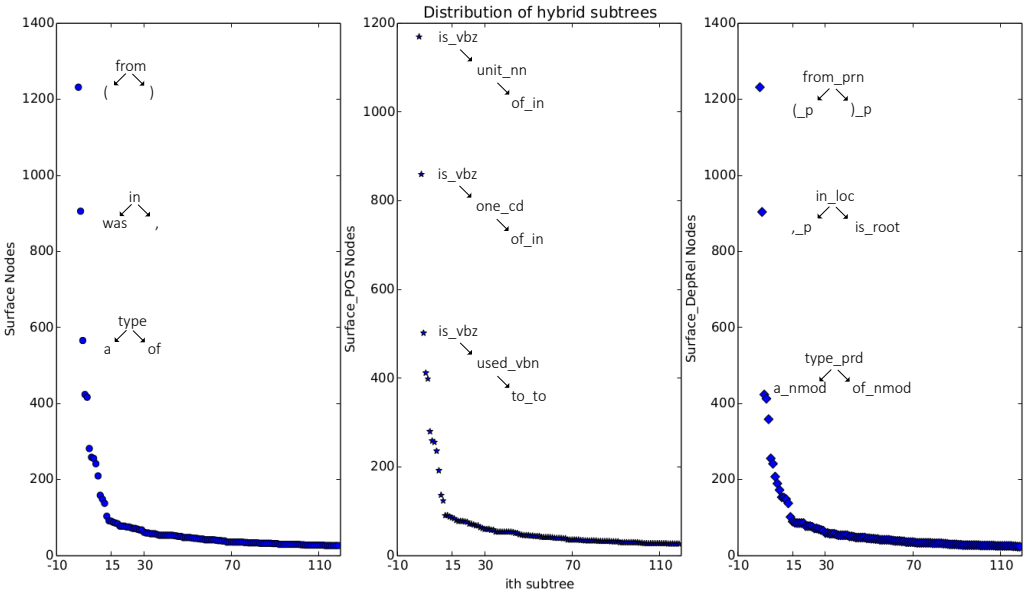


Figure 3.2: Distribution of three selected types of hybrid subtrees illustrating the asymmetry of their frequency distribution. The three most frequent instances of each type are shown in their corresponding figure.

2. **Degree of X**: The *degree* of a node X in a graph is the number of edges adjacent to it, i.e. the sum of of its children + 1 (its head). We reduce the search space of X to

$$\mathrm{X} \in \{\mathrm{PRD}, \mathrm{SBJ}, \mathrm{APPO}\}$$

because in this way, subject nodes with many modifiers are given more importance. For example, in the sample sentence in Figure 3.1, the degree value of the fifth (prd) node is 4.

3. **Morphosyntactic chains starting in node** X: X may have the same node value as in the previous feature, i.e. prd, sbj and appo. If it exists in the sentence, we extract all the children from that node recursively until leaf nodes are reached. We then extract POS and dependency relation chains and order them according to their order in the sentence. This approach for feature extraction has proven useful in other NLP tasks, such as Semantic Role Labeling [Hacioglu, 2004]. For example, in our sample sentence[3] we would extract the following chain from the prd node (in breadth-first fashion): $\{\mathtt{nmod}_4, \mathtt{prd}_5, \mathtt{coord}_6, \mathtt{conj}_7, \mathtt{nmod}_8, \mathtt{pmod}_{10}, \ldots\}$ until the last child is reached (green and yellow colored dependencies in Figure 3.1).

[3]Subinidices indicate the word's position.

type of subtree	example
⟨ sf, sf, sf ⟩	⟨ TARGET, refers, to ⟩ ⟨ is, used, in ⟩
⟨ pos, pos, pos ⟩	⟨ dt, jj, nn ⟩ ⟨ in, nn, vbd ⟩
⟨ dr, dr, dr ⟩	⟨ sbj, root, prd ⟩ ⟨ pmod, coord, conj ⟩
⟨ (sf,pos), (sf,pos), (sf,pos) ⟩	⟨ (is, vbz), (a, dt), (unit, nn) ⟩ ⟨ (a, dt), (form, nn), (of, in) ⟩
⟨ (sf,dr), (sf,dr), (sf,dr) ⟩	⟨ (in, loc), (TARGET, sbj), (was, root) ⟩ ⟨ (is, root), (any, prd), (of, nmod) ⟩
⟨ (pos, dr), (pos, dr), (pos, dr) ⟩	⟨ (nn, pmod), (cc, coord), (nn, conj) ⟩ ⟨ (dt, nmod), (nnp, name), (nnp, pmod) ⟩

Table 3.1: Summary of the types of subtrees used and examples of each type. sf=surface form, pos=part of speech, dr=dependency relation.

4. **Ordered cartesian product of two subtrees**: The ordered cartesian product of two graphs G_1 and G_2 produces a new graph H with the vertex set $V(G_1) \times V(G_2)$, with the tuples $\{(i_1, i_2), (j_1, j_2)\}$ forming an edge if $\{i_1, j_1\}$ forms an edge in G_1 and $i_2 = j_2$, or $\{i_2, j_2\}$ forms an edge in G_2 and $i_2 = j_2$. Our intuition is that by extending the relationships between pairs of specific head nodes and their children, deeper relations between modifiers of `sbj` and `prd` nodes, for example, would be captured and reinforced. We perform this operation only if the head of G_1 has the syntactic function `sbj` and the head of G_2 has the head `prd` or `appo`. The result is a string that contains surface, POS information or dependency information, chained over H. In our working example, the dependency level of this feature would be "(`sbj`+`prd`), (`sbj`+`nmod`), (`sbj`+`coord`), (`nmod`+`prd`), (`nmod`+`nmod`), (`nmod`+`coord`)".

5. **Semantic similarity**: We hypothesize that high semantic similarity between words in candidate definiendum and definiens position, for example, might point towards a definitional sentence. In our sample sentence, this would be the case between *abortion* and *removal*. We extend this feature to other nodes like appositives or their modifiers, and apply it to the following pairs: (`sbj`,`prd`), (`sbj`, `appo`), (`prd`, `jj`+`pmod`) and (`appo`, `jj`+`pmod`). Us-

ing WordNet as our reference semantic inventory, we compute the average similarity between all the synsets associated to a given lemma. In this example, we would evaluate the similarity between (`abortion.n.01`, `abortion.n.02`) and (`removal.n.01`, `removal.n.02`). Similarity is computed using the Leacock Chodorow Similarity [Leacock et al., 1998] measure $LCsim$, denoted as:

$$LCsim(ws_1, ws_2) = -log \text{ pathlen}(ws_1, ws_2)$$

where ws_1, ws_2 are word senses, and pathlen(ws_1, ws_2) is the shortest number of edges between those two word senses in WordNet.

3.1.3 Evaluation

The above features are incorporated into a sentence-level feature vector, and this information is used for training different classification algorithms present in the `Weka` workbench [Witten and Frank, 2005]. The evaluation results we report are based on 10-fold cross-validation on the WCL dataset. Table 3.2 shows the scores for the different setups on which the experiments were carried out. For each algorithm, the different configurations are: S_1, which includes the full feature set, S_2 disregards chain and cartesian product features, and S_3 disregards chain, cartesian product, degree and similarity features. Likewise, we show comparative results with competitor systems in Table 3.3. These systems are:

- **Bigrams**: Baseline 1 based on the bigram classifier for soft pattern matching proposed by [Cui et al., 2005].
- **Star Patterns**: Baseline 2 based on a pattern-matching approach in which infrequent words are replaced by '*' and then matched against candidate definitions [Navigli and Velardi, 2010].
- **WCL 1**: Implementation of the Word-Class Lattice model where a lattice is learned for each cluster of sentences [Navigli and Velardi, 2010].
- **WCL 3**: Implementation of the Word-Class Lattice model where a lattice is learned for each of the components of a definition, namely *Definiendum*, *Definitor* and *Definiens* [Navigli and Velardi, 2010].
- **DependencyDE**: The best configuration of our approach.

In the light of these scores, it seems reasonable to argue that a classification approach for DE can improve substantially by including features that account for the

	NaïveBayes			VPerceptron			SVM			LogisticR			DTrees			RandomF		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
S_1	81.9	78.0	75.9	78.9	85.2	84.4	**85.9**	**85.3**	**85.4**	83.4	82.7	84.7	85.7	85.3	85.2	83.4	82.9	82.2
S_2	75.7	75.9	75.5	80.5	79.4	79.5	82.0	81.2	81.3	82.2	81.5	81.6	80.8	79.4	79.6	79.9	79.4	79.5
S_3	53.1	58.6	49.0	56.9	59.8	52.2	56.9	59.8	52.2	56.9	59.8	52.2	55	59.2	49.3	56.9	59.8	52.2

Table 3.2: Scores of **DependencyDE** obtained using different machine learning algorithms.

morphosyntactic structure of the sentence. Moreover, deep analyses of syntactic trees and the relation among dependents contributes decisively to DE.

The highest scoring approach in terms of Precision is achieved by the WCL systems, with almost 1. However, the highest score in terms of Recall (85.3) and F-Measure (85.4), on the other hand, are achieved by **DependencyDE**. The improvement in 2 points over **WCL 3** shows that our linguistically-motivated features are a better way to model differences between definition and non-definition sentences in the encyclopedic genre than the lattices algorithm proposed in [Navigli and Velardi, 2010].

	DependencyDE	WCL 1	WCL 2	Star Patterns	Bigrams
Precision	85.9	**99.8**	99.8	86.7	66.7
Recall	**85.3**	42.1	60.7	66.1	82.7
F-Measure	**85.4**	59.2	83.5	75.1	73.9

Table 3.3: Comparative table of results between our approach and the reported scores in Navigli and Velardi (2010).

3.1.4 Conclusion

In this section we have summarized a DE system which leverages syntactic dependencies in a novel way. Parting ways from previous approaches, our method specifically encodes semantic relations between two separate branches in the syntactic tree, at any depth level. This allows for deeper understanding of the syntactic structure of the sentence, and thus contributes decisively to outperforming current methods for DE in the WCL corpus. However, we observed that neither in this experiment, nor in our competitors, was semantic information considered. While we have tangentially introduced this idea by incorporating a feature on semantic similarity between term and (likely-to-be) hypernyms, in the next section we specifically address the semantics of candidate definitions by means of their distributed representations.

3.2 SemanticDE: Definition Extraction Using Sense-based Embeddings

We have reviewed our first contribution in the DE field, where we modeled definition sentences as feature vectors containing information derived from their syntactic tree, and used this information for training a set of machine learning classifiers to discriminate between definitions and non-definitions. While our results improved the state-of-the-art at the time of publication, we hypothesize that semantic information, neglected insofar, contribute dramatically to achieving even better results. The main intuition is that a term's definition usually includes concepts which are semantically related, and hence this relatedness may be modeled by exploiting similarities across vector space models. In this section we describe our approach to DE which puts this intuition into practice.

Specifically, we propose to investigate an approach which combines off-the-shelf WSD and Entity Linking with sense-based vector representations as a cornerstone for modeling textual data. Our experimental results confirm, indeed, that semantic information contributes dramatically to extracting definitional knowledge from corpora.

3.2.1 Entity Linking

The first step of our approach consists in running Babelfy [Moro et al., 2014], a state-of-the-art WSD and Entity Linking tool which leverages BabelNet [Navigli and Ponzetto, 2012] as its reference sense inventory, over the WCL dataset. In this way, we obtain disambiguations for content text snippets, which are used to build a semantically rich representation of each sentence. Consider the following definition and its concepts, represented with their corresponding BabelNet synset id:

> The⟨O⟩ Abwehr⟨01158579n⟩ was⟨O⟩ a⟨O⟩ German⟨00103560a⟩ intelligence ⟨00047026n⟩ organization⟨00047026n⟩ from⟨O⟩ 1921⟨O⟩ to⟨O⟩ 1944⟨O⟩.

This disambiguation procedure yields two important pieces of information. On the one hand, the set of concepts, represented as BabelNet synsets, e.g. the synset with id bn:01158579n for the concept Abwehr_{bn}[4]. On the other hand, we also obtain a set of non-disambiguated snippets (either single word or multiword terms), which can be also used as indicators for spotting a definitional text fragment in a corpus (from the above example: {*the*, *was a*, *from 1921 to 1944*}).

[4]For clarity, we use the subscript *bn* to refer to the concept's BabelNet id, rather than using the actual numeric id.

3.2.2 Sense-Based Distributed Representations of Definitions

Our second step relies on SENSEMBED [Iacobacci et al., 2015]. This is a VSM where not words, but rather *senses* are included in the vector space, along with their BabelNet id. SENSEMBED vectors are the result of a two-step approach: First, a large text corpus is disambiguated with Babelfy. Then, *word2vec* [Mikolov et al., 2013c, Mikolov et al., 2013a] is applied to the disambiguated corpus, yielding a vectorial latent representation of word senses. This enables a disambiguated vector representation of concepts. For instance, for the term "New York" (BabelNet id bn:00041611n), there are vectors for lexicalizations such as "NY", "New York", "Big Apple" or even "Fun City". Similarly, one text-level concept may be associated with more than one vector as well, one for each of the BabelNet synsets that include such concept as its lexicalizations.

The representation of a sentence leveraging both Babelfy and SENSEMBED is as follows. We first consider the text-level mentions provided by Babelfy. In other words, we use this tool simply as a NER/phrase chunker in order to obtain input concepts to look up in the sense embeddings model. Then, given a pair of concept (e.g. *intelligence organization*) or entity (e.g. *Abwehr*) mentions (x, y), we compute their semantic similarity SIM$(\cdot)$, which outputs the cosine score of their two closest senses. These pairwise similarity scores are afterwards used for computing a *compactness graph* over a sentence, and this information ultimately becomes the input for our graph-based set of features (denoted as Δ).

We compute SIM$(\cdot)$ as follows. Let S be the set of senses included in SENSEMBED and Γ the set of associated vectors to each sense. We first retrieve all the available senses in S of both x and y, namely $S(x) = \{s_x^1, ..., s_x^m\}$ and $S(y) = \{s_y^1, ..., s_y^z\}$. Then, we retrieve from Γ the corresponding sets of vectors $V(x) = \{v_x^1, ..., v_x^m\}$ and $V(y) = \{v_y^1, ..., v_y^z\}$. Finally, we compare each possible pair of senses and select the one maximizing the cosine similarity between their corresponding vectors, i.e.

$$\text{SIM}\,(x, y) = \max_{v_x \in V(x), v_y \in V(y)} \frac{v_x \cdot v_y}{||v_x|| ||v_y||}$$

This disambiguation strategy at word or phrase level is further leveraged in subsequent experiments throughout this dissertation. Hence, when we refer to this approach (e.g. in Chapters 5 and 6), we denote it as **L2S** (Lemmas to Sense).

Let us illustrate the result of our **L2S** disambiguation strategy with an example. Given the definition of the term *bat*, "A bat is a mammal in the order Chiroptera", we obtain a set D of three concepts: bat$_{bn}$, mammal$_{bn}$ and Chiroptera$_{bn}$. For each pair of concepts $c_i, c_j \in D$, we compute SIM(c_i, c_j), and perform this operation over all pairs in D.

In Table 3.4, we show the SIM representation of this definition (*d*) and one

non-definitional sentence (*n*) also referring to *bat*: "This role explains environmental concerns when a bat is introduced in a new setting". In this distractor sentence, disambiguated concepts are role$_{bn}$, environmental_concern$_{bn}$ and batch_language$_{bn}$. Note the higher SIM scores for concept pairs in the definitional sentence (in bold). Also, note that since the non-definition is less *semantically compact*, our procedure assigned to one single term (e.g. *bat*$_{bn}$) vectors corresponding to different lexicalizations depending on which concept it was being disambiguated against (*bat* is incorrectly disambiguated as the batch programming language and as a batch file). This also affects the connectiveness of the resulting graph, which is more likely to be fully connected when concepts tend to be semantically closer in the space, and hence are less likely to be disambiguated differently (see Figure 3.3 for a visual representation of the resulting graph of a definitional and a non definitional sentence about the term 'bat').

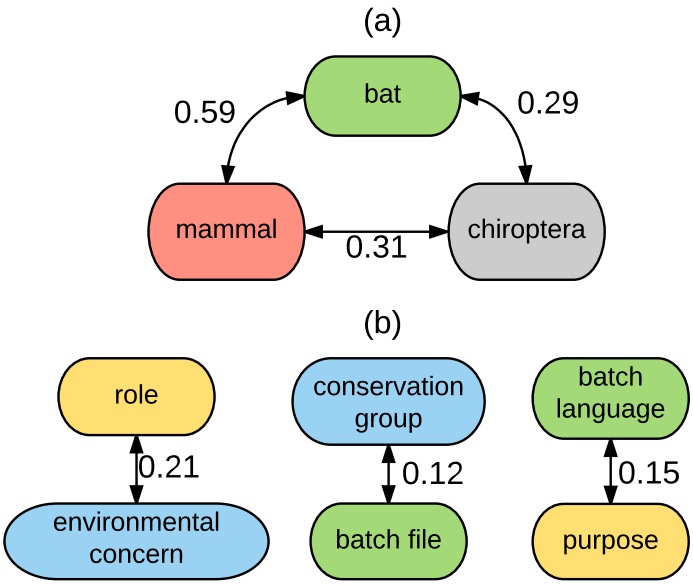


Figure 3.3: Graph representation of two sentences. In the definition (a) the semantic compactness favours a fully connected graph, unlike the case of the non definition (b).

3.2.3 Sense-based Features

We design three types of features: (1) Bag-of-Concepts; (2) Bag-of-non-disambiguated text snippets; and (3) Similarity and compactness metrics over Δ. These features are then used to train different classification algorithms, whose performance is evaluated in 10-fold cross validation.

Vector	Vector'	SIM
bat$_d$	**mammal**$_d$	0.59
bat$_d$	**chiroptera**$_d$	0.29
mammal$_d$	**chiroptera**$_d$	0.31
role$_n$	environmental_concern$_n$	0.21
purpose$_n$	batch_language$_n$	0.15
conservation_group$_n$	batch_file$_n$	0.12

Table 3.4: Representation of a definition and a non-definition in terms of the similarities of its concepts.

- **Bag of disambiguated Concepts**: We extract the 100 most frequent BabelNet synsets in the training data. Each concept behaves as a binary feature, with value being either *True* or *False*, referring to whether such concept was found in the sentence. In most folds, the most frequent synsets refer to ancient languages such as Greek or Latin, or to scientific disciplines such as Maths or Geography, typical indicators of definitions in the encyclopedic genre.

- **Bag of non-disambiguated Concepts**: We extract the 500 most frequent text snippets that Babelfy did not disambiguate. The vector construction procedure is the same as in Bag-of-Concepts. In this case, we obtain results consistent with previous studies in that the pattern "is a" is the most frequent and hence a feature with high predictive power, followed by "is the", "of a" and "is any".

- **Semantic Features**: We put forward a novel set of features stemming from the *semantic compactness* hypothesis, described in Section 3.2.2. We build on this intuition to propose the following features:

  - **AllSims**: The sum of all the SIM scores in Δ.
  - **AvgSims**: The average of the SIM scores in Δ.
  - **AvgBiggestSubGraph**: As shown in Figure 3.3, we can represent a candidate sentence as a non-directed graph, in which each node is a concept and each edge is weighted according to their SIM score. However, there are cases in which not all components of the graph are connected because one mention may be associated to *n* different senses (vectors) depending on which concept it is disambiguated against. This feature is the average of the cosine scores of the biggest con-

nected subgraph generated from Δ[5]. Note that if the sentence graph is complete, **AvgSims** and **AvgBiggestSubGraph** yield the same score.

- **TopDegreeScore**: First, we obtain the node with highest degree in the graph representation described above, i.e. the most connected node. Then, we compute the average SIM score over this node and its neighbours. We hypothesize that this measure should reward concepts whose disambiguation remains the same regardless of the concept they are disambiguated against (meaning less ambiguity), which can be seen as another *semantic compactness* measure.
- **NumEdges**: The number of edges of the graph described above. As the disambiguation options for a given concept increases, so will increase the number of edges of the graph representation.
- **MaxScore and MinScore**: The maximum and minimum SIM score among all the concept pairs in Δ. We hypothesize that in a definitional sentence, there will be at least one pair highly similar, the one between the defined term and the hypernym.

These features are used to perform a set of experiments with the machine learning toolkit WEKA [Witten and Frank, 2005].

3.2.4 Evaluation

Our approach (**SemanticDE**) shows competitive results, outperforming previous systems on the same dataset. We compare against three main competitors: (1) The WCL algorithm (WCL); (2) A supervised machine-learning setting (BdC) in which syntactic dependencies are used to construct word representations in terms of their direct descendants [Boella et al., 2014]; and (3) our approach based on syntactic dependences: **DependencyDE** (Section 3.1).

As is the case in all the systems described, performance is evaluated with the classic Precision, Recall and F-Score measures at sentence-level. Table 3.5 shows the performance of all systems.

In addition, we complement our experiments by evaluating the relevance of each individual feature from our feature set. To this end, we compute their Information Gain score, which measures the decrease in entropy when the feature is given vs. absent [Forman, 2003]. The feature ranking provided in Table 3.6 summarizes the discriminative power of the features derived from SENSEMBED, reinforcing our claim that semantic information can be effectively applied to the DE task.

[5]Graph operations performed in our experiments were done with the Python library NetworkX: https://networkx.github.io/

	Precision	Recall	F-Score
WCL	**98.8**	60.7	75.2
BdC	88.1	76.2	81.6
DependencyDE	85.9	85.3	85.4
SemanticDE	86.1	**86.0**	**86.0**

Table 3.5: SemanticDE results on the WCL dataset

InfGain Score	Feature
“Contains:is_a"	0.19
AvgSims	0.13
AvgBiggestSubGraph	0.12
MaxScore	0.07
MinScore	0.06
TopDegreeScore	0.04
“Contains:is_an"	0.03
“Contains:bn00103785a"	0.02
NumEdges	0.01
AllSims	0.01

Table 3.6: SemanticDE top 10 features, by Information Gain score

3.2.5 Conclusion

As a concluding remark, let us summarize the main contributions of the **SemanticDE** experiments. This supervised approach for DE benefits substantially from introducing quantitative and distributional information derived both from WSD and Entity Linking (thanks to BABELFY), as well as sense-based embeddings (specifically, SENSEMBED vectors). We showed that the *semantic compactness* of a definition is a trait that can be exploited for improving the quality of DE systems.

Inspired by the results obtained with this set of semantic features, in addition to the good performance of our previous contribution (**DependencyDE**), a natural extension of this work would be combining both intuitions into one single model,

which could potentially perform well in out-of-domain settings due to its combination of linguistic regularities and semantic information. This is a challenging and motivating task that remains for future work.

3.3 SequentialDE: Description and Evaluation of a DE system for the Catalan language

Multilingual DE remains a fairly unexplored area. This is usually due to lack of domain-specific data, as well as little work on exporting existing DE systems to other languages. For these reasons, we were compelled to explore the more difficult task of DE for a language other than English, on the same encyclopedic textual genre as our two previous contributions (**DependencyDE** and **SemanticDE**). We decided to opt for the Catalan language due to the fact that it constitutes an example of a fairly under resourced language, for which lexical resources, corpora, as well as validation datasets are scarce. Second, it is one of the languages included in Wikicorpus [Reese et al., 2010], which allows us to concentrate on the experiment itself without having to focus on a possibly noise-inducing step regarding corpus acquisition and preprocessing.

In addition, and from a purely methodological perspective, in this experiment we explore DE as a sequence-to-sequence classification task, rather a binary classification problem. This changes substantially the experimental setup. First, because we no longer model feature vectors over each sentence, but rather over each token. And second, because in this kind of sequence-to-sequence classification problem, the standard practice is to have three labels, namely **B** (beginning), **I** (inside), and **O** (outside) the target class (in our case, definitions), rather than the two labels used in binary classification.

3.3.1 Creating a Catalan corpus for DE

We use a reference corpus as a pivot between English and Catalan Wikipedias, namely the WCL dataset. We produce a Catalan mapping from WCL, which constitutes our *evaluation* set. We map all unambiguous terms that are defined in this corpus, and which have an equivalent page in Viquipèdia[6], the Catalan Wikipedia. This process includes certain heuristics to ensure we are not mapping any noisy page (e.g. a redirection page) or a blank page. The end result of this mapping process is a set of definitional and non-definitional sentences extracted from Wikipedia in the Catalan language. Distractors (non definitions) are obtained

[6] ca.wikipedia.org

by randomly collecting sentences in the rest of a term's corresponding Wikipedia page where the target term is explicitly mentioned.

In what follows, we provide the reader with two examples of definitional and non-definitional senteces from our Catalan corpus, both referring to the term *iot* (yacht).

- **Def -** Un iot és una embarcació d'esbarjo o esportiva propulsada a vela o a motor amb coberta i amb cabina per a viure-hi.

  Def - *A yacht is a recreational or sports vessel propelled by a sail or by an engine, with deck and cabin to live in.*

- **Nodef -** Tot i això la majoria de iots a vela privats solen tenir una eslora de 7 a 14m, ja que el seu cost augment ràpidament en proporció a l'eslora.

- **Nodef -** *However, most private yachts propelled by sail usually have a length between 7 and 14 metres, as their cost increases quickly with regard to the length size*.

For training, we compile a subset of Wikicorpus, following the same method as for our validation dataset compiled from WCL. For each term and its corresponding Wikipedia article, we extract the first sentence, which can be safely assumed to be a definition. Distractors (*syntactically plausible false definitions*), are obtained by collecting those sentences where the term is explicitly mentioned.

Each definitional sentence is tagged with two labels, namely **B** for its first word, and **I** for the remaining words in the sentence. Conversely, non-definitional (distractor) sentences have all their tokens tagged with the **O** label.

3.3.2 Data Modeling

Both datasets are preprocessed using the part of speech tagger included in Freeling [Atserias et al., 2006a]. Since ours is a sequential labeling task, we take advantage of a powerful graphical model for sequential tagging, namely Conditional Random Fields (CRF) [Lafferty et al., 2001][7].

Formally, a sentence s is a sequence of word-level feature vectors, such that $s = [f_1, f_2 ... f_n]$, where $n = |s|$ and f_i receives a label $y \in \{\text{B, I, O}\}$. Features are computed, for each iteration, over both the current and the contextual tokens, in a $[-3, 3]$ window. This also allows for a finer grained evaluation, in that we explore the label-wise performance of the algorithm, with particular focus on the "B" label. This is important because, in a definitional text snippet, the

[7] We use the CRF++ package `https://taku910.github.io/crfpp/`.

first word usually corresponds to the definiendum, and therefore this can be useful information for downstream applications such as dictionary/glossary building, or dictionary example lookup. In what follows, we describe the features used to model each $f_i \in s$.

- **sur**: Surface form of the current word, with no normalization (i.e. no lower casing, as we want to keep capitalization to account for acronyms, abbreviations or name entities which may be indicative of definitional knowledge).
- **lem**: Word lemma (normalized).
- **pos**: A word's part of speech.
- **pos-prob**: The probability given to **pos** by Freeling.
- **bio-np**: First, we apply a simple shallow parsing stage over part of speech tags using the regular expression [JN]*N. Then, BIO tags are assigned to each identified noun phrase. Let us provide an example of what this tagging would look like in a sample sentence:
  - El⟨**b-np**⟩ verd⟨**i-np**⟩ és⟨o-np⟩ un⟨o-np⟩ dels⟨o-np⟩ tres⟨o-np⟩ colors⟨**b-np**⟩ primaris⟨**i-np**⟩ additius⟨**i-np**⟩
  - *Green*⟨**b-np**⟩ is⟨o-np⟩ one⟨o-np⟩ of⟨o-np⟩ the⟨o-np⟩ three⟨o-np⟩ primary⟨**b-np**⟩ additive⟨**i-np**⟩ colors⟨**i-np**⟩
- **def-tf** A frequency count for each word over the definitional sentences in our train set.
- **gen-tf** A frequency count for each word in a general purpose corpus, formed by newswire texts[8]. We base our approach on the intuition that certain words or expressions usually used to express definitional knowledge may show significantly lower frequency in generic language, such as "es considera" (*it is considered*) or "es defineix com" (*is defined as*).
- **def-tf*idf** We compute *term frequency*inverse document frequency* for each word over the definitional subset of our training corpus. Document frequencies are computed at sentence level. Formally:

$$tfidf(w, d, D) = tf(w, d) \times idf(w, D)$$

[8]For the interest of the reader, the corpus was initially available from `www.corpora.heliohost.org`, but the project seems to be discontinued.

where $tf(w, d)$ is the frequency of word w in document d. Likewise, $idf(w, D)$ is computed as follows:

$$\frac{|D|}{|\{d \in D : w \in d\}|}$$

where D is a document collection and $|D|$, its cardinality.

- **gen-tf*idf** Based on a similar intuition as in **def-tf*idf**, we compute this metric word-wise with the same corpus as in **gen-tf**.

- **termhood** This metric determines the likelihood of a single token to be part of a terminological unit, i.e. a domain-specific expression, showing a much higher occurrence in domain-specific corpora than in generic language [Kit and Liu, 2008]. The *termhood* metrics measures this intuition as follows:

$$\text{Termhood(w)} = \frac{\text{r}_\text{D}(\text{w})}{|\text{V}_\text{D}|} - \frac{\text{r}_\text{B}(\text{w})}{|\text{V}_\text{B}|}$$

Where $\text{r}_\text{D}(w)$ refers to the frequency-wise ranking of word w in the *specific* corpus (in this case, the definitional training data), and $\text{r}_\text{B}(w)$ refers to w's ranking in a generic corpus. Denominators denote the size of each corpus. If word w only appears in the general corpus, we set the value of Termhood(w) to $-\infty$, and to ∞ in the opposite case.

- **bio-D and bio-d** For each sentence (definitional or not) in our training corpus, we locate the first verb, and tag as *definiendum* (D) all tokens before it. Then, we tag as *definiens* (d) all words that come after, until the end of the sentence. Then, we use the information provided by the **bio-np** feature, obtaining a tagging like the following:

  - El⟨**b-definiendum**⟩ verd⟨**i-definiendum**⟩ és⟨o-definiens⟩ un⟨o-definiens⟩ dels⟨o-definiens⟩ tres⟨o-definiens⟩ colors⟨**b-definiens**⟩ primaris⟨**i-definiens**⟩ additius⟨**i-definiens**⟩

- **def-prom** We introduce the notion of *definitional prominence*, aiming at modeling the likelihood of a word w to appear in a definitional sentence ($s = def$). To this end, we consider its frequency in both definitional and non definitional sentences in our training corpus. Formally:

$$\text{DefProm(w)} = \frac{\text{DF}}{|\text{Defs}|} - \frac{\text{NF}}{|\text{Nodefs}|}$$

where DF $= \sum_{i=0}^{i=n}(s_i = \textit{def} \wedge w \in s_i)$ and NF $= \sum_{i=0}^{i=n}(s_i = \textit{nodef} \wedge w \in s_i)$. Similarly as with the *termhood* feature, in cases where a word w is only found in definitional sentences, we set the DefProm(w) value to ∞, and to $-\infty$ if it was only seen in non-definitional sentences.

- **D-prom** We also introduce *definiendum prominence* in order to model the intuition that a word appearing more often in position of potential *definiendum* might reveal its role as a definitional keyword. This feature is computed as follows:

$$\text{DP}(w) = \frac{\sum_{i=0}^{i=n} w_i \in \text{term}_D}{|DT|}$$

  where term_D is a noun phrase (i.e. a term candidate) appearing in potential definiendum position and |DT| refers to the size of the candidate term corpus in candidate definienda position.

- **d-prom** Similarly computed as **D-prom**, but considering position of potential definiens.

We will refer to this same feature set (which we denote as **SeqDEFeats**) in further experiments described in this thesis (Section 3.4). Unless otherwise noted, the number of features, learning algorithm and context windows remain the same.

Each feature vector is provied to a CRF learner, which models both label probabilities (i.e. how likely is a given feature vector to be tagged as BIO) and transition probabilities (i.e. how likely is a feature vector with BIO tag, to *transition into* the same or a different tag for the next token). We present in the following section label-wise results for our system, as well as several baselines resulting from ablation tests.

3.3.3 Evaluation

We compute Precision, Recall and F-measure for each of the three available labels. While for practical purposes, labels B and I could both be considered correct as in both cases both denote that a word *is part of* a definition, in our evaluation, only exact matches are considered true positives.

We evaluate four systems, which we describe as follows:

- **Baseline** This configuration only considers for training the **sur** feature, without looking at any contextual information.

- **C-1** It learns only linguistic features (**sur**, **lemma**, **pos**, etc.) over a $[-3, 3]$ window.

- **C-2** It learns only statistical features (**tf-def**, **tfidf-def**, etc.) over the same window as **C-1**.

- **C-3** It considers both linguistic and statistical features in the same context window.

Table 3.7 shows the results obtained with each of these configurations. Row identifiers refer to: (1) Whether the score is on Precision, Recall or F-Measure; and (2) Which of the BIO labels was being evaluated, or an average (M), which reflects the overall behavior of the system. We can observe that, starting from a baseline which only obtains 67.31 F-Score, we are able to substantially increase our numbers by including linguistic and statistical features, which obtain F=75.85 and F=75.68 respectively. Moreover, combining both feature sets, performance increases to F=86.69, which indicates that a combination of both linguistic and statistical feature sets contribute to competitive performance in DE systems. Note that this is an evaluation performed strictly at *word*-level, which means that an additional heuristic would have to be applied in cases where a candidate sentence contains words labeled both as definitional and non-definitional. If we aim at selecting only full sentences in a real application, we would look at the proportion of one class vs the other, and set a threshold to make an ultimate decision. For example, in our next contribution (**WeakDE**, Section 3.4), we follow the criterion to only tag as definitional those sentences where *all* words were labeled as such.

As for *error analysis*, we note certain patterns in the types of mistakes the model makes. For instance, it seems to misclassify as definitions sentences which (loosely) follow a genus-et-differentia structure, as in the following example, from the Wikipedia page corresponding to the term "divendres" (*Friday*):

> Tant és així que quan una persona és molt desgraciada es diu que deu ser nascuda en divendres.
> *In fact, when someone is very unfortunate, it is said that he/she must have been born on Friday.*

Here, the model gets triggered by the fact that the first word "Tant"[9] was tagged as a proper noun by Freeling, which combined with the context in which appears, i.e. followed by the verb *to be* (*és*), is a strong indicator of the sentence being a definition, considering the type of register (encyclopedic language) used in this experiment.

In terms of false negatives (FNs) (definition sentences incorrectly classified as being non-definitional), it is more difficult to identify a recurrent error pattern

[9]The phrase "tant és així" translates roughly into English as "in fact".

in which the model may incur. A qualitative analysis of the resulting classification hints towards the fact that a large portion of FNs are fairly *irregular* in their linguistic articulation. However, we have detected errors in the following cases:

- Chemical compounds: In cases where the defined term is a chemical compound, its acronym is mistakenly tagged as a sequence of proper nouns, and thus the part-of-speech sequence as well as definitional and statistic features fail to capture the sentence structure. Specifically, these sentences would have two noun phrases in definiendum position, a very rare case in definition sentences. For instance:

  > El triti, T o ^{3}H és un d'els isòtops de l'hidrògen.
  > *Tritium, T or ^{3}H, is one of the isotopes of hydrogen.*

  Here, the acronym "T" is tagged as a proper noun, a piece of information which propagates to other features reliant on part-of-speech, such as **bio-np** and **bio-D/bio-d**. Specifically, it produces two detected noun phrases at definiendum position (one very rare, being simply one capital "T"), which is rare in encylopedic definitions.

- Long definiendum-genus distance: We have identified several definitions where the heads of both definiendum and genus are very far apart, these sentences tend to be misclassified as non-definitional, as in the following case:

  > El **brandi** és el terme general utilitzat per nomenar la **beguda** alcohòlica feta a partir de vi de baixa qualitat, aiguardent o fins i tot most.
  > *Brandy is the general term used to name the alcoholic drink made from low-quality wine, schnapps or even must.*

  Having eight words separating *brandi* (brandy) and *beguda* (drink) makes this a fairly non-standard definition, and we hypothesize it may be the source of error. Further experiments where longer distances (or where distances are encoded in terms of syntactic dependencies) should be conducted to verify this hypothesis.

- "Stop-hypernyms": There are cases where a definition may not include an explicit genus, as in:

  > L'estat confessional és **aquell** que declara una religió concreta com a oficial, amb diversos graus de penetració en la vida pública.
  > *A confessional state is that which declares a specific religion as official, with different degrees of permeation in public life.*

	Baseline	C-1	C-2	C-3
P-B	67.50	89.29	80.68	**93.60**
R-B	51.72	57.47	**88.62**	85.87
F-B	58.57	69.93	84.47	**89.57**
P-I	58.49	84.89	72.25	**90.71**
R-I	49.58	51.82	**88.80**	83.48
F-I	53.67	64.35	79.68	**86.95**
P-O	88.03	**89.19**	77.24	79.36
R-O	91.43	**97.76**	53.08	88.23
F-O	89.79	**93.28**	62.92	83.56
P-Avg	71.34	87.78	76.72	**87.89**
R-Avg	64.24	69.01	76.83	**85.85**
F-Avg	67.31	75.85	75.68	**86.69**

Table 3.7: Results for each of the relevant labels (BIO), plus average results (*-Avg), in terms of Precision, Recall and F-score.

Here, the fact that there is no clear hypernym for "estat confessional" (confessional state), is a strong (and wrong) indicator that this is a non-definition sentence.

3.3.4 Conclusion

In this section, we have described and evaluated a set of experiments on DE which show two main outstanding features with respect to **DependencyDE** and **SemanticDE**, namely the methodological approach (sequence to sequence classification rather than sentence-level binary classification), and the language chosen for performing the experiments (Catalan). One of our main conclusions is that the proposed novel set of lexicographic and statistical features we introduce over definitional corpora contribute to the learning process, as shown by our ablation tests. In addition, we have provided insights for potential sources of misclassifiaction, both in terms of false positives (i.e. non-definition words incorrectly classified as pertaining to a definition) and false negatives. Our qualitative analysis opens specific directions for improvement, for example in improving the encoding of contexts (one of the powerful features of the CRF algorithm we used), as well as better preprocessing for handling rarer cases such as acronyms.

3.4 WeakDE: Weakly Supervised Definition Extraction

In the previous sections we have described experiments for DE in a supervised manner. In all of them, we took advantage of the manually annotated and validated WCL dataset (or an automatic adaptation to another language), which has become since its release a standard benchmarking for DE systems. However, experiments in this setting show two main inherent drawbacks. First, in a real word scenario, it is less likely for a definition to appear following the canonical *genus et differentia* model, and therefore a system trained only with encyclopedic information may fall short. Second, any model learned over the WCL dataset is inherently constrained by linguistic regularities showed in this type of textual genre, and therefore it may not be useful in a domain where language may evolve over time.

In our next contribution, we aim at bridging this gap via *Weakly Supervised DE*. We propose an approach which, from a starting set of encyclopedic definition seeds, self-trains iteratively and gradually fits its classification capability to a target domain-specific test set. Let us first discuss the data creation step, which is followed by the details of the algorithm, as well as evaluation results.

3.4.1 Corpus compilation

In terms of corpora, we part ways from previous approaches, which focused mostly on encyclopedic or technical documents. Prominent examples include German technical texts [Storrer and Wellinghoff, 2006], the IULA Technical Corpus (in Spanish) [Alarcón et al., 2009], the BNC corpus [Rodríguez, 2004], Wikipedia [Navigli and Velardi, 2010], ensembles of domain glossaries and Web documents [Velardi et al., 2008], or technical texts in various languages [Westerhout and Monachesi, 2007b, Przepiórkowski et al., 2007, Borg et al., 2009, Degórski et al., 2008, Del Gaudio et al., 2013].

Our proposed weakly DE system requires the following corpora: (1) a general-domain (encyclopedic) set of seeds of definitions (denoted as TS), and (2) a domain-specific development set, e.g. a collection of papers (DS). For our experiments, we use as TS the WCL dataset. Let us highlight the fact that, while this dataset includes semantic information manually annotated such as the definiendum or hypernymy, we do not exploit any of it, which makes the seed-construction step highly flexible as it only requires the sentence definition/non-definition class. We use as DS a subset of the ACL ARC corpus [Bird et al., 2008], processed with ParsCit [Councill et al., 2008]. In this dataset, a well-formedness confidence score is given to each sentence (as these come from *pdf* parsing and noise is introduced in the process). For example, one of the papers included in this anthology

corpus is [Yeh, 2000]. For illustrative purposes, we show a sentence (1) from the original paper, and its correspondent version in the ACL-ARC corpus (2), where noise (shown in italics) is introduced due to the *pdf* to text conversion.

> **(1)** One cannot directly compare the two systems from the descriptions given in Ferro et al. (1999) and Buchholz et al. (1999) ...
>
> **(2)** One can not directly *coral)* are the two systems from the descriptions given in Ferro et al . (1999) and Buchholz et al ...

We exploit the given confidence score and keep 500k sentences with a well-formedness confidence score of over .95. Still, noise is inevitably present even at such a restrictive threshold, and it stems from issues related to font formatting, footnotes, presence of equations or examples from languages with non-ascii encoding.

For evaluation, we use two datasets: First, a set of 50 abstracts of papers in the field of NLP[10]. Here, the target term is defined in the first sentence, and additional information may appear in the form of "syntactically plausible false definitions". Second, the W00 [Jin et al., 2013] corpus, a subset of the ACL Anthology manually annotated with definitions, and which includes highly variable definitions both in terms of content and syntax. The MSR-NLP is a manually constructed small list of 50 abstracts in the NLP field, amounting to 304 sentences: 49 definitions and 255 non-definitions. They are extracted from the Microsoft Academic Research website[11], where abstracts including a definition provide a "Definition Context" section. This small dataset complies with the stylistic requirements of academic abstract writing, i.e. the use of well-developed, unified, coherent and concise language, and understandability to a wide audience[12]. A different register can be found in the W00 dataset, which includes many definitional sentences that are highly domain-specific, sometimes including the definition of a very specific concept, and showing higher linguistic variability (e.g. the definiendum might not appear at the beginning of the sentence, and unlike most abstracts, citations might be present). We illustrate this difference with two sentences containing a definition from the MSR-NLP (1) and the W00 (2) corpora:

(1) The Hidden Markov Model (HMM) is a probabilistic model used widely in the fields of Bioinformatics and Speech Recognition .

(2) This corpus is collected and annotated for the GNOME project (Poesio, 2000), which aims at developing general algorithms for generating nominal expressions

[10]Henceforth, we refer to this corpus as the *MSR-NLP* dataset.
[11]http://academic.research.microsoft.com/
[12]http://www.cameron.edu/~carolynk/Abstracts.html

Note that in the case of (2), only the sequence "GNOME project aims at developing general algorithms for generating nominal expressions" is labeled as definition in the original dataset. In this chapter a definitional sentence is generalized as *being* or *containing* a definition, which enables casting the task as a sentence-classification problem.

Intuitively, we would expect a general-purpose DE system to be more likely to label sentence (1), as it includes the required elements for a canonical genus-et-differentia definition. This motivates our experiments, where we attempt to fit a model iteratively to be able to perform better in sentences like (2).

3.4.2 Data modeling

We approach the DE task as a sentence classification problem, where a sentence can be either a definition (*def*) or not (*nodef*). However, instead of modeling sentence-level features like sentence length or depth of the parse tree, we rather encode word-level features in order to exploit individual items' characteristics in terms of position within the sentence, frequency or relevance in a definition corpus. These word-level features are used for classifying each word in a sentence (*def*|*nodef*). This is a similar setting as the one of **SequentialDE**. In fact, the same **SeqDEFeats** configuration is used in both experiments.

Iter	Best definition in DS	MSR-NLP			W00		
		P	R	F	P	R	F
1	A term is a word or a word sequence	100	9.09	16.68	65.38	1.25	2.47
10	An abbreviation is defined as a shortened form of a written word or phrase used in place of the full form	83.13	44.4	57.88	69.84	11.35	19.53
120	A bunsetsu is one of the linguistic units in Japanese and roughly corresponds to a basic phrase in English	25.5	90.71	39.81	60.71	69.68	64.89
182	That is to say a site is a candidate site when it is found to have either an English page linking to its Chinese version or a Chinese page linking to its English version	22.92	92.53	36.74	62.55	76.63	68.88
200	Figure 1 and Figure 2 present the overall system configuration and data flow of the integrated system	23.34	96.72	37.6	62.27	78.45	69.43

Table 3.8: Definitions extracted throughout the bootstrapping process from the ACL ARC corpus and P/R/F results at that iteration on the two evaluation corpora (without post-classification heuristics, described in Section 3.4.3). Note the gradual increase in syntactic and terminological variability in the extracted definitions.

We adopt two extraction strategies depending on whether we operate over DS or any of the two evaluation corpora (MSR-NLP and W00). In the case of

DS, the goal is to extract complete high-quality definitional and non-definitional sentences. Therefore, we only consider as potential candidates for bootstrapping those sentences where all the words have the same label (i.e. discarding, for example, a 10-word sentence where nine are tagged as *def* and one as *nodef*). This is in fact the most frequent case by a large margin, so we are confident that there are very few potentially relevant sentences being left out. Since evaluation is carried out at word level, this constraint does not apply.

3.4.3 Bootstrapping

As noted earlier, the initial TS consists of the WCL dataset, which makes our model suitable for DE in well-formed encyclopedic texts. However, our hypothesis that it would perform poorly in a linguistically more complex setting (e.g. in a corpus like the W00 dataset) is confirmed by the results at iteration 1 (see Table 3.8). Our bootstrapping approach is aimed at gradually obtaining a better fit model for W00, starting from our generic baseline trained exclusively on the WCL corpus. The following description of our approach is summarized in Algorithm 1.

As mentioned above, TS is a manually labelled dataset where each sentence $s \in S$ is given a label $d \in D = \{def, nodef\}$. Likewise, DS is an unlabelled subset of the ACL-ARC corpus, which amounts to 500k sentences. The first step is to initialize (1) the training set vocabulary V, which simply contains all the words in TS; and (2) the feature set F associated to each word $w \in V$. Then, for each iteration until we reach 200, the algorithm extracts the best-scoring sentences as predicted by our CRF-based classififer (recall that only sentences where all words are assigned the same label are considered) for both labels *def* and *nodef* (s' and s'' respectively), and uses them to increase the initial feature set and vocabulary[13]. Next, it removes s' and s'' from DS, trains and evaluates a model on both the MSR-NLP and the W00 datasets, and repeats until it reaches our manually set end point.

One important aspect to consider is that increasing the size of the training data does not have an effect of the features associated to a word. Incorporating definitions having concepts related to the target domain (NLP in our case) is a step forward, but their definitional salience remains the same, as they were calculated before initializing the bootstrapping algorithm. For this reason, we include a feature update step at iteration 100, our motivation being that, for evaluation purposes, we will have the same number of iterations before and after such step. It consists in resetting F to $\emptyset$ and recalculating it. We hypothesize that the new feature values can reflect better the linguistic idiosyncrasies of a domain-specific definitional

[13]The "stop at 200th iteration" is a stopping criterion arbitrarily set and in the future we plan to investigate longer learning cycles.

Algorithm 1 Bootstrapping for DE

Require:

$TS = \{(S, d \in D)\}$ Initial labelled train seeds.
$DS = \{S\}$ Subset of the ACL-ARC corpus.
MSR-NLP: Test set 1.
W00: Test set 2.

$V := \{w : \exists\,(s, d) \in TS \wedge w \in s\}$
$F := \{f_{TS}\,(w) : w \in V\}$
for $i = 0, i < 200, i++$ **do**
  $s' = argmax_{s \in DS}\ P\,(s = def)$
  $s'' = argmax_{s \in DS}\ P\,(s = nodef)$
  for $w \in s' \cup s''$ **do**
    if $w \notin V$ **then**
      $F = F \cup \{f_{TS}\,(w)\}$
      $V = V \cup \{w\}$
    end if
  end for
  $TS = TS \cup \{(s', def)\,, (s'', nodef)\}$
  $DS = DS \setminus \{(s', def)\,, (s'', nodef)\}$
  if $i = 100$ **then**
    $F = \emptyset$
    for $w \in V$ **do**
      $F = F \cup \{f_{TS}\,(w)\}$
    end for
  end if
$model_i = trainModel\,(TS_i, F_i)$
$evaluateModel\,(model_i, \{\text{MSR-NLP}, \text{W00}\})$
end for

corpus. After 200 iterations, our bootstrapped dataset TS_{boot} includes the original training data and 400 new sentences: 200 definitions and 200 non-definitions. As the bootstrapping process advances, s' and s'' show greater linguistic variability because the training data includes more non-canonical definitions (Table 3.8).

Finally, our last step consists in applying a post-classification heuristic inspired by [Cai et al., 2009]. It consists in a set of rules for label-switching aimed at increasing recall without hurting precision significantly. Let w_i be a word classified as not being part of a definition (*nodef*) at iteration i, we can rectify its class (w_i^{new}) to being part of a definition (*def*) as follows:

$$w_i^{new} = \begin{cases} def & \text{if } P_{def}(w_i) > \theta \\ def & \text{if } P_{nodef}(w_i) < \lambda, w_i^{\text{syn}} = \texttt{pred} \end{cases}$$

Where w_i^{syn} refers to the dependency relation of the word examined at iteration i, and `pred` is the *predicative complement* syntactic function of the word.

Our goal is to increase the number of *def* words in a sentence in cases where they were discarded by a small margin. We hypothesize that this could be particularly useful in "borderline" cases (some words classified in a sentence as *def*, some as *nodef*), where this heuristics helps our algorithm to make a decision always favouring definition labelling over non-definition. As for the constants, θ and λ are empirically set to .35 and .8 respectively after experimenting with several thresholds and inspecting manually the resulting classification.

3.4.4 Evaluation

We evaluate the performance of our approach at each iteration on both datasets (MSR-NLP and W00) using the classic Precision, Recall and F-Measure scores. All the scores reported in this chapter are at word-level.

The learning curves shown in Figure 3.4 demonstrate that our approach is suitable for fitting a model to a domain-specific dataset starting from general-purpose encyclopedic seeds. Unsurprisingly, performance on the MSR-NLP corpus drops soon after reaching its peak due to the fact that the training set gradually becomes less standard. Interestingly, the feature-update step has a dramatic influence in performance in both corpora: On one hand, the performance peak in a dataset with less linguistic variability (MSR-NLP) is reached early, and after iteration 100, where the feature update step occurs, Precision decreases, while Recall remains the same. On the other hand, the numbers in the W00 dataset are fairly stable until iteration 100, where a significant improvement in both Precision and Recall is achieved.

Let us look first at the results without applying recall-boosting post-classification heuristics: The performance of our models decreases in the MSR-NLP corpus after a few iterations (our best model is reached at iteration 23, where F=76.23), and this situation is unsurprisingly aggravated by the feature update step. However, our results improve significantly in the W00 dataset[14] after feature updating. Our best-performing model reaches F=70.72 at iteration 198.

Moreover, we observed a minor improvement after incorporating the label-switching heuristics in both corpora. Specifically, for the MSR-NLP corpus the improvement was from the aforementioned F=76.34 to F=77.46, while in the W00

[14]Note that since the W00 corpus is also a subset of the ACL ARC dataset, we first confirmed that it did not overlap with our dev-set.

dataset, it improved from F=70.72 to F=71.85. Tables 3.9 and 3.10 show Precision, Recall and F-Score for our best models in both datasets.

These numbers confirm that we are able to generate a domain and genre-sensitive model provided we have a development set available of similar characteristics. The discrepancy in terms of performance as the bootstrapping algorithm advances is an indicator that the models we obtain become more tailored towards the specific corpus, and therefore less apt for performing well in the encyclopedic genre. Our approach seems suitable for partially alleviating the lack of manually labelled domain-specific data in the DE field.

Let us also refer to the importance of having a development set as close as possible to the target corpus in terms of register and domain, and with a reasonable level of quality. In relation to this, we also performed experiments with a development set automatically constructed from the Web, but due to lack of preprocessing for noise filtering, results were unsatisfactory and therefore unreported in this dissertation.

	Iteration	P	R	F
Pre-PCH	198	62.69	81.11	70.72
Post-PCH	198	62.47	82.01	71.85

Table 3.9: Best results on the W00 dataset before (Pre-PCH) and after (Post-PCH) applying the post-classification heuristics.

	Iteration	P	R	F
Pre-PCH	23	80.69	72.24	76.23
Post-PCH	20	78.2	76.7	77.44

Table 3.10: Best results on the MSR-NLP dataset before (Pre-PCH) and after (Post-PCH) applying the post-classification heuristics.

As for comparative evaluation, we cannot contrast our results directly with the ones reported in [Jin et al., 2013], since while in both cases word-level evaluation is carried out, in our case we generalized all the words inside a sentence containing a definition to the label *def*. In addition, as it is pointed out in [Jin et al., 2013], only in [Reiplinger et al., 2012] there is an attempt to extract definitions from the ACL ARC corpus, but their evaluation relies on human judgement, and their reported coverage refers to a pre-defined list of terms.

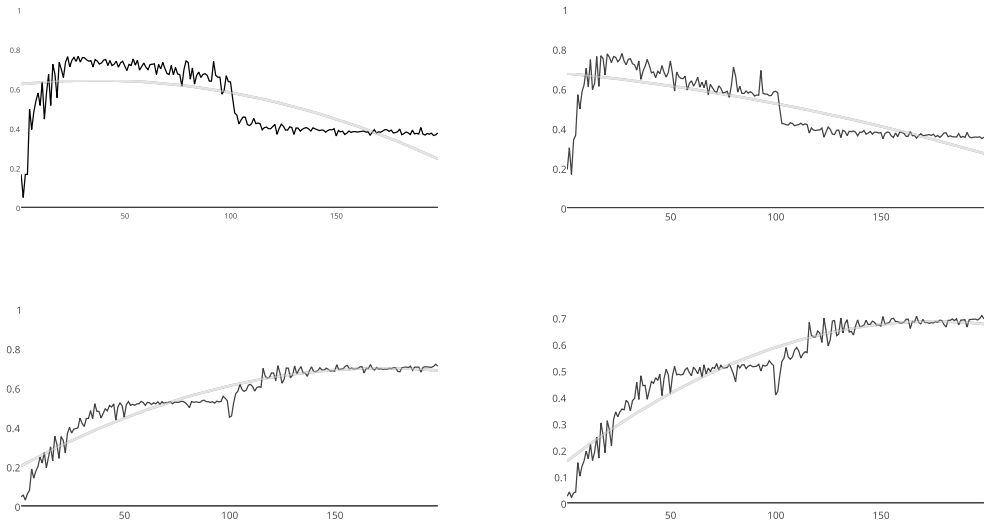


Figure 3.4: F-Score against iteration on the MSR-NLP (top row) and W00 datasets (bottom row), with bootstrapping + post-classification heuristics (left column) and only bootstrapping (right column).

In general, the results we report are consistent with the ones obtained in previous work for similar tasks. For instance, prior experiments on the WCL dataset showed results ranging from F=54.42 to F=75.16 [Navigli and Velardi, 2010, Boella et al., 2014]. In the case of the W00 dataset, [Jin et al., 2013] reported numbers between F=40 and F=56 for different configurations. Since the availability of manually labelled gold standard is scarce, other authors evaluated Glossary/Definition Extraction systems in terms of manually assessed precision [Reiplinger et al., 2012, De Benedictis et al., 2013].

3.4.5 Feature analysis

In order to understand the discriminative power of the features designed for our experiments, we computed Information Gain. We did this for the original training set TS, and for the training set resulting at iteration 200 TS_{boot}. Then, we captured the top 30 features and averaged their Information Gain score over all the available contexts. We compare these features in both datasets TS and TS_{boot} (see Figure 3.5).

We observe an improvement of definitionally-motivated features after iteration 100, which combined with the gradual improvement in performance in the W00 dataset, suggests that **def_prom** and **d_prom** contribute decisively to domain-specific DE, while **D_prom** proved less relevant. Note that in our setting, we do

not focus in term/definition pairs, but rather a full-sentence definition. Therefore, we do not know a priori which term is the definiendum, and thus we do not perform a generalization step to convert it to a wildcard, which is common practice in the DE literature [Navigli and Velardi, 2010, Reiplinger et al., 2012, Jin et al., 2013, Boella et al., 2014]. This provokes high sparsity in **D_prom** and we hypothesize that this may be the reason for this feature to not gain predictive power after many iterations or the feature update step.

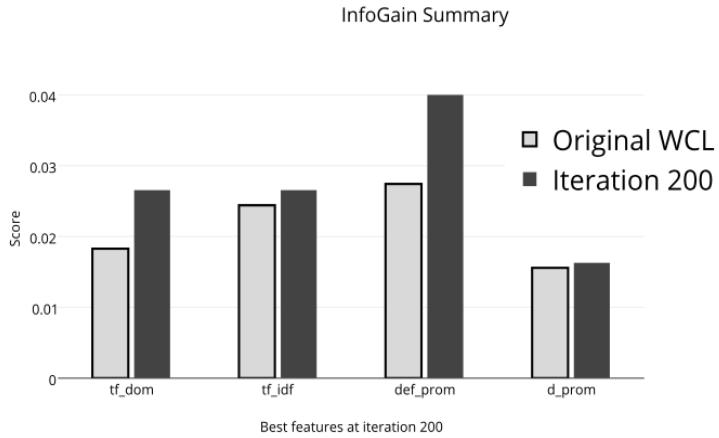


Figure 3.5: Information Gain for the best features at the end of the bootstrapping process. Note the substantial improvement in def_prom (definitional prominence).

3.4.6 Conclusion

In this section, we have described a weakly supervised DE approach that gradually increments the size of the training set with high quality definitions and clear examples of non-definitions. Two main conclusions can be drawn: (1) The definition-aware features we introduce show, in general, high informativeness for the task of DE; and (2) Our approach is valid for generating genre and domain specific training data capable of fitting corpora, even though this differs greatly in terms of content and register from the encyclopedic genre. In addition, a small and focused benchmarking dataset of real-world definitions in the NLP domain has been released, which can be used both for linguistic and stylistic purposes and for evaluating DE systems.

As for the limitations detected in this system, as derived from our exprimental results, we have identified three main areas where **WeakDE** could improve. First, by providing a more thorough evaluating on longer cycles, and ideally,

re-computing statistical features over training data at every iteration. Second, it seems natural to also incorporate, in addition to corpus-based features, distributional and dependency information, which on their own we have shown that work well for identifying definitions in corpora. And third, it would be interesting to expand this implementation to capture highly likely *sequences of tokens* with high probability of being definitional (i.e. do not reformulate the task as sentence classification using word-level information).

Chapter 4

HYPERNYM DISCOVERY

In this chapter, we cover our two main contributions towards the identification of hypernymic relations in corpora. Our first contribution, **DefinitionHypernyms**, explores the extent to which a machine learning approach which extensively exploits syntactic information can be applied to the task of, given a textual definition, discover the text fragment in which the hypernym of the definiendum is mentioned. Our second contribution, **TaxoEmbed**, is a supervised distributional approach which takes as input a concept and a domain of knowledge (e.g. `tourism`), and returns a ranked list of its most likely hypernyms, obtained from all the available vocabulary in a word embeddings model.

4.1 DefinitionHypernyms: Combining CRF and Dependency Grammar

Previous contributions for discovering hypernyms in definitions where the WCL dataset (presented and described in Chapter 2) was used as benchmark [Navigli and Velardi, 2010, Boella and Di Caro, 2013] combined machine learning and lexico syntactic cues. We improved these systems by putting forward a sequential approach based on CRF. The main idea is to iterate token-wise over a candidate sentence, and tag each token as being at the beginning (**B**), inside (**I**), or outside (**O**) a hypernym. This is a similar idea as for **WeakDE** (Section 3.4), but in this case the *sequence to sequence learning* scenario fits perfectly to our task. Input instances are sentences which are already being identified as definitions, and the challenge is to identify the best hypernym for a given definiendum.

For this task, we trained a model strongly reliant on linguistic cues, as provided by a sentence' syntactic dependencies. For the specific case of identifying hypernyms in definitions, this approach seemed particularly suitable considering, for example, that over 98% of the definitions in the dataset have one word with

the prd syntactic function. Additionally, we found over 850 cases where the word with prd function was a direct dependent of the root verb, and it also was the first word of a manually tagged hypernym: this means that 46% of the term-hypernym relations in this dataset would be extracted applying a simple mapping rule. This syntactic consistence, together with the good results shown previously by machine learning approaches in this and other related tasks, motivates us to leverage syntactic information as input features to our system.

4.1.1 Features

Our feature set is a combination of linguistic, syntactic and stochastic information. We use a similar set of features as in the **SequentialDE** and **WeakDE** systems (Chapter 3). Specifically, those shared by both approaches are (recall they are computed at word level): (1) A word's surface form, part of speech and dependency relation with its syntactic head; and (2) morphosyntactic chains. The set of novel features we introduce for this specific model are the following:

1. **Head Id (*headID*) and Dependency Relation (*depen*)**: These two features refer to the syntactic function of the current word and the unique identifier of its governor or head. For example, subject (sbj), object (obj), predicative complement (prd) or nominal modifier (nmod).

2. **Definiendum (*term*) and definiens (*def-nodef*)**: Whether the word is a definiendum term (i.e. it matches exactly the Wikipiedia page title to which the text snippet belongs to), and whether such word is part of the definiens. We apply a simple heuristic rule that tags all words after the first verb of the sentence as definiens.

3. **PageRank (*p-rank*)**: We compute the popularity of a node in a sentence with the PageRank algorithm (this is achieved by considering a parsed sentence as an undirected acyclic graph, where each word corresponds to a single node). Our intuition is that hypernyms in encyclopedic definitions usually have a higher number of modifiers than the rest of the words in the definition, and therefore a PageRank-based metric should be helpful to model this salient characteristic. As in previous graph-computation operations in this dissertation, we use the Python library NetworkX [Hagberg et al., 2008].

4. **Node Outdegree (*outdgr*)**: The out-degree of a node in a syntactic dependency tree is equal to the number of dependents. The intuition is similar as in the previous case, but here we explicitly aim at encoding only first-level modifiers for each node.

5. **Syntactic Salience (*syntS*)**: In addition to the above features, we are interested in a more general metric to assess the extent to which a word and its associated linguistic information describes a textual genre. Motivated by the fact that in textual definitions not only are hypernyms likely to appear, but they show syntactic regularities, we count how many times a word is part of the most frequent subtrees in the dataset taking into consideration different ranges of linguistic information (from only the word's surface form to subtrees including the word's surface form, part-of-speech and syntactic funtion).

Numeric features such as node degree, pagerank or syntactic salience are discretized, i.e. within a range between the smallest and highest score, each value is assigned a discrete type between 1 and 10. This coarse-grained set of attributes allows us to understand better each feature's effect in the learning process and perform more sensible error analysis.

Having prepared our sets of features, these are used for training and evaluating a CRF classifier. Given the inherent ability of CRF for learning prior and posterior contextual information in a sequential classification task, we design three experiments where three context windows are considered: [-1,1], [-2,2] and [-3,3]. For each window, we design feature sets incrementally adding one feature at a time (see in Table 4.1 a matrix outlining all the feature sets used in our experiments). We report scores derived from 10-fold cross validation.

4.1.2 Recall-Boosting heuristics

After manually inspecting the output of the classifier, we observe that there are cases in which the discrepancy between the predicted label and the gold standard can be, at best, questionable. In fact, [Boella et al., 2014] mention issues derived from the complexity of what actually constitutes a valid hypernym in a textual definition and its effect on the quality of the annotation of the WCL dataset (introduced and described in Chapter 2). Among others, they refer to incorrect relationships, e.g. incorrectly annotating a meronym as a hypernym, or inconsistent modifier attachment, e.g. cases where the same modifier attached to two semantically-related concepts is sometimes included as part of a multiword hypernymic phrase, and others not.

	sur	lemma	pos	headID	depen	def-ndef	term	p-rank	outdgr	chains	syntS
FeatSet1	x										
FeatSet2	x	x									
FeatSet3	x	x	x								
FeatSet4	x	x	x	x							
FeatSet5	x	x	x	x	x						
FeatSet6	x	x	x	x	x	x					
FeatSet7	x	x	x	x	x	x	x				
FeatSet8	x	x	x	x	x	x	x	x			
FeatSet9	x	x	x	x	x	x	x	x	x		
FeatSet10	x	x	x	x	x	x	x	x	x	x	
FeatSet11	x	x	x	x	x	x	x	x	x	x	x

Table 4.1: Different feature sets adding one feature at a time.

This motivated a post-classification heuristic in a similar fashion as in Section 3.4. Specifically, let $token_i$ be a word classified as not being part of a hypernymic phrase (O). We perform a label-switching step replacing its current label with either B or I, yielding $token_i^{update}$. The following conditions are considered:

$$token_i^{update} = \begin{cases} \text{B} & \text{if } P_{\text{B}}(token_i) > \theta \wedge P_{\text{B}}(token_i) > P_{\text{I}}(token_i) \\ \text{I} & \text{if } P_{\text{I}}(token_i) > \theta \wedge P_{\text{I}}(\text{token}_i) > P_{\text{B}}(token_i) \\ \text{B} & \text{if } P_{\text{O}}(token_i) < \lambda \wedge token_i^{Synt} = \texttt{prd} \end{cases}$$

Where token_i^{Synt} refers to the syntactic function of the word token_i, and where θ and λ are constants empirically set to the same values as in the **WeakDE** approach (.35 and .8 respectively).

These heuristics contribute to increase F-Score in feature sets 1 and 2 when considering [-1,1] contexts. Likewise, F-Score also improves after this step in feature sets 1, 2 and 3 when considering [-2,2] and [-3,3] contexts. In many configurations, Recall improves almost 10 points, and while in strict comparison against gold standard the drop in precision affects negatively the overall F-Score in the majority of feature sets considered, we found that in some cases our greedier approach detected a better hypernym than the one manually annotated in the gold standard. Let us look at the following sample definition:

> An *abzyme* (from antibody and enzyme), also called catmab (from catalytic monoclonal antibody), is a **monoclonal antibody** with catalytic activity.

In the manually annotated dataset, the hypernym is "antibody", and in the majority of our experiments our algorithm identifies "monoclonal antibody", thus producing a false positive in our word-level evaluation. However, it is not clear that "antibody" is a better hypernym for "abzyme" than "monoclonal antibody". In fact, there is a Wikipedia entry for "monoclonal antibody"[1], which suggests that the prediction of our algorithm is correct since "monoclonal" is not a property of "antibody" but rather defines a monosemic type of antibody.

4.1.3 Evaluation

We evaluated at token-level in terms of Precision, Recall and F-Measure by adding one feature at a time to the CRF-trained model. These results are shown in Table 4.2 (DC for the CRF-based *definition configuration as is*, and Boost for the *recall-boosted* configurations). Four main conclusions can be drawn: (1) Word-level morphosyntactic features are highly informative in the encyclopedic genre (see the boost in performance after these features are added to the model), which reinforces our intuition that encyclopedic definitions do follow follow certain syntactic patterns and show regularities that can be exploited; (2) The best-performing model (highest F-Score) is *FeatSet8*, which includes all linguistic features, definitional information, and page-rank; (3) Unsurprisingly, the best performing models for each feature set are those including the largest context window ([-3,3]); and (4) Recall-Boosting post-classification rules increase F-Score only in the most basic feature sets. We provide further discussion on feature relevance in Section 4.1.3.1.

Finally, we compared our best-performing model with existing state-of-the-art systems reported in the literature. Firstly, the Word-Class Lattices algorithm [Navigli and Velardi, 2010], and secondly an approach conceptually similar to ours that also modelled the problem in terms of syntactic dependencies [Boella et al., 2014] (Table 4.3).

As for *error analysis*, similarly as in Section 3.3, it seems easier to infer systematic patterns of errors in false positives (words or sequences of words misclassified as being hypernyms) rather than false negatives (the opposite case). First, let us recall the already mentioned case of longer hypernyms detected by our system. This may be due to either to the annotation procedure followed during the construction of the WCL corpus, but also derived from the fact that some apparently valid hypernyms do not have a corresponding Wikipedia page. Consider the following example:

An alexandrine is a line of *poetic* ***meter***.

[1] http://en.wikipedia.org/wiki/Monoclonal_antibody

		DC-1:1	DC-2:2	DC-3:3	Boost-1:1	Boost-2:2	Boost-3:3
	P	48.51	65.22	70.33	30.35	40.22	46.46
$FeatSet1$	R	31.96	41.45	48.34	65.44	72.06	75.23
	F	38.49	50.64	57.25	41.43	51.6	57.41
	P	49.36	61.87	66.55	32.12	41.77	47.84
$FeatSet2$	R	33.92	44.33	51.13	64.52	71.26	74.27
	F	40.17	51.58	57.79	42.85	52.66	58.18
	P	64.93	67.58	72.65	41.98	49.38	55.32
$FeatSet3$	R	33.17	47.23	56.62	64.68	71.34	75.36
	F	43.85	55.54	63.31	50.86	58.34	63.79
	P	70.32	72.41	74.32	48.05	53.2	58.47
$FeatSet4$	R	44.98	55.37	60.87	70.07	74.63	76.37
	F	54.8	62.71	66.89	56.99	62.1	66.22
	P	76.04	75.85	76.17	56.03	58.67	62.05
$FeatSet5$	R	54.33	61.52	64.73	74.68	76.86	78.49
	F	63.34	67.88	69.94	64.01	66.51	69.31
	P	80.19	82.99	**84.22**	62.44	68.14	73.08
$FeatSet6$	R	63.26	72.04	75.69	79.85	82.42	**84.99**
	F	70.68	77.12	79.71	70.04	74.59	78.58
	P	80.08	83.05	84.15	62	68.43	73.25
$FeatSet7$	R	63.15	72.04	75.51	79.57	82.47	84.96
	F	70.57	77.13	79.58	69.66	74.77	78.67
	P	80.11	82.56	84.01	62.67	68.34	72.59
$FeatSet8$	R	63.47	72.02	76.12	79.68	82.27	84.82
	F	70.79	76.91	**79.85**	70.13	74.64	78.22
	P	79.94	82.31	83.82	62.01	68.04	72.44
$FeatSet9$	R	63.68	72.06	75.94	79.58	82.26	84.64
	F	70.86	76.82	79.66	69.67	74.46	78.06
	P	79.6	81.86	83.6	62.4	68.64	72.71
$FeatSet10$	R	63.86	71.35	75.74	79.02	81.69	84.51
	F	70.85	76.23	79.47	69.7	74.59	78.15
	P	79.72	81.87	83.43	62.69	68.7	73.1
$FeatSet11$	R	64.48	71.62	75.36	79.22	82.13	84.16
	F	71.28	76.03	79.17	69.94	74.81	78.22

Table 4.2: Performance of DefinitionHypernyms at three context windows ([-1:1], [-2:2] and [-3:3]).

	Precision	Recall	F-Score
N&V WCL-1	77	42.09	54.42
N&V WCL-3	78.58	60.74	68.56
B&DiC	83.05	68.64	75.16
DefinitionHypernyms	**84.01**	**76.12**	**79.85**

Table 4.3: Comparative Evaluation between our best performing model ($FeatSet8$ with no post-classification heuristics) and the results reported in [Navigli and Velardi, 2010] and [Boella et al., 2014].

The hypernym in italics is the prediction of our algorithm, while the bold hypernym is the gold standard. It could be argued that using "poetic meter" as a hypernym for *alexandrine* is at least an option as valid as simply using "meter"[2].

Other cases of error, however, stem clearly from a convoluted morphosyntactic structure in the definition, as in the following case.

> Bioterrorism is **terrorism** by international release or *dissemination* of biological agents (bacteria, viruses or toxins); these may be in a naturally-occurring or in a human-modified form.

Here, our model made two mistakes. First, it disregarded *terrorism* as a hypernym for *bioterrorism*, while selecting another term for it (*dissemination*). As mentioned, this may be due to the several nested prepositional phrases that occur in the sentence. Finally, let us refer to cases where a term's hypernym has a high number of modifiers, which results in discrepancies in terms of selecting a nested noun phrase which is acting as a modifier of a higher head noun as valid hypernym, as in the following example:

> A broch is an **iron age drystone** *hollow-walled structure* of a type found only in Scotland.

The bold hypernym (*iron age drystone*) was not selected by our model, which however captured *hollow-walled structure* as a valid hypernym of the term *broch*. It seems that selecting one option over the other may owe to contextual facts which are not accounted for only in this textual piece (e.g. the purpose of any downstream task, or the domain in which this definition appears).

[2]In fact, this hypernym is used in another definition of a type of "alexandrine", namely the "French alexandrine" `en.wikipedia.org/wiki/French_alexandrine`.

4.1.3.1 Information Gain

Recall that Information Gain (IG) measures the decrease in entropy when the feature is present vs. absent. We rank our features according to their IG score. We denote each feature as `featurename`Position$X$ =`featurevalue`, where X is the relative position of that feature at the current iteration during training. For instance, the feature deprelPosition-1=`nmod` means that the algorithm is considering, for the prediction of word w_i (where i is its position in the sentence), whether the previous word ($i-1$) functions as a noun modifier (`nmod`). Looking at the best features in our model (Table 4.4), we can conclude the following[3]: (1) Hypernym extraction algorithms improve by a huge margin if provided with syntactic information; (2) Previous work has demonstrated improvement in the task of DE by informing the classifier with terminological information [Jin et al., 2013]. This seems to hold the other way round as well; (3) We also observe an interesting set of features clumped together with the same value and the same Information Gain score. These are *no_value* feature scores, which means that the context specified (e.g. $i = -1$) is null due to the current iteration being at the beginning or end of the sentence. This might point to hypernyms being consistently mentioned at a certain position in a sentence; (4) the discretization of our numeric values might have been too coarse-grained for being discriminative enough in a classification task. Finally, (5) After looking at the last row in Table 4.4, we observe the highest graph-based ranking feature (in position 24) referring to the fact that a word has a child with `nnp` part-of-speech and dependency relation `sbj`.

4.1.4 Conclusions

We have described a system for Hypernym Extraction from textual definitions in the WCL corpus. We experimented with linguistic, definitional and graph-based features which operated over the sentence parse tree. Our best model achieves better results than existing approaches on this dataset. The experiments carried out also showed that linguistic and definitional information are by far the most important features in our configuration, and only few exceptions among the graph-based features can be considered informative.

Finally, these experimental results open several avenues for future work. For example, we would like to draw statistics to measure accurately how many of the false positives in which our approach incurred after applying the Recall-Boosting heuristics could be correct hypernyms by looking at generic encyclopedias or domain-specific knowledge bases. Also, since the contribution of graph-based

[3]The full set of features and their Information Gain rank can be accessed at: The complete Information Gain score list can be accessed at `bitbucket.org/luisespinosa/definitionhypernyms`. There are 2,111 features with non-zero IG score.

Rank	Feature	InfoGain
1	deprelPosition0=PRD	0.0682345
2	posPosition0=nn	0.0538957
3	deprelPosition-1=NMOD	0.0517277
4	defnodefPositiond0=def	0.0349189
5	defnodefPosition0=nodef	0.0349189
6	defnodefPosition1=def	0.0349189
7	headIDPosition-1	0.0320474
8	deprelPosition-2=ROOT	0.0315236
9	defnodefPosition+1=nodef	0.0300525
10	defnodefPosition-3=nodef	0.0300255
24	chainsPosition0=dt_NMOD&nnp_SBJ	0.0182301

Table 4.4: Selected best features for Hypernym Extraction. Each feature reads as follows: $featureName$Position=value, where Position refers to the context in which appears at the current iteration. For instance, Position=-1 refers to one word before the word at the current iteration.

features was very limited, we would like to explore with finer-grained discretization heuristics as well as with the raw numeric values. Finally, it would be interesting to test our approach on other large datasets, such as WiBi [Flati et al., 2014] or the Linked Hypernyms Dataset [Kliegr, 2014].

4.2 TaxoEmbed: Supervised Distributional Hypernym Discovery via Domain Adaption

In this section, we explore an approach for supervised distributional HD which is purely distributional, i.e. it does not rely on any corpus co-occurrence of candidate hyponym-hypernym pair (unlike, for example, our previously discussed **DefinitionHypernyms** algorithm). We also propose to use a very large KB containing thousands of term-hypernym relations both for training and evaluation (with different train and test splits), namely Wikidata. The remainder of this section is organized as follows. First, we introduce the resources we exploit in the design and training of TaxoEmbed[4], then we describe how we incorporate training data from various resources, along with a description of the training algorithm, and conclude with evaluation and conclusion.

4.2.1 Preliminaries

TAXOEMBED leverages the vast amounts of training data available from structured and unstructured knowledge resources, along with the mapping among these resources and a state-of-the-art vector representation of word senses.

BabelNet constitutes our sense inventory, as it is currently the largest single multilingual repository of named entities and concepts, integrating various resources such as WordNet, Wikipedia or Wikidata. As in WordNet, BabelNet is structured in synsets. Each synset is composed of a set of words (*lexicalizations* or *senses*) representing the same meaning. For instance, the synset referring to *the members of a business organization* is represented by the set of senses *firm, house, business firm*. BabelNet contains around 14M synsets in total. We exploit BabelNet as (1) A repository for the manually-curated hypernymic relations included in **Wikidata**; (2) A semantic pivot of the integration of several Open Information Extraction (OIE) systems into one single resource, namely **KB-UNIFY**(we provide further details about KB-U in Section 6.2); and (3) A sense inventory for

[4]Note that while in some cases these resources have already been described, we feel the need to provide a refresher, which will also serve to point to specific characteristics particularly important in these experiments.

SensEmbed, the sense-based vector representation we use in our experiment. In the following we provide further details about (1) and (2), as (3) has already been covered earlier in this dissertation (Section 3.2).

4.2.2 Training Data

Wikidata is a semantic database that is both human and machine-readable, and which includes information stemming not only from direct input from Wikimedia editors, but also facts ported from the Google project Freebase (cf. Section 1.2.1.3). Specifically, our initial training set $\mathcal{W}$ consists of the hypernym branch of Wikidata, specifically the version included in BabelNet. Note that we collapse under hypernym all Wikidata relations under the *instance-of* relation, which encodes pairs such as {(*Barack Obama*, *human*)}, {(*puma*, *taxon*)} or {(*Barcelona*, *municipality of Spain*), (*Barcelona*, *city*), (*Barcelona*, *tourist destination*) $\cdots$}. Each term-hypernym $\in \mathcal{W}$ is in fact a pair of BabelNet synsets, e.g. the synset for *Apple* (with the company sense), and the concept *company*

KB-UNIFY (KB-U)[5] is a knowledge-based approach, based on BabelNet, for integrating the output of different OIE systems into a single unified and disambiguated knowledge repository. For now, let us simply note that KB-U generates a KB of triples where arguments are linked to their corresponding BabelNet synsets, and relations are replaced by *relation synsets* of semantically similar OIE-derived relation patterns (see Chapter 6 for a full description of the method behind KB-U). The original experimental setup of KB-U includes NELL [Carlson et al., 2010] as one of its input resources: since NELL features its own manually-built taxonomic structure and relation type inventory (hence its own *is-a* relation type), we identified the relation synset containing NELL's *is-a*[6] and then drew from the unified KB all the corresponding triples, which we denote as $\mathcal{K}$. These triples constitute, similarly as in the previous case, a set of term-hypernym pairs automatically extracted from OIE-derived resources, with a disambiguation confidence of above 0.9 according to KB-U's scoring policy.

Prior to any preprocessing or further mapping, initially our two main training sets have the following size: $|\mathcal{W}| = 5{,}301{,}867$ and $|\mathcal{K}| = 1{,}358{,}949$.

4.2.3 TaxoEmbed Algorithm

Our approach can be summarized as follows. First, we take advantage of a clustering algorithm for allocating each BabelNet synset of the training set into a domain

[5] `http://lcl.uniroma1.it/kb-unify`

[6] represented by the relation `generalizations`.

cluster C (Section 4.2.3.1). Then, we expand the training set by exploiting the different lexicalizations available for each BabelNet synset (Section 4.2.3.2). Finally, we learn a cluster-wise linear projection (a *hypernym transformation matrix*) over all pairs (term-hypernym) of the expanded training set (Section 4.2.3.3).

4.2.3.1 Domain Clustering

[Fu et al., 2014] induced semantic clusters via k-means in a HD task for Chinese, where k was tuned on a development set. Instead, we aim at learning a function sensitive to a predefined knowledge domain, under the assumption that vectors clustered with this criterion are likely to exhibit similar semantic properties (e.g. similarity). First, we allocate each synset into its most representative domain, which is achieved by exploiting the set of thirty four domains available in the Wikipedia featured articles page[7]. `Warfare`, `transport`, or `music` are some of these domains. In the Wikipedia featured articles page each domain is composed of 128 Wikipedia pages on average. Then, in order to expand the set of concepts associated with each domain, we leverage NASARI[8] [Camacho-Collados et al., 2015, Camacho-Collados et al., 2016], a distributional approach that has been used to construct explicit vector representations of BabelNet synsets. In vector space modeling jargon, explicit means that each dimension is interpretable, i.e. it is associated with either words or BabelNet synsets. These interpertable dimensions come from leveraging both corpus-based statistics from Wikipedia as well as knowledge from WordNet. Thus, domains are built via building a lexical vector for each Wikipedia domain by *concatenating all Wikipedia pages representing the given domain* into a single text. Finally, given a BabelNet synset b, we calculate the similarity between its corresponding NASARI lexical vector and all the domain vectors, selecting the domain leading to the highest similarity score:

$$\hat{d}(b) = \max_{d \in D} WO(\vec{d}, \vec{b}) \tag{4.1}$$

where D is the set of all thirty-three domains, $\vec{d}$ is the vector of the domain $d \in D$, $\vec{b}$ is the vector of the BabelNet synset b, and WO refers to the *Weighted Overlap* comparison measure [Pilehvar et al., 2013], which is defined as follows:

$$WO(\vec{v_1}, \vec{v_2}) = \sqrt{\frac{\sum_{w \in O} \left(rank_{w,\vec{v_1}} + rank_{w,\vec{v_2}}\right)^{-1}}{\sum_{i=1}^{|O|}(2i)^{-1}}} \tag{4.2}$$

where $rank_{w,\vec{v_i}}$ is the rank of the word w in the vector $\vec{v_i}$ according to its weight, and O is the set of overlapping words between the two vectors. In order to have

[7] `https://en.wikipedia.org/wiki/Wikipedia:Featured_articles`
[8] `http://lcl.uniroma1.it/nasari`

a highly reliable set of domain labels, those synsets whose maximum similarity score is below a certain threshold are not annotated with any domain. We fixed the threshold to 0.35, which provided a fine balance between precision and recall in our development set. By following this approach almost 2 million synsets are labelled with a domain. See [Camacho-Collados et al., 2016] for an in-depth evaluation of the NASARI vectors.

4.2.3.2 Training Data Expansion

Prior to training our model, we benefit from the fact that a given BabelNet synset may be associated with a fixed number of senses, i.e. different ways of referring to the same concept, to expand our set of training pairs[9]. For instance, the synset b associated with the concept *music_album* is represented by the set of lexicalizations $\mathcal{L}_b = \{$album, music_album ... album_project$\}$. We take advantage of this synset representation to expand each term-hypernym synset pair. For each term-hypernym pair, both concepts are expanded to their given lexicalizations and thus, each synset pair term-hypernym in the training data is expanded to a set of $|\mathcal{L}_t|.|\mathcal{L}_h|$ sense training pairs.

This expansion step results in much larger sets $\mathcal{W}^*$ and $\mathcal{K}^*$, where $|\mathcal{W}^*| =$ 18,291,330 and $|\mathcal{K}^*| =$ 15,362,268. Specifically, they are 3 and 11 times bigger than the original training sets described in Section 4.2.2. These numbers are also higher than those reported in recent approaches for hypernym detection, which exploited Chinese semantic thesauri along with manual validation of hypernym pairs [Fu et al., 2014] (obtaining a total of 1,391 instances), or pairs from knowledge resources such as Wikidata, Yago, WordNet and DBpedia [Shwartz et al., 2016], where the maximum reported split for training data (70%) amounted to 49,475 pairs.

4.2.3.3 Learning a Hypernym Discovery Matrix

The gist of our approach lies on the property of current semantic vector space models to capture relations between vectors, in our case hypernymy. This can be found even in disjoint spaces, where this property has been exploited for machine translation [Mikolov et al., 2013b] or language normalization [Tan et al., 2015]. For our purposes, however, instead of learning a global linear transformation function in two spaces over a broad relation like hypernymy, we learn a function sensitive to a given domain of knowledge. Thus, our training data becomes restricted

[9]However, in these cases data is less prone to noise as it is in its majority derived only from manual efforts (with the exception of Yago's automatic mapping between WordNet synsets and Wikipedia cateegories).

to those term-hypernym BabelNet sense pairs $\left(x^d, y^d\right) \in C_d \times C_d$, where C_d is the cluster of BabelNet synsets labelled with the domain d.

For each domain-wise expanded training set T^d, we construct a hyponym matrix $\mathbf{X}^d = [\vec{x}_1^d \dots \vec{x}_n^d]$ and a hypernym matrix $\mathbf{Y}^d = [\vec{y}_i^d \dots \vec{y}_n^d]$, which are composed by the corresponding SENSEMBED vectors of the training pairs $\left(x_i^d, y_i^d\right) \in C_d \times C_d, 0 \leq i \leq n$.

Under the intuition that there exists a matrix Ψ so that $\vec{y}^d = \Psi\vec{x}^d$, we learn a transformation matrix for each domain cluster C_d by minimizing:

$$\min_{\Psi^C} \sum_{i=1}^{|T^d|} ||\Psi^C \vec{x}_i^d - \vec{y}_i^d||^2 \tag{4.3}$$

The resulting matrix Ψ^C is a Moore-Penrose pseudoinverse [Penrose, 1956] of $\mathbf{X}$, obtained by using its singular-value decomposition and including all its "large" values[10]. Then, for any unseen term x^d, we obtain a ranked list of the most likely hypernyms of its lexicalization vectors $\vec{x_j}^d$, using as measure cosine similarity:

$$\text{argmax}_{\vec{v} \in \mathcal{S}} \frac{\vec{v} \cdot \Psi^C \vec{x_j}^d}{||\vec{v}||||\Psi^C \vec{x_j}^d||} \tag{4.4}$$

At this point, we have associated with each sense vector a ranked list of candidate hypernym vectors. However, in the (frequent) cases in which one synset has more than one lexicalization, we condense the results into one single list of candidates, which we achieve with a simple ranking function $\lambda(\cdot)$, which we compute as $\lambda(\vec{v}) = \frac{cos(\vec{v}, \Psi^C \vec{x}^d)}{rank(\vec{v})}$, where $rank(\vec{v})$ is the rank of $\vec{v}$ according to its cosine similarity with $\Psi^C \vec{x}^d$. We adopt this policy to have an additional factor that rewards candidate hypernyms with low cosine but which are nevertheless the best candidates found by the system in the vector space (not necessarily found nearby the product of the hyponym's vector and Ψ).

The above operations allow us to cast the hypernym discovery task as a ranking problem. This is also particularly interesting to enable a flexible evaluation framework where we can combine highly demanding metrics for the quality of the candidate given at a certain rank, as well as other measures which consider the rank of the first valid retrieved candidate.

4.2.4 Evaluation

The performance of TAXOEMBED is evaluated by conducting several experiments, both automatic and manual. Specifically, we assess its ability to return valid hypernyms for a given unseen term with a held-out evaluation dataset of

[10]The cutoff for defining *large* values is set to *largest_singular_value*$\times 1e-15$.

250 Wikidata term-hypernym pairs (Section 4.2.4.1). In addition, we assess the extent to which TAXOEMBED is able to correctly identify hypernyms *outside of Wikidata* (Section 4.2.4.2).

4.2.4.1 Experiment 1: Automatic Evaluation

For each domain, we retain 5k, 10k, 15k, 20k and 25k Wikidata term-hypernym training pairs for different experiments, and evaluate on 250 test pairs for each of the 10 domains. Moreover, we aim at improving TAXOEMBED by including 1k and 25k extra OIE-derived training pairs per domain (generating two more systems, namely 25k+K^d_{1k} and 25k+K^d_{25k}). These OIE-derived instances are those contained in KB-U (see Section 4.2.2). Moreover, in order to quantify the empirically grounded intuition of the need to train a cluster-wise transformation matrix [Fu et al., 2014], we also introduce an additional configuration at 25k (25k+K^r_{50k}), where we include 50k additional pairs randomly drawn from KB-U, and two more settings with only random pairs coming from Wikidata (100k$^r_{wd}$) and KB-U (100k+$^r_{kbu}$).

We also include a distributional supervised baseline[11] based on word analogies [Mikolov et al., 2013a], computed as follows (denoted as Baseline). First, we calculate the difference vector of each training SENSEMBED vector pair ($\vec{x}^d$,$\vec{y}^d$) of a given domain d. Then, we average all the difference vectors of all training pairs to obtain a global vector $\vec{V}_d$ for the domain d. Finally, given a test term t we calculate the closest vector of the sum of the corresponding term vector and $\vec{V}_d$:

$$\hat{h} = \text{argmax}_{\vec{h} \in \mathcal{S}} cos(\vec{V}_d + \vec{t}, h) \tag{4.5}$$

This baseline has shown to capture different semantic relations and to improve as training data increases [Mikolov et al., 2013a].

Evaluation metrics.

We computed, for each domain and for the above configurations, the following metrics: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and R-Precision (R-P). They are defined as follows:

1. **MRR** takes into account the position of the first valid candidate in a ranked list of options. Formally,

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

[11] Using the 25k domain-filtered expanded Wikidata pairs as training set.

	art			biology			education			geography			health		
Train	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P
5k	0.12	0.12	0.12	0.63	0.63	0.59	0.00	0.00	0.00	0.08	0.07	0.07	0.08	0.08	0.07
15k	0.21	0.20	0.18	**0.84**	0.72	0.79	0.22	0.22	0.21	0.15	0.14	0.14	0.08	0.07	0.07
25k	**0.29**	0.27	**0.26**	**0.84**	**0.83**	**0.81**	0.33	0.32	0.30	**0.23**	**0.22**	**0.21**	0.09	0.09	0.08
25k+K^d_{1k}	**0.29**	**0.28**	**0.26**	**0.84**	0.80	0.79	0.32	0.29	0.27	0.22	**0.22**	**0.21**	0.09	0.09	0.08
25k+K^d_{25k}	0.26	0.24	0.22	0.70	0.63	0.56	**0.38**	**0.36**	**0.33**	0.15	0.13	0.12	0.11	0.11	0.10
25k+K^r_{50k}	0.28	0.26	0.24	0.82	0.77	0.72	0.36	0.33	0.30	0.17	0.16	0.16	**0.12**	0.11	0.10
100k$^r_{wd}$	0.00	0.00	0.00	**0.84**	0.81	0.77	0.00	0.00	0.00	0.01	0.01	0.01	0.07	0.06	0.06
100k$^r_{kbu}$	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	**0.12**	0.12	0.11
Baseline	0.13	0.12	0.10	0.58	0.57	**0.57**	0.10	0.10	0.09	0.12	0.09	0.05	0.07	**0.13**	**0.14**
	media			music			physics			transport			warfare		
Train	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P	MRR	MAP	R-P
5k	0.28	0.28	0.27	0.10	0.10	0.09	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
15k	0.14	0.13	0.12	0.08	0.07	0.07	0.36	0.35	0.34	0.25	0.23	0.21	0.01	0.01	0.01
25k	0.46	0.45	0.43	0.30	0.28	0.26	**0.41**	**0.40**	**0.38**	0.46	0.43	0.39	**0.05**	**0.05**	**0.04**
25k+K^d_{1k}	0.43	0.42	0.41	**0.32**	**0.30**	**0.28**	0.39	0.38	0.37	0.47	0.44	0.40	0.04	0.04	0.01
25k+K^d_{25k}	0.52	**0.51**	0.49	0.26	0.25	0.23	0.37	0.36	0.34	0.48	0.45	0.41	0.04	0.03	0.03
25k+K^r_{50k}	0.46	0.45	0.43	0.29	0.28	0.25	0.31	0.30	0.29	**0.52**	**0.49**	**0.46**	**0.05**	0.04	**0.04**
100k$^r_{wd}$	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.01
100k$^r_{kbu}$	0.08	0.07	0.07	0.01	0.01	0.00	0.00	0.00	0.00	0.10	0.10	0.10	0.00	0.00	0.00
Baseline	**0.57**	0.43	**0.52**	0.03	0.03	0.03	0.05	0.04	0.04	0.29	0.25	0.21	0.04	0.04	**0.04**

Table 4.5: Overview of the performance of TAXOEMBED using different training data samples.

where Q is a sample of experiment runs and $rank_i$ refers to the rank position of the *first* relevant outcome for the *i*th run. For our task, this is probably the most important metric, as it reveals, if we were only to select one valid hypernym for a given term, how often this first valid hypernym would be provided in the first positions of the returned candidates.

2. **MAP** is a complementary metric to MRR, which disregards the order of the correct retrieved candidates, and only takes into account whether these were retrieved within the *k* first positions in a predefined *Precision@k* measure. This is particularly useful in the case of TAXOEMBED, as for each candidate synset, we perform several queries, as many as its associated senses.

3. **R-Precision** is a sort of MAP, which only differs in the fact that it uses the number of valid hypernyms for a given synset, and uses this number as a cutoff (hence, the *R* variable). In practice, both MAP and R-Precision are strongly correlated.

We summarize the main outcome of our experiments in Table 4.5. Results suggest that the performance of TAXOEMBED increases as training data expands.

This is consistent with the findings shown in [Mikolov et al., 2013b], who showed a substantial improvement in accuracy in the machine translation task by gradually increasing the training set. Additionally, the improvement of TAXOEMBED over the baseline is consistent across most evaluation domain clusters and metrics, with domain-filtered data from KB-U contributing to the learning process in about two thirds of the evaluated configurations. These are very encouraging results considering the noisy nature of OIE systems, and that the resource we obtained from KB-U is the result of error-prone steps such as Word Sense Disambiguation and Entity Linking, as well as relation clustering.

As far as the individual domains are concerned, the `biology` domain seems to be easier to model than the rest, likely due to the fact that fauna and flora are areas where hierarchical division of species is a field of study in itself, which traces back to Aristotelian times [Mayr, 1982], and therefore has been constantly refined over the years. Also, it is notable how well the $100k^{r}_{wd}$ configuration performs on this domain. This is the only domain in which training with no semantic awareness gives good results. We argue that this is highly likely due to the fact that a vast amount of synsets are allocated into the `biology` cluster (60% of them, and up to 80% in hypernym position). This produces the so-called lexical memorization phenomenon [Levy et al., 2015], as the system memorizes prototypical biology-related hypernyms like *taxon* as valid hypernyms for many concepts. This contrasts with the lower presence of other domains, e.g. 5% in `media`, 4% in `music`, or 2% in `transport`.

Another remarkable case involves the `education` and `media` domains, which experience the highest improvement when training with KB-U (5 and 6 MRR points, respectively). One of the main sources for *is-a* relations in KB-U is NELL, which contains a vast amount of relation triples between North American academic entities (professors, sports teams, alumni, donators; as well as media celebrities). Many of these entities are missing in Wikidata, and relations among them encoded in NELL are likely to be correct because in most cases these are unambiguous entities which occur in the same communicative contexts. For example, leveraging KB-U we were able to include the pair (*university_of_north_wales*, *four_year_college*), an *is-a* relation missing in Wikidata. In fact, many high quality *is-a* pairs like this can be found in KB-U for these two domains.

We also computed P@k (number of valid hypernyms on the first k returned candidates), where k ranges from 1 to 5. Numbers are on the line of the results shown in Table 4.5 and therefore are not provided in detail. The main trend we found is showcased in Figure 4.1, which shows the illustrative example of the `transport` domain. As we can see, all values of k exhibit a similar P@k curve, with a gradual increase of performance as the training set becomes larger.

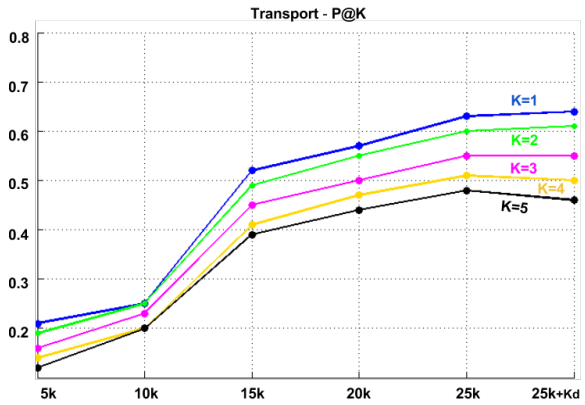


Figure 4.1: P@k scores for the `transport` domain.

4.2.4.1.1 False positives. We complement this experiment with a manual evaluation of *theoretical* false positives. Our intuition is that due to the nature of the task, some domains may be more flexible in allowing two terms to encode an *is-a* relation, while others may be more restrictive. We asked human judges to manually validate a sample of 200 *wrong pairs* from our best run in each domain, and estimated precision over them. As expected, *hard science* domains like `physics` obtain very low results (about 1% precision). In contrast, other domains like `education` (12% precision), or `transport` (16% precision), probably due to their multidisciplinary nature, allow more valid hypernyms for a given term than what is currently encoded in Wikidata.

4.2.4.2 Experiment 2: Extra-Coverage

In this experiment we evaluate the performance of TAXOEMBED on instances not included in Wikidata. For this experiment we use two configurations: the first one includes 25k domain-wise expanded training pairs (TaxE$_{25k}$), whereas the second one adds 1k pairs from KB-U (TaxE$_{25k+K^d}$). The idea is to assess whether the inclusion of additional training data, even if it is coming from potentially noisy sources, results in an improvement in coverage. We randomly extract 200 test BabelNet synsets (20 per domain) whose hypernyms are missing in Wikidata.

We compare against the taxonomy learning and Information Extraction systems Yago [Suchanek et al., 2007], WiBi [Flati et al., 2014] and DefIE [Delli Bovi et al., 2015]. Yago and WiBi are used as *upper bounds* due to the nature of their hypernymic relations (this is why their numbers are not highlighted in bold in Table 4.6). They include a great number of manually-encoded taxonomies (e.g.

	art			biology			education			geography			health		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TaxE$_{25k}$	0.45	0.45	0.45	**0.40**	**0.40**	**0.40**	**0.60**	**0.60**	**0.60**	0.35	0.35	0.35	**0.45**	**0.45**	**0.45**
TaxE$_{25k+K^d}$	0.50	**0.50**	**0.50**	**0.40**	**0.40**	**0.40**	0.55	0.55	0.55	0.35	0.35	0.35	**0.45**	**0.45**	**0.45**
DefIE	**0.63**	0.35	0.45	0.36	0.20	0.25	0.57	0.20	0.29	**0.66**	**0.40**	**0.50**	0.25	0.15	0.18
Yago	0.88	0.75	0.81	0.62	0.25	0.36	0.94	0.80	0.86	0.79	0.75	0.77	0.28	0.10	0.15
Wibi	0.70	0.70	0.70	0.58	0.50	0.54	0.94	0.80	0.86	0.75	0.75	0.75	0.66	0.50	0.57
	media			music			physics			transport			warfare		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TaxE$_{25k}$	0.10	0.10	0.10	0.45	0.45	0.45	0.15	**0.15**	0.15	0.35	**0.35**	0.35	0.25	0.25	0.25
TaxE$_{25k+K^d}$	0.10	0.10	0.10	0.40	0.40	0.40	0.15	**0.15**	0.15	0.25	0.25	0.25	0.45	**0.45**	**0.45**
DefIE	**0.81**	**0.45**	**0.58**	**0.71**	**0.50**	**0.58**	**0.42**	**0.15**	**0.22**	**0.54**	0.30	**0.38**	**0.60**	0.30	0.40
Yago	0.76	0.65	0.70	0.84	0.55	0.67	0.80	0.40	0.53	0.93	0.70	0.80	0.81	0.65	0.72
Wibi	0.90	0.90	0.90	0.89	0.85	0.87	0.68	0.55	0.61	0.87	0.70	0.77	0.66	0.50	0.57

Table 4.6: Precision, recall and F-Measure between TAXOEMBED, two taxonomy learning systems (Yago and WiBi), and a pattern-based approach that performs hypernym extraction (DefIE).

exploiting WordNet and Wikipedia categories). Yago derives its taxonomic relations from an automatic mapping between WordNet and Wikipedia categories. WiBi, on the other hand, exploits, among a number of different Wikipedia-specific heuristics, categories and the syntactic structure of the introductory sentence of Wikipedia pages. Finally, DefIE is an automaic OIE system relying on the syntactic structure of pre-disambiguated definitions[12]. Three annotators manually evaluated the validity of the hypernyms extracted by each system (one per test instance).

Table 4.6 shows the results of TAXOEMBED and all comparison systems. As expected, Yago and WiBi achieve the best overall results. However, TAXOEMBED, based solely on distributional information, performed competitively in detecting new hypernyms when compared to DefIE, improving its recall in most domains, and even surpassing Yago in technical areas like `biology` or `health`. However, our model does not perform particularly well on `media` and `physics`. In most domains our model is able to discover novel hypernym relations that are not captured by any other system (e.g. *therapy* for *radiation treatment planning* in the `health` domain or *decoration* for *molding* in the `art` domain).

In fact, the overlap between our approach and the remaining systems is actually quite small (on average less than 25% with all of them on the Extra-Coverage experiment). This is mainly due to the fact that TAXOEMBED only exploits distributional information and does not make use of predefined syntactic heuristics, suggesting that the information it provides and the rule-based comparison sys-

[12]For this experiment, we included DefIE's *is-a* relations only.

tems may be complementary. We foresee a potential avenue focused on combining a supervised distributional approach such as TAXOEMBED with syntactically-motivated systems such as Wibi or Yago. This combination of a distributional system and manual patterns was already introduced by [Shwartz et al., 2016] on the hypernym detection task with highly encouraging results.

4.2.5 Conclusion

We have presented TAXOEMBED, a supervised taxonomy learning framework exploiting the property that was observed in [Fu et al., 2014], namely that there exists, for a given domain-specific terminology, a shared linear projection among term-hypernym pairs. We showed how this can be used to learn a hypernym transformation matrix for discovering novel *is-a* relations, which are the backbone of lexical taxonomies. First, we allocate almost 2M BabelNet synsets into a predefined domain of knowledge. Then, we collect training data both from a manually constructed knowledge base (Wikidata), and from OIE systems. We substantially expand our initial training set by expanding both terms and hypernyms to all their available senses, and in a last step, to their corresponding disambiguated vector representations. w Evaluation shows that the general trend is that our hypernym matrix improves consistently as training data is increased as long as it comes from high quality sources (Wikidata). In addition, inclusion of OIE-derived information is more questionable, although we find that in half of the domains studied, in at least one of the metrics the best performing system included information from KB-U of any kind (with or without domain specificity).

Our best domain-wise configuration combines 25k training pairs from Wikidata and additional pairs from an OIE-derived KB. The domains in which the addition of the OIE-based information contributed the most are `education`, `transport` and `media`. For instance, in the case of `education`, this may be due to the over representation of the North American educational system in IE systems like NELL. We accompany this quantitative evaluation with manual assessment of precision of false positives, and an analysis of the potential coverage comparing it with knowledge taxonomies like Yago or WiBi, and with DefIE, a *quasi*-OIE system.

Chapter 5

TAXONOMY LEARNING

Previous chapters have focused on the identification of definitions from corpora, hypernym extraction from definitions, or hypernym discovery from embeddings spaces. In this chapter we describe a novel taxonomy learning system called EXTASEM!. Our algorithm takes as input a domain terminology (e.g. a list of terms in the Food domain), and returns an extended version of this terminology, with many concepts linked with high confidence to a reference sense inventory, and fully taxonomized, in the form of a directed acyclic graph where edges encode hypernymic relations.

5.1 ExTaSem! Extending, Taxonomizing and Semantifying Domain Terminologies

As explained in Chapter 2, previous methods for inducing taxonomic relations can be (broadly) classified into linguistic or statistic. Linguistic methods are those that, extending Hearst's patterns [Hearst, 1992], exploit linguistic evidence for unveiling hypernym relations [Kozareva and Hovy, 2010, Navigli et al., 2011, Flati et al., 2014, Luu Anh et al., 2014]. Other approaches are based purely on statistical evidence and graph-based measures [Fountain and Lapata, 2012, Alfarone and Davis, 2015]. However, none of these approaches addressed explicitly the problem of ambiguity and semantically-motivated domain pertinence, albeit a few cases in which all this was tackled tangentially [Kozareva and Hovy, 2010, Velardi et al., 2013]. EXTASEM! is designed to bridge the gap between relation extraction and graph construction, on one hand, and domain pertinence on the other. Starting from a list of domain terms, EXTASEM! induces a full-fledged taxonomy by leveraging a large semantic network, from which *high quality knowledge* in the form of textual definitions is retrieved for each domain. Then, (hyponym, hypernym) pairs are extracted via a CRF-based sequential classifier. In addition, a state-

of-the-art vector space representation of individual word senses is exploited for constructing a domain taxonomy only made up of semantically pertinent edges[1]. Finally, our approach does not require a step for graph pruning or trimming, a must in some of the systems mentioned above.

In terms of taxonomy evaluation, EXTASEM! is able to reliably reconstruct gold standard taxonomies of interdisciplinary domains such as `science`, `terrorism` or `artificial intelligence`, as well as more specific ones like `food` or `equipment`. In addition, it has the capacity to extend and semantify an input taxonomy, i.e. increase its size and link many of its nodes to a reference sense inventory.

In what follows we describe the pipeline of EXTASEM! and the resources enabling its semantic properties. Let $\mathbf{I}_\varphi$ be a set of terms in domain φ, where: $\varphi \in \{\text{Food}, \text{Equipment}, \text{Science}, \text{Chemical}, \text{AI}, \text{Terrorism}\}$[2], and let $\mathbf{T}_\varphi$ be the final domain taxonomy, which can be described as a directed acyclic graph. The root node of the taxonomy corresponds with a generic umbrella term of the target domain φ. In the following we describe how EXTASEM! learns $\mathbf{T}_\varphi$ from $\mathbf{I}_\varphi$.

5.1.1 Domain Definition Harvesting

Following previous work in Definition Extraction [Saggion, 2004, Navigli and Velardi, 2010], EXTASEM! extracts candidate hypernyms of terms by mining textual definitions retrieved from reliable knowledge sources. In this way we can focus on the semantic coherence and the completeness of the taxonomy we build with respect to both the addition of novel terms and edges and the evaluation of their quality against reference sense inventories. Moreover, by gathering definitions from reliable knowledge sources we reduce the risk of semantic drift in our taxonomy and the need of costly and often imprecise pruning approaches. These approaches are usually adopted when evidence is harvested from non-curated data like the web [Kozareva and Hovy, 2010] or the output of Open Information Extraction (OIE) systems [Alfarone and Davis, 2015].

The first component of the EXTASEM! pipeline is the Domain Definition Harvesting (DDH) module. Given a domain terminology $\mathbf{I}_\varphi$, the DDH module collects a corpus of domain definition sentences D_φ retrieved from BABELNET that constitutes our *global definition repository*.

The DDH module consists of two sequential phases (see Figs. 5.1 and 5.2): the *Domain Pertinence Scorer* of Wiki-Categories (DDH-CatDPScorer) and the *Domain Definitions Gathering* (DDH-DefGath). The DDH-CatDPScorer generates a list of Wikipedia Categories, each one characterized by a score that quantifies

[1] Taxonomies available at http://bitbucket.com/luisespinosa/extasem

[2] See Section 5.1.5 for the motivation behind the choice of these domains.

its pertinence to the domain of the input terminology $\mathbf{I}_\varphi$. Then, the DDH-DefGath further prunes this list of Wikipedia Categories with respect to their domain relevance and semantic coherence and, then, exploits the pruned Category list to populate the corpus of domain definition sentences (D_φ). Hereafter we describe each phase in detail.

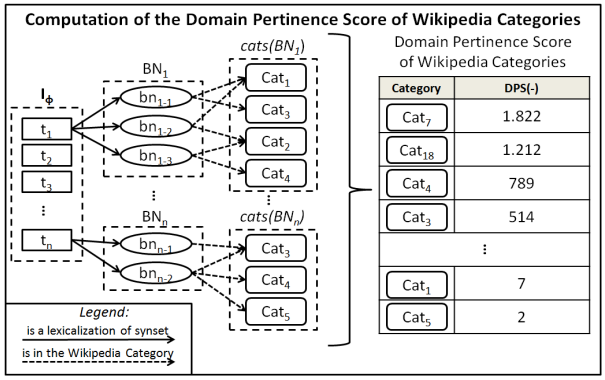


Figure 5.1: DDH: DPS computation phase.

DDH-CatDPScorer (see Fig. 5.1): for each term τ belonging to the input domain terminology $\mathbf{I}_\varphi$, we collect the BABELNET synsets BN_τ that include the term τ as one of their lexicalizations. Then, exploiting the Wikipedia Bitaxonomy [Flati et al., 2014] integrated in BABELNET, for each set of BABELNET synsets BN_τ, we compute $cats(BN_\tau)$, i.e. the set of Wikipedia Categories that include at least one BABELNET synset in BN_τ. We compute the Domain Pertinence Score (DPS) of each Wikipedia Category CAT$_n$:

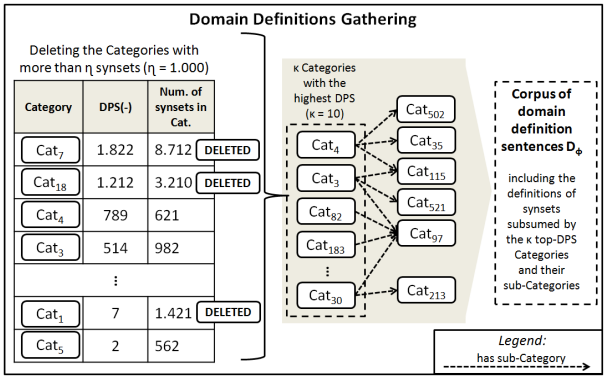


Figure 5.2: DDH: Domain Definitions Gathering phase.

$$DPS(\text{CAT}_n) = \sum_{\tau \in \mathbf{I}_\varphi} \begin{cases} 1 & \text{if CAT}_n \in cats(BN_\tau) \\ 0 & \text{if CAT}_n \notin cats(BN_\tau) \end{cases}$$

The DPS of each Category is equal to the number of terms τ that represent one of the lexicalizations of a BABELNET synset included in the same Category (thus belonging to BN_τ). We rely on the intuition that the greater the DPS of a Category is, the higher is the relevance of that Category to the domain of the terminology $\mathbf{I}_\varphi$. The output of the DDH-CatDPScorer phase is the list of all Wikipedia Categories that have a DPS greater than zero.

DDH-DefGath (see Fig. 5.2): from the list of Wikipedia Categories that have a DPS greater than zero, we filter out those that include more than η synsets. We applied this procedure since we noted that often Wikipedia Categories that include large amounts of synsets are one-size-fits-all repositories. These Categories may not be relevant to characterize our domain of interest since they often group huge amounts of semantically heterogeneous synsets, thus showing low semantic coherence. Examples of these Categories are: *Living People* or *English Language Films*. As a consequence of the analysis of several cases, we empirically set the Category exclusion threshold η to 1000. From the filtered list of Categories, we select the κ Categories with the highest DPS (top-κ). From each BABELNET synset that is included in a top-κ Category or one of its sub-Categories, we collect all the full-sentence definitions in BABELNET (which includes Wikipedia, WikiData, OmegaWiki, WordNet, and Wiktionary definitions). In all the experiments we report, we set κ equal to 10. The set of full-sentence definitions we collect constitutes our corpus of domain definition sentences D_φ.

In short, the DDH module generates a set of Wikipedia Categories that are pertinent to the domain of the input terminology $\mathbf{I}_\varphi$. The corpus of domain definition sentences D_φ consists of all the definitions of the synsets included in these Categories and in their sub-Categories. For instance, if our input domain terminology is about `food` and includes the term *orange*, we will retrieve from BabelNet all its senses (i.e. BabelNet synsets). Possible senses of *orange* are: '*the color of carrots, pumpkins and apricots*' and '*the fruit of citrus*'. Orange as a color is included in the Wikipedia Category **Optical spectrum**, while orange as a fruit belongs to the Category **Fruit**. Since in the input terminology there are many other names of fruits that are lexicalizations of synsets included in the category **Fruit**, the DPS of this Category will be considerably higher than the DPS of the Category **Optical spectrum**. In this way we can select the synset definitions of the most domain-pertinent Categories (with highest DPS values) to populate our corpus of domain definition sentences D_φ.

5.1.2 Hypernym Extraction

A core component of our pipeline is the Hypernym Extraction (HE) module. Given a textual definition $d_\tau \in D_\varphi$, we obtain the *longest and most specific* hypernym of τ. Then, exploiting the syntactic structure of each multiword hypernym, we propose a hypernym-decomposition step for increasing the depth of the graph, which is preferred in taxonomy learning [Navigli et al., 2011]. Method-wise, we cast this HE subtask as a sequence-to-sequence classification problem, where we train a model similarly as in Section 3.4 (same feature set, and also using CRF as learning algorithm).

The model (which is trained to sequentially predict whether a word in a definition constitutes the definiendum's hypernym) is applied to D_φ to extract a set $\mathbf{H}_\varphi$ of (hyponym, hypernym) pairs. At this stage, τ may be associated with more than one hypernym, as we may extract several candidates from different definition sources. For example, for τ = TRUFFLE, extracted candidates are CONFECTION, GANACHE CENTER, and CHOCOLATE CANDY. Note that GANACHE CENTER is a wrong hypernym for TRUFFLE, and will eventually be pruned out.

5.1.3 Fine-Graining Hyponym - Hypernym Pairs

We propose a *hypernym decomposition* heuristic over the syntactic dependencies in a definition d_τ as follows: (1) Extract from the sentence the dependency subtree rooted at the head of the hypernym candidate; (2) Remove one modifier at a time until the hypernym candidate consists only of one token. A syntactic constraint is introduced to retain only relevant modifiers, i.e. only nouns, adjectives and verbs are kept. This procedure outputs a finer-grained set of relations, denoted as $\mathbf{H}'_\varphi$. For example, JAPANESE SNACK FOOD $\mapsto$ {JAPANESE SNACK FOOD, SNACK FOOD, FOOD}.[3]

We take advantage of the fact that in many cases the extracted hypernyms are multi-word terms and propose a *hypernym decomposition* heuristic that leverages the syntactic dependencies in d_τ. The main idea is, for each hypernym $hyper \in d_\tau$, to get the subtree of the dependency parsed sentence rooted at $hyper$, and construct further hypernym candidates derived from $hyper$, pivoting over the head of the hypernym. Our syntactic constraint ξ states that syntactic dependants of the hypernym's head are valid modifiers only if their part-of-speech is either JJ, NN* or V*, and if their syntactic function is *nmod* (noun modifier). The procedure (summarized in Algorithm 2) outputs a finer-grained set of relations, denoted as H'.

[3] A manual analysis over a random sample of 100 edges in the AI domain showed that compositionality failed in less than 6% of the cases.

Algorithm 2 Hypernym Decomposition

Require: $\mathbf{H}_{\varphi}$ // (hyponym, hypernym) pairs.
Output: $\mathbf{H}'_{\varphi}$ // Finer-grained (hyponym, hypernym) pairs
$//pos =$ A word's position in a sentence
$//\aleph_x =$ Returns the syntactic head of phrase x
$//dP(x, y) =$ Obtains the dependency parse rooted as term x in sentence y
$\mathbf{H}'_{\varphi} = \mathbf{H}_{\varphi}$
for $(\tau, hyp)_{d_\tau} \in \mathbf{H}_{\varphi}$ **do**
  $h' = hyp$
  while $|h'| >= 2$: **do**
    $s =$ leftmost successor of h' in $dP\left(h', d_\tau\right)$
    if $pos\left(s\right) < pos\left(\aleph_{dP(h', d_\tau)}\right)$ **then**
      if $\xi = True$ **then**
        $\mathbf{H}' = \mathbf{H}' \cup \{(h', h' - s)\}$
      end if
    end if
    $h' = h' - s$
  end while
end for
return $\mathbf{H}'$

After the hypernym decomposition step, we construct a set of *candidate paths* P^{φ} from $\mathbf{H}'_{\varphi}$. A candidate path $p^{\varphi}_{\tau} \in P^{\varphi}$ is defined as a path from a term node τ to the root node φ, and includes as intermediate nodes those created during the syntactic decomposition step. From our previous example, {JAPANESE SNACK FOOD, SNACK FOOD, FOOD} $\mapsto$ {JAPANESE SNACK FOOD $\rightarrow$ SNACK FOOD $\rightarrow$ FOOD}[4]. In the following section, we explain how EXTASEM! constructs a domain-pertinent taxonomy from P^{φ}.

5.1.4 Path Weighting and Taxonomy Induction

We expect *good paths* to be relevant to the domain. In previous work, this has been approached in a plethora of ways. For instance, by leveraging syntactic evidence, by capturing in domain corpora hyponym, hypernym pairs related by a predefined syntactic relation [Luu Anh et al., 2014]. Relevance of candidate (sets of) hypernymic relations has also been computed by combining the aforementioned syntactic evidence in triples extracted by OIE systems [Alfarone and Davis, 2015], or by "forcing" a kind of domain pertinence by querying the web with a term and

[4]Henceforth, we denote edges as *term→hypernym*.

a generic hypernym (which inherently is performing an *a priori* disambiguation) [Kozareva and Hovy, 2010]. However, recent work in vectorial representations of semantically-enhanced items has shown state-of-the-art performance in several word similarity and word relatedness tasks [Camacho-Collados et al., 2015, Speer et al., 2016]. This suggests that these representations may be much more suitable for our semantics-intensive path weighting policy. Thus, we incorporate a module based on SENSEMBED, which operates on the back of a sense inventory S with a corresponding vector space Γ.

We model the relevance of p_τ^φ to φ (e.g. Food or Chemical) by computing its *domain pertinence*. This is given by the weighting function $w(\cdot)$, computed as the cumulative semantic similarity between each node $n \in p_\tau^\varphi$ and φ. We follow an **L2S** cosine-based disambiguation strategy (cf. Chapter 3). This is aimed both at accurately disambiguating an input text-level concept with respect to a target domain, and to obtain a score for a whole candidate path, from leaf node to root concept. For instance, our aim would be to assign to n =apple the closest sense to φ so that for the node *apple*, the correct sense in the FOOD domain is that of the fruit, and not that of the company.

Then, we weigh each path as follows:

$$w(p_\tau^\varphi) = \sum_{l \in L(p_\tau^\varphi)} \cos(l, \varphi)$$

where $\cos(\cdot)$ computes the cosine similarity between two vectors, and $L(p_\tau^\varphi)$ is the set of *linkable nodes* in a path, i.e. those nodes with at least one vector representation associated with them. This yields P_W^φ, a weighted set of candidate edges. For instance, {(MIKADO→JAPANESE SNACK FOOD), (JAPANESE SNACK FOOD→SNACK FOOD), (SNACK FOOD→FOOD)}$_{w=0.3}$.

Finally, the taxonomy induction module generates a full-fledged semantified taxonomy $\mathbf{T}_\varphi$ with many intermediate nodes which were not present in $\mathbf{I}_\varphi$, as well as a large number of novel non-redundant edges. This last step is described in Algorithm 3. We empirically set a threshold θ to .135, and apply it over all domains.

5.1.5 Evaluation

Evaluating the quality of lexical taxonomies is an extremely difficult task, even for humans [Kozareva et al., 2009]. This is mainly because there is not a single way to model a domain of interest [Velardi et al., 2013], and even a comparison against a gold standard may not reflect the true quality of a taxonomy, as gold standard taxonomies are not complete. This is especially relevant in multidisciplinary and evolving domains such as Science [Bordea et al., 2015]. Thus, we

Algorithm 3 Taxonomy Induction

Require: Threshold θ, weighted paths P_W^φ
Output: Disambiguated domain taxonomy $\mathbf{T}_\varphi$

```
A(term) = {ancestors of term}
T_φ = ∅
for ρ_τ^φ ∈ P_W^φ do
    if w(ρ_τ^φ) > θ then
        for (term, hyp) ∈ ρ_τ^φ do
            if hyp ∉ A(term) then
                T_φ = T_φ ∪ {term → hyp}
            end if
        end for
    end if
end for
return T_φ
```

evaluated EXTASEM! from two different standpoints, namely: (1) Reconstructing a gold-standard taxonomy; and (2) Taxonomy quality and semantic content, where we look at structural features like number of edges or graph depth. We used the following data for our experiments:

1. **TexEval 2015**: We evaluated on Semeval-2015 Task 17 (TexEval) domains (cf. Chapter 2): Science (*sci.*), Food (*food*), Equipment (*equip.*) and Chemical (*chem.*). For each domain, two terminologies and their corresponding gold standard taxonomies were available. Such gold standards came from both domain-specific sources (e.g. for *chem.*, the ChEBI taxonomy[5]) and the WordNet subgraph rooted at the domain concept (e.g. the WordNet subtree rooted at *chemical* in the case of *chem.*). Note that since WordNet is integrated in BABELNET, evaluation over WordNet gold standard would artificially favour our approach, so we decided to only evaluate on the domain-specific taxonomies. We compared our results against the taxonomies produced by task participants.

2. **Additional *multidisciplinary* domains**: We assessed the EXTASEM! taxonomies in the domains of Artificial Intelligence (*AI*) [Velardi et al., 2013] and Terrorism (*terr.*) [Luu Anh et al., 2014]. For the same fairness reason as above, we avoid domains covered in previous work where the gold standard comes from WordNet, such as Animals, Plants and Vehicles, used in [Velardi et al., 2013, Kozareva and Hovy, 2010, Alfarone and Davis, 2015].

[5] https://www.ebi.ac.uk/chebi/

	Food			Science			Chem.			Equip.		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
INRIASAC	.18	.51	.27	.17	.44	.25	.08	.09	**.09**	.26	.49	.34
LT3	.28	.29	.29	.40	.38	**.39**	-	-	-	.70	.32	.44
ntnu	.07	.05	.06	.05	.04	.04	.02	.002	.001	.01	.006	.009
QASSIT	.06	.06	.06	.20	.22	.21	-	-	-	.24	.24	.24
TALN-UPF	.03	.03	.03	.07	.25	.11	-	-	-	.14	.15	.15
USAARWLV	.15	.26	.20	.18	.37	.24	.07	.09	.08	.41	.36	.39
EXTASEM!	.28	.66	**.39**	.27	.32	.29	.05	.02	.03	.51	.56	**.54**

Table 5.1: Comparative edge-level Precision, Recall and F-measure scores. Refer to [Bordea et al., 2015] for a description of each of the systems listed.

5.1.5.1 Reconstructing a Gold Standard

Experiment 1 - TexEval 2015

For this experiment, we introduced a modification to the pipeline. We complemented DDH with a web-search stage. We queried the Bing[6] search engine with terms whose definitions were not found in BabelNet, and from a concatenation of web pages web_τ, we kept as candidate hypernyms all the terms from the initial terminology found in web_τ. Then, applying the disambiguation procedure described in Section 5.1.4, we kept at most the best three candidates (i.e. those who were semantically closest to τ) for each term, and added one edge between each best candidate and τ[7].

The taxonomies generated by EXTASEM! are compared against participant systems in TexEval. The evaluation criterion in this experiment is to assess how well systems can replicate a gold standard in any of the four evaluated domains. This is done via Precision, Recall and F-Score at edge level.

The results of this experiment suggest show that EXTASEM! ranks first in half of the domains (Table 5.1), and second and third in Science and Chemical respectively. As can be appreciated, if we average the results of all the systems

[6]https://datamarket.azure.com/dataset/bing/search

[7]We introduce this variation with respect to the original EXTASEM! pipeline to evaluate a *precision-oriented* version of the system, where at most three candidate hypernyms are retrieved for each term. If we were to include the whole pipeline (with the terminology expansion module), our results would look artificially low, as we would have thousands of edges for each domain with nodes initially absent in the original terminology.

participating in this experiment across the four domains, our approach ranks first (F=0.31, the second best system being LT3 with F=0.28).

5.1.5.1.1 Experiment 2 - Evaluation of a Subsample The Cumulative Fowlkes & Mallows Measure (CFM) [Velardi et al., 2013] has become a *de-facto* standard for evaluating lexical taxonomies against ground truth. It was introduced as a re-work of the original Fowlkes&Mallows measure [Fowlkes and Mallows, 1983], and was used as one of the evaluation criteria in TexEval 2015. This measure assigns a score between 0 and 1 according to how well a system clusters similar nodes at different cut levels.

Previous approaches evaluated the capacity of their systems to replicate a lexical hierarchical structure given a terminology, mirroring their output against WordNet in most cases [Navigli et al., 2011, Velardi et al., 2013, Fountain and Lapata, 2012, Kozareva and Hovy, 2010, Luu Anh et al., 2014]. In this experiment, we took advantage of extensive human input, and asked domain experts to reconstruct a sample of 100 concepts from taxonomies produced by EXTASEM!. The reason for having a sample of 100 terms is that it is a compact enough sample to avoid the "messy organization" previous authors have reported [Velardi et al., 2013, Kozareva and Hovy, 2010], while being a larger sample than experiments performed similarly, e.g. in [Fountain and Lapata, 2012], where the terminologies given to human judges were only of 12 terms.

For each 100-term sample, a domain expert was asked to order hierarchically as many concepts as possible, but was allowed to leave out any node if it was considered noisy. We used these expert taxonomies as gold standard. We also evaluated a baseline method based on substring inclusion consisting in creating a hyponym→hypernym pair between two terms if one is prefix or suffix substring of the other. Table 5.2 shows results in terms of edge overlap (RECALL) and CFM. The agreement between EXTASEM! and human experts was high, performing much better than the baseline.

	Baseline		**EXTASEM!**	
	RECALL	**CFM**	**RECALL**	**CFM**
Food	0.49	0.02	**0.79**	**0.50**
Science	0.22	0.01	**0.57**	**0.64**
Equip.	0.43	0.01	**0.77**	**0.50**
Terr.	0.54	0.07	**0.69**	**0.27**
AI	0.51	0.02	**0.77**	**0.49**

Table 5.2: CFM for domain 100-term gold standard comparison.

5.1.5.2 Taxonomy Quality

Experiment 1 - Structural Evaluation

According to [Bordea et al., 2015], the purpose of taxonomy structural evaluation is to: (1) Quantify its size in terms of nodes and edges; (2) Assess whether all components are connected; and (3) Quantify semantic richness in terms of proportion of intermediate nodes versus leaf nodes (which are considered less important). Thus, we compare automatic taxonomies produced by EXTASEM! with gold standard taxonomies from TexEval 2015 (TEXE) in all domains, as well as automatic taxonomies produced in Artificial Intelligence (*AI*) [Velardi et al., 2013] and Terrorism (*terr.*) [Luu Anh et al., 2014]. We evaluated over these parameters: Number of nodes (NODES); number of edges (EDGES); number of connected components (C.C); number of intermediate nodes, i.e. those which are neither root or leaf nodes (I.N); maximum depth of the taxonomy (MD); and average depth (AD).

EXTASEM! produces bigger taxonomies with more intermediate nodes in three out of four TexEval domains. This does not affect negatively the structural properties of these taxonomies, as they also improve in terms of MD and are only slightly behind in AD in some domains. The case of the Science domain is remarkable, where the automatic EXTASEM! taxonomy shows greater AD than the gold standard. The one domain that poses most difficulties for our approach is Chemical due to the low coverage this domain has in BABELNET.

As for comparison against automatic taxonomies, while AD and MD are lower than Velardi et al.'s *OntoLearn Reloaded*, note that in their approach many upper-level (not domain-specific) nodes are introduced, which are described as "general enough to fit most domains"[8]. Finally, our evaluation suggests that the Terrorism taxonomy in [Luu Anh et al., 2014] does not have all the components connected. We therefore report statistics on its biggest connected subgraph. Additionally, since it was not constructed on the back of an umbrella root node, we do not report numbers on depth. This reflects the complexity of the taxonomy learning task, where perfectly valid domain-specific taxonomies may be shaped as trees or as directed acyclic graphs, with or without root nodes. Full domain-wise details are provided in Table 5.3.

5.1.5.2.1 Experiment 2 - Hypernym Extraction We considered WIBI as our main competitor in the task of hypernym extraction due to the similarities in terms of (hyponym, hypernym) extraction from a definition setting.

For each domain, two experts were presented with 100 randomly sampled terms and two possible hypernyms, the hypernym selected by EXTASEM! and

[8] Some of these nodes are *abstraction*, *entity*, *event* or *act*.

	Food		Science		Equipment		Chemical		Terrorism		Art. Intel.	
	TexE	**ExTaSem!**	**TexE**	**ExTaSem!**	**TexE**	**ExTaSem!**	**TexE.**	**ExTaSem!**	**LA et al.**	**ExTaSem!**	**Vel. et al.**	**ExTaSem!**
NODES	1556	3647	452	2124	612	2062	17584	4932	123	510	2388	1556
EDGES	1587	3930	465	2243	615	2214	24817	5355	243	548	2386	1610
C.C	1556	3647	452	2124	612	2062	17584	4932	N.A	510	2386	1556
I.N	69	1980	53	611	57	995	3349	2051	N.A	292	747	730
MD	6	9	5	8	6	9	18	8	N.A	7	13	7
AD	3.8	3.6	3.7	3.9	3.6	3.5	9	3.7	N.A	3.4	6.7	3.5

Table 5.3: Taxonomy structure results. Comparison between ExTaSem! and TexEval results (TexE), as well as [Luu Anh et al., 2014] (LA et al.) and [Velardi et al., 2013] (Vel. et al.).

	FOOD		SCIENCE		EQUIPMENT	
	Valid	Best	Valid	Best	Valid	Best
WIBI	0.85	0.29	0.85	0.39	0.84	0.3
EXTASEM!	**0.94**	**0.91**	**0.91**	**0.83**	**0.90**	**0.83**
	CHEMICAL		**AI**		**TERRORISM**	
	Valid	Best	Valid	Best	Valid	Best
WIBI	**0.75**	0.03	0.76	0.39	**0.79**	0.24
EXTASEM!	0.64	**0.32**	**0.84**	**0.80**	0.78	**0.73**

Table 5.4: Human judgement on the quality of the hypernymic relations provided by WIBI and EXTASEM! for 6 domains.

the one from WIBI. Each pair was shuffled to prevent evaluators from guessing which could be the source. For each pair of hypernym candidates, evaluators had to decide which of the two options constituted a *valid* hypernym in the given domain. They were allowed to leave this field blank for both systems. If both the hypernyms in WIBI and EXTASEM! were valid, evaluators were asked to decide which system offered the *best* hypernym (or both if it was the same), and for this we asked them to consider the hypernym's semantic relatedness and closeness to the hyponym, as well as relevance to the domain. For example, for the hyponym CHUPA CHUPS, we would prefer LOLLIPOP over COMPANY in the `food` domain, even if strictly speaking both options would be valid. We computed inter-rater agreement with the Cohen's Kappa metric over the *valid* and *best* classes, with average results of 0.53 and 0.36.

The results in Table 5.4 suggest that in general the hypernyms extracted with our procedure are better, i.e. more appropriate to the domain and more informative, than the ones extracted from the syntactically-motivated heuristic described in [Flati et al., 2014].

5.1.6 Conclusion

This section presented and evaluated EXTASEM!, a system that constructs a domain-specific semantically rich taxonomy from an input terminology. It consists of three main modules, namely: (1) Domain Definition Harvesting, where BABELNET and WIBI are leveraged in order to obtain a significant amount of definitional evidence; (2) Hypernym Extraction and Decomposition, based on a CRF-based sequential classifier and a syntactically-motivated hypernym decomposi-

tion algorithm; and (3) Path Disambiguation and Graph Induction, on the back of SENSEMBED, a state-of-the-art vector space representation of individual word senses.

Parting ways from previous approaches in which is-a relation evidence was gathered from non curated data like the web or OIE systems, EXTASEM! explicitly tackles the semantics of each candidate (hyponym, hypernym) pair, as well as its pertinence to the target domain. Our system achieves state-of-the-art performance in reconstructing gold standard taxonomies, and is able to extend them retaining their domain relevance. We further discuss assets related to this system in Chapter 7, as well as potential avenues for future work due to this system's limitations in Chapter 8.

Chapter 6

CREATION, ENRICHMENT AND UNIFICATION OF KNOWLEDGE RESOURCES

The task of knowledge formalization can dramatically influence the construction, extension and enrichment of existing knowledge resources. In this chapter, we describe several experiments in this direction. First, we exploit the combination of semantic and syntactic information for learning a KB in the music domain entirely from scratch (Section 6.1). Then, we focus on: Making sense of weakly structured (OIE systems) and polysemous (at the text level) information, so that it becomes seamlessly integrated into one single resource (Section 6.2); enriching WordNet with collocational information, a very important component in any lexicographic resource (Section 6.3); and extending the medical terminology SnomedCT in Spanish, in Section 6.4.

6.1 MKB: Creating a Music Knowledge Base from Scratch

Our first contribution in the area of KB creation and extension focuses on a highly specific use case. The main idea is to explore whether NLP techniques can constitute the methodological core behind the creation of a KB where content is (mostly) non-textual. Specifically, we decided to investigate the music domain, as it has received little attention by the NLP community, for example, for exploiting textual data in Music Information Retrieval (MIR) and Music Recommendation systems. Let us first, however, provide the reader with the necessary context to understand the current state of music-related KBs, and what is missing. This project, a joint effort from Luis Espinosa-Anke and Sergio Oramas, is a collaboration between

the Music Technology Group and the Natural Language Processing group at Universitat Pompeu Fabra, in the context of the Music Meets NLP project, supported by the Maria de Maeztu Units of Excellence Program.

6.1.1 Background

While the number of resources available in the music domain (at any degree of structuring) is scarce, there are however some notable cases that currently constitute the best examples of music-specific KRs. On one hand, we find specialized resources containing music-related information, such as MUSICBRAINZ and DISCOGS, two manually curated Music Knowledge Bases (MKBs). They are open music encyclopedias of music metadata built collaboratively and openly available. MUSICBRAINZ, in addition, is regularly published as Linked Data by the LINKEDBRAINZ project[1]. Another type of resources which contain musical information are generic KRs with a music branch or subset, WIKIPEDIA being an outstanding example. These resources include a remarkable amount of music data, such as artist, album and song biographies, definitions of musical concepts and genres, or articles about music institutions and venues. However, their coverage is biased towards the best known artists, and towards products from Western culture. Finally, let us refer to the notable case of GROVE MUSIC ONLINE[2], a music encyclopedia containing over 60k articles written by music scholars. However, it has the drawback of not being freely open, as it runs by subscription.

Other than the aforementioned curated repositories, to the best of our knowledge, there is not a single automatically learned open MKB. A first step in this direction was taken in [Oramas et al., 2014, Sordo et al., 2015], applying Information Extraction (IE) techniques to big corpora of music related texts extracted from the web. Moreover, in [Oramas et al., 2015a], a Flamenco MKB is created by combining data from curated KBs and information extracted from blogs and websites.

Despite their scarcity, MKBs are becoming increasingly popular in MIR applications, such as artist similarity and music recommendation [Celma and Serra, 2008, Oramas et al., 2015b, Leal et al., 2012, Ostuni et al., 2015]. MKBs have also been exploited as sources of explanations in music recommender systems. According to [Celma and Herrera, 2008], giving explanations of the recommendations provides transparency to the recommendation process and increases the confidence of the user in the system. In [Passant, 2010], explanations of recommendations are created by exploiting DBPEDIA's structured information, whilst in [Sordo et al., 2015], explanations are based on an automatically learned MKB.

[1] http://linkedbrainz.org/

[2] http://www.oxfordmusiconline.com

6.1.2 Methodology

We propose a pipeline that learns a full-fledged MKB taking as input a musical text corpus (not lyrics, but rather text documents *about music*), coming from the Songfacts[3] website (see Section 6.1.3.1). This is a well suited resource both for KB learning and as a testebed for IE due to its specificity. Songfacts documents, while not being as rigid as encyclopedic or newswire text, remain well-formed, sentences make sense, and there is no need for *ad-hoc* preprocessing (as it is required in social networks, e.g. Twitter).

6.1.2.1 Notation

Our method focuses on the extraction of semantic relations between pairs of linked entities (e.g. *Born in the USA*$_{dbr}$, *Bruce Springsteen*$_{dbr}$[4]), which are in turn associated to specific entity types (e.g. *Album*, *MusicalArtist*). In our KB, a relation r is defined by the tuple $\langle \boldsymbol{e_d}, \boldsymbol{e_r}, \boldsymbol{v_d}, \boldsymbol{v_r}, \boldsymbol{p}, \boldsymbol{c} \rangle$, where $\boldsymbol{d}$ and $\boldsymbol{r}$ refer to domain and range positions, $\boldsymbol{e_d}$ and $\boldsymbol{e_r}$ to the entities involved in the relation, $\boldsymbol{v_d}$ and $\boldsymbol{v_r}$ to their associated entity types, $\boldsymbol{p}$ to a relation pattern, and $\boldsymbol{c}$ to a cluster pattern. A relation pattern is a relation label that may be used in one or several relations (e.g *was recorded by frontman*, *was recorded by singer/songwriter*). Relation patterns with similar semantic and syntactic characteristics may be grouped into cluster patterns (e.g. *was recorded by*). Moreover, we denote as $\mathcal{R}$ the set of all extracted relations included in the KB. For each $r \in \mathcal{R}$, triples of different nature can be constructed by arbitrarily combining elements in r.

- $t_{\boldsymbol{p}} : \langle \boldsymbol{e_d}, \boldsymbol{p}, \boldsymbol{e_r} \rangle$, e.g. {*Born in the USA*$_{dbr}$ *- was recorded by frontman - Bruce Springsteen*$_{dbr}$}.
- $t_{\boldsymbol{c}} : \langle \boldsymbol{e_d}, \boldsymbol{c}, \boldsymbol{e_r} \rangle$, e.g. {*Born in the USA*$_{dbr}$ *- was recorded by - Bruce Springsteen*$_{dbr}$}.
- $\tau_{\boldsymbol{p}} : \langle \boldsymbol{v_d}, \boldsymbol{p}, \boldsymbol{v_r} \rangle$, e.g. {*Album - was recorded by frontman - MusicalArtist*}.
- $\tau_{\boldsymbol{c}} : \langle \boldsymbol{v_d}, \boldsymbol{c}, \boldsymbol{v_r} \rangle$, e.g. {*Album - was recorded by - MusicalArtist*}.

Finally, different subsets of $\mathcal{R}$ may be constructed by selectively filtering all $r \in \mathcal{R}$.

- $\mathcal{R}_p = \{r_1^p, ...r_n^p\}$ All relations with a specific relation pattern p.
- $\mathcal{R}_c = \{r_1^c, ...r_n^c\}$ All relations with a specific cluster pattern c.

[3] `http://www.songfacts.com`

[4] We use the *dbr* subscript to refer to disambiguated entities linked to DBPEDIA resources.

- $\mathcal{R}_{\tau_p} = \{r_1^{\tau_p}, ...r_n^{\tau_p}\}$ All relations with a specific relation pattern, and domain and range entity types.
- $\mathcal{R}_{\tau_c} = \{r_1^{\tau_c}, ...r_n^{\tau_c}\}$ All relations with a specific cluster pattern, and domain and range entity types.

In what follows, we describe a method for acquiring new entities, types and relations, and combining them in a meaningful way for KB construction.

6.1.2.2 Morphosyntactic Processing

Our morphosyntactic preprocessing module takes as input a collection of text documents in the music domain. First, sentence splitting and tokenization is carried out thanks to the *Stanford NLP tokenizer*[5]. Next, a dependency parse tree is obtained similarly as in Chapter 3. The result is a syntactic representation for every sentence, as can be seen in Figure 6.1.

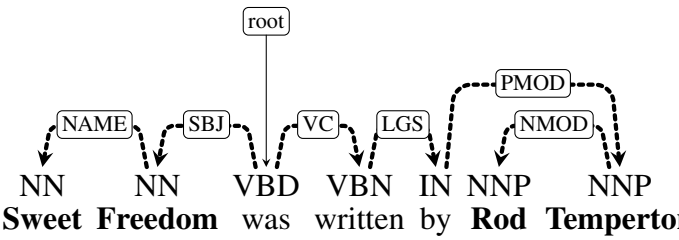


Figure 6.1: Example sentence with dependency parsing tree

6.1.2.3 Semantic Processing: Entity Linking

There is no benchmark of Entity Linking (EL) systems in the music domain [Oramas et al., 2016]. Therefore, we do not know *a priori* how well each of them works in music corpora. Musical entities may raise a plethora of challenges, derived mostly from ambiguity and polysemy. For example, an album may have the same name as the band who recorded it (e.g. *Weezer* the band and their first album). Moreover, an artist, a song or an album may have words or expressions much more common in another domain or area of knowledge (e.g. *Berlin, The Who*). Thus, the choice of the best EL algorithm or off-the-shelf tool(s) is crucial, as potential errors may propagate throughout the different modules and hinder considerably the quality of the resulting KB.

Among the available EL systems we considered, namely TAGME [Ferragina and Scaiella, 2010], BABELFY and DBPEDIA Spotlight [Mendes et al., 2011], we opted for the latter, as it has shown to be the least prone to errors in musical texts (further details are provided in Section 6.1.3.3).

[5] `http://nlp.stanford.edu/software/tokenizer.shtml`

Adding Co-references

In the music domain, prototypical factoid documents such as artist biographies, album reviews, or song tidbits, normally refer to one specific entity. Based on this observation, we exploit co-referential pronouns and *resource-specific co-references*, replacing them by the name of the reported entity. A similar approach is used in [Voskarides and Meij, 2015], where the frequency of pronouns "he" and "she" is computed in every document (Wikipedia articles in this specific case) to determine the entity's gender, and then, these pronouns are replaced by the entity title. Similarly, in [Oramas et al., 2014], a gender identifier web service is used to determine the gender of subjects in artist biographies as part of an IE pipeline.

We have observed an exploitable *resource-specific co-reference* in music reviews, where terms like "this album" or "the song" can be replaced by the document's title. In the corpus used in these experiments (see Section 6.1.3.1), the expressions "this song" and "the song" are replaced with the name of the song as it appears in the document, and disambiguated with the URI of the entity they unequivocally refer to.

Co-reference resolution is a difficult and crucial task in NLP, affecting tasks such as Information Extraction [Soon et al., 2001] or document summarization [Saggion and Gaizauskas, 2004b]. It is also sensitive to the domain in which it appears (see, for instance, the case of the patents domain [Bouayad-Agha et al., 2014]). We acknolwedge the difficulty of this task. However, while addressing this problem in its entirety is out of the scope of this dissertation, the aforementioned strategy allows us to increase the coverage of entity mentions while maintaining a high precision.

Type Filtering

In DBPEDIA, most resources are associated with types via the `rdf:type` property. In addition, among the different types present in DBPEDIA (coming from the DBPEDIA ontology, YAGO types, or `schema.org`), the DBPEDIA ontology provides a relatively small and tidy taxonomy of 685 classes based on WIKIPEDIA infoboxes. Other KBS such as YAGO or Freebase have their own ontological structure, which is in general broader and noisier. MUSICBRAINZ, in contrast, has a very narrow set of entity types.

This type of information can be exploited in order to narrow down the set of allowed types for a given candidate and its potential annotations. In this way, we ensure that all entities will be, at least, related to the music domain. Restricting the search space to types such as Artist or Song reduces considerably the number of errors derived from cross-domain ambiguity. For instance, the EL system detects a substantial amount of entities whose DBpedia type is `FictionalCharacter`,

which are in most of the cases misleading song titles or band names with fictional characters of the same name. This situation is observed also with other types of entities such as *Athlete*, *Species* or *Disease*.

Depending on the envisioned application of the KB resulting from our pipeline, the predefined set of entity types may vary. In our case we restricted them to Musical Artists, Other Artists, Songs, Albums, Genres, Films and Record Labels. In Table 6.1 we present the mapping we designed between the DBPEDIA ontology, MUSICBRAINZ entity types and our selected set of types.

Our MKB	**DBPEDIA ontology**	**MUSICBRAINZ**
MusicalArtist	Person/Artist/MusicalArtist Organization/Band Writer/MusicComposer Writer/SongWriter	Artist
OtherArtist	Person/Artist (¬ MusicalArtist) Person/Writer(¬ MusicComposer & ¬ SongWriter)	—
Album	Work/MusicalWork/Album	Release
Song	Work/MusicalWork/Song Work/MusicalWork/Single	Recording Work
Genre	TopicalConcept/Genre	—
Film	Work/Film	—
RecordLabel	Agent/Organization/Company/RecordLabel	Label

Table 6.1: Music type mapping across resources

6.1.2.4 Syntactic Semantic Integration

The information obtained from the syntactic and semantic processes is combined into a graph representation of the sentence. For each music entity identified during the semantic enrichment step (Section 6.1.2.3), all nodes in the dependency tree with a correspondence with an entity mention are collapsed into one single node: *Sweet* and *Freedom* into *Seet Freedom (Album)*, and *Rod* and *Temperton* into *Rod Temperton (Artist)*. Figure 6.2 shows the resulting syntactic-semantic representation of a sentence.

6.1.2.5 Relation Extraction and Filtering

Our approach to RE is lightweight, unsupervised and rule-based. Having syntactic and semantic information available, potential relations between entities may be discovered by traversing the dependency tree. Two entities in such tree are considered to be related if there is a path between them that does not contain any

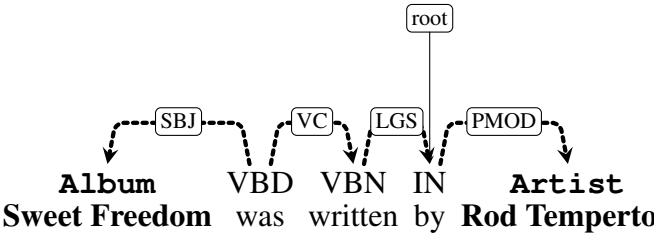


Figure 6.2: Semantic integration on syntactic dependencies.

other entity in between, and does not contain parentheses. If there is more than one path, we consider only the shortest path as the most representative path of the relation.

Our method encodes a relation pattern between two entities as all words in the shortest path between them. In the example provided in Figure 6.2, the shortest path between *Sweet Freedom* and *Rod Temperton* contains the words *was*, *written* and *by*.

While RE via shortest path in syntactic trees is common practice in the literature [Delli Bovi et al., 2015, Moro and Navigli, 2013, Nakashole et al., 2012], not all shortest paths are valid, and incorrect relations may be extracted from overly long and syntactically complex sentences. We aim at surmounting these problems by defining three filtering heuristics over surface forms (*lemma-paths*), part-of-speach patterns (*pos-paths*), and labels of syntactic dependencies (*dependency-paths*).

First, we filter out all relations with reporting verbs (e.g. "say", "tell" or "express") in the lemma-path. The intuition being that these verbs may tend to appear in syntactically more complex sentences because they enforce the inclusion of at least two verbs. Hence, semantic relations in them may not be encoded via shortest paths. We illustrate this with the following sample sentence, where the relation extracted with syntactic tree traversal by means of shortest path would be incorrect:

> **Sentence:** `Nile Rodgers` *told* NME that the first album he bought was Impressions by `John Coltrane`.
>
> **Relation**: `nile_rodgers` told that was impressions by `john_coltrane`

Second, we only selected relations where the syntactic function that connects in the dependency-path the first entity with the first word of the relation pattern is a subject (which may be preceded by a nominal modifier or an apposition), a direct or indirect object, a predicative complement or a verb chain. When this condition holds, the relation is considered *valid*. If the above condition does not hold, an extra validation step is applied over the POS-path in order to capture

relations without verbs, which seem to be idiosyncratic of the music domain, e.g. $\langle e_d, \texttt{frontman of}, e_r \rangle$, or $\langle e_d, \texttt{guitarist and singer}, e_r \rangle$.

6.1.2.6 Dependency-Based Loose Clustering

In this section we describe a simple clustering algorithm aimed at reducing the sparsity of the relation pattern set in the KB.

Let us consider the following three relation patterns: (1) *was written by blunt producer*, (2) *was written by singer/producer*, and (3) *was written by manager and guitarist*. Intuitively, these three relation patterns seem to be semantically similar, and if all of them were expressed as *was written by*, the original meaning would not be lost, and the set of relations would become more compact.

This observation, which we found to occur quite frequently, motivated the inclusion of a dependency-based loose clustering module. First, we perform a second run of dependency parsing over all relations extracted by our system, aiming at discovering their root node. We apply this second run because the root of the original sentence does not need to correspond with the relation pattern's root. Then, our algorithm considers all possible paths from the root to every leaf node of the relation pattern dependency tree, and selects the path that complies with a predefined syntactic constraint (e.g. a sequence of verbs plus adverb or preposition, or adverb plus nominal and preposition modifiers) based on regular expressions of syntactic labels. The sequence of tokens that matches this regular expression constitutes the cluster pattern. The complete set of defined regular expressions is included in the released source code (see Chapter 7).

As an illustrative case, consider the extracted relation pattern *is track was released on label* from the sentence `Sing Out The Song` *is the 7th track on Wishbone Four which was released in the UK May 1973 on the* `MCA` *label*. After re-parsing the relation pattern, we obtain the parse tree shown in Figure 6.3 and a cluster pattern over those nodes in the dependency tree that satisfy one of the regular expressions crafted in the aforementioned syntactic constraint. Finally, the obtained relation is `Sing_out_the_song` *was released on label* `MCA`. Filtering out spurious information in OIE following similar approaches has proven effective while not being computationally expensive [Fader et al., 2011].

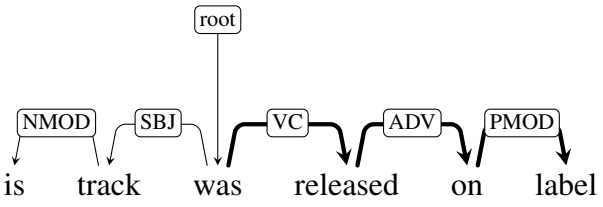


Figure 6.3: Example of a parsed relation pattern and a valid cluster pattern.

Ours is a *loose clustering* method because it does not enforce a pattern to fully match all rules, but rather allows partial matching. This module provides an enrichment of all $r \in \mathcal{R}$ such that $r = \langle e_d, e_r, \upsilon_d, \upsilon_r, p, c \rangle$, where c is the cluster pattern derived from the relation pattern p. A relation cluster is the set of all relations with the same cluster pattern, and is denoted as $\mathcal{R}_c$.

Cluster pattern c	**Typed cluster pattern** τ_c	**Relation triples** t_p
was written by	S *was written by* MA	s1 *was writen by artist* ma1
		s2 *was written by composer* ma2
		s3 *was written by singer* ma2
		s4 *was writen by* ma1
		s5 *was written by frontman* ma3
	A *was written by* MA	a1 *was written by frontman* ma3
		a2 *was written by guitarist* ma1
		a3 *was written by artist* ma2
		a4 *was written by frontman* ma5

Table 6.2: Example of a relation cluster $\mathcal{R}_c$, where $c =$ *was written by*. *S* refers to Song, *MA* to MusicalArtist and *A* to Album types, whilst *sX* refers to Song, *maX* to MusicalArtist and *aX* to Album entities.

6.1.2.7 Scoring

So far, our approach has identified entity mentions in text and has linked them in meaningful relations, filtering out those that did not comply with predefined linguistic rules. We incorporate one additional factor $score(r)$ that takes into account statistical evidence computed over $\mathcal{R}$. It has three main components, which we flesh out as follows.

We hypothesize that the relevance of a cluster may be inferred by the number and proportion of triples it encodes, and whether these are evenly distributed. Our metric encompasses a combination of three different components. First, we focus on the *degree of specificity* of the relation cluster, as previous work has demonstrated that this can contribute to improving IE pipelines [Delli Bovi et al., 2015]. Second, we analyze *intrinsic features* of the relation pattern, such as frequency, length and fluency[6]. Finally, we incorporate a *smoothing factor*, namely the pro-

[6]We adopt this term from Machine Translation, where it is used to assess how good an au-

portion of the related typed cluster pattern in the cluster.

A cluster $\mathcal{R}_c$ may be decomposed into a set of typed cluster patterns τ_c (see Table 6.2). The intuition behind the specificity measure of a cluster is that clusters with one prominent τ_c are more specific, i.e. they are largely used for encoding one specific type of relations. One example of this would be *performed with*, which enforces a relation to include MusicalArtists on both the domain and range sides. Thus, we define $\mathcal{L}_c$ as the list of cardinalities (number of triples) of every typed cluster pattern $\tau_c \in \mathcal{R}_c$, being $\mathcal{L}_c = \{|\mathcal{R}_{\tau_c^1}|, ..., |\mathcal{R}_{\tau_c^n}|\}$. We define the specificity measure as the variance of $\mathcal{L}$, expressed as $s(\mathcal{R}_c) = var(\mathcal{L}_c)$.

Furthermore, we consider a *relation's fluency* metric, which is aimed at capturing its comprehensibility. Simply put, the more the sentence's original word order is preserved in the relation pattern, the more understandable it should be. This metric is introduced due to the fact that word order is lost after modelling text via syntactic dependencies, and so we design a *penalty measure* over the number of jumps needed to reconstruct the original ordered word sequence. Let k be the number of tokens in the relation pattern, w_i the ith word in the pattern, and $h(w_i)$ a function that returns the correspondent word index in the original sentence, we put forward a fluency measure f defined as:

$$f(p) = \frac{\sum_{i=1}^{k} \alpha |h(w_i) - h(w_{i-1})|}{k} \tag{6.1}$$

where $\alpha = 2$ if $h(w_{i-1}) > h(w_i)$ and $\alpha = 1$ otherwise. Note that higher values of f means low fluency. For instance, for the relation pattern *is hit for* the score would be much higher than a mixed-up order relation pattern such as *joined because added were and hit*.

Finally, the global confidence measure for each relation $r \in R$ is expressed as follows:

$$score(r) = \left(s(\mathcal{R}_c) + \frac{|\mathcal{R}_p|}{|p| + 2^{f(p)}} \right) \times \frac{|\mathcal{R}_{\tau_c}|}{|\mathcal{R}_c|} \tag{6.2}$$

As an illustrative example of the measure, the score of a relation with the typed cluster pattern ⟨Song, *was released on*, RecordLabel⟩, will have a much higher score than a relation whose typed cluster pattern is ⟨Album, *was released on*, MusicalArtist⟩. This latter pattern is incorrect, probably due to a disambiguation error in the EL step. Relations like this show the type of errors which our proposed confidence score is expected to prune out.

tomatic translation is. In our case, a "translation" can be understood as creating a relation triple (target) from an input sentence (source).

6.1.3 Experiments

In this section, we describe our experimental setting. We refer first to the source raw corpus, and second to the resulting KBs as output of different branches of our approach.

6.1.3.1 Source dataset

Songfacts[7] is an online database that collects, stores and provides facts, stories and trivia about songs. These are collaboratively written by registered users, and reviewed by the website staff. It contains information about more than 30,000 songs from nearly 6,000 artists. This information may refer to what the song is about, who wrote it, who produced it, who collaborated with whom or who directed the videoclip associated with the song. These texts are rich sources of information not only for well-known music facts, but also for music-specific trivia, as in the following sample sentence (about David Bowie's *Space Oddity*): "Bowie wrote this song after seeing the 1968 Stanley Kubrick movie 2001: A Space Odyssey".

We crawled the Songfacts website in mid-January 2014. Then, for each song article, we performed a mapping between the song and its MUSICBRAINZ song ID, using the MUSICBRAINZ Search API. We successfully mapped 27,655 songs.

The RE pipeline was run over the 27,655 document Songfacts corpus, which amounts to 306,398 sentences. After the Semantic Processing step, we obtained 202,767 entity mentions (8,880 for *Albums*, 3,136 *Record Labels*, 74,908 *Songs*, 107,253 *Musical Artists*, 1,760 *Genre* labels, 3,467 for *Other Artist*, and 3,363 for *Film*). There were 48,122 sentences with at least two entities, and it is on this subset where we apply our RE pipeline.

6.1.3.2 Learned Knowledge Bases

Our aim is to assess to what extent each of the modules integrating our approach contributes to the quality of the resulting KB. After executing the whole pipeline, we generate two *learned* KBs (KBSF-ft and KBSF-th), two *baseline* KBs (KBSF-co and KBSF-raw), and a *competitor* KB (KBSF-rv).

The *learned* KBs are the result of applying the RE method to the Songfacts dataset under different conditions. KBSF-ft is derived from applying the RE pipeline entirely, and KBSF-th comes from a selection of all triples in KBSF-ft with a confidence score above a certain threshold (which comes from the metric described in Section 6.1.2.7). To determine the best threshold to prune KBSF-ft, we aimed at maximizing the number of triples and at the same time minimizing the number of relation patterns. Our intuition is that less patterns means a tidier

[7] http://www.songfacts.com

KB. Therefore, we computed the percentage of triples and relation patterns from KBSF-ft that remain in a pruned KB, whose triples have a score greater than a certain threshold θ. We computed these percentages for every θ value ranging from 0 to 1 in bins of 0.01 (see Figure 6.4). Our goal was to discover the θ value which maximizes the distance between the amount of triples and the amount of relation patterns in a pruned KB. After confirming a maximized difference with $\theta = 0.05$, we created KBSF-th, whose triples have a score greater than or equal to 0.05. In this pruned KB, we have 36.56% of KBSF-ft triples, with only 12.52% of its relation patterns.

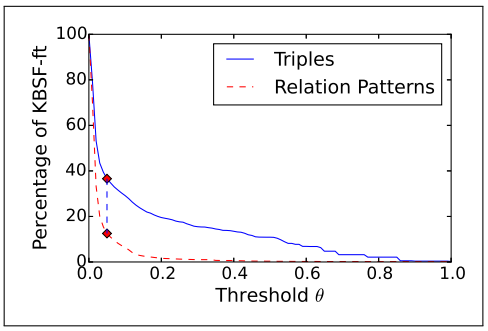


Figure 6.4: Percentage of triples and relation patterns from KBSF-ft that remain after pruning at different values of θ. Maximum distance at $\theta = 0.05$.

In addition, we created two baseline KBs for evaluation purposes. KBSF-co is a baseline which consists of simple entity co-occurrence. More specifically, if two entities are mentioned in the same sentence, an unlabelled triple that anchors them is added to the KB. In addition, KBSF-raw was created following the RE pipeline, but without applying the filtering process described in Section 6.1.2.5. Finally, KBSF-rv constitutes the competitor KB, and is built as follows: After running REVERB over the Songfacts dataset, we search coinciding relations, at both domain and range positions, that include entity mentions identified in our disambiguation step. These relations are included in KBSF-rv. Statistics about the five KBs are reported in Table 6.3.

KB	Entities	Triples	Relation Patterns	Cluster Patterns
KBSF-ft	20,744	32,055	20,438	14,481
KBSF-th	10,977	11,720	2,484	828
KBSF-co	30,671	113,561	—	—
KBSF-raw	29,280	71,517	47,089	32,712
KBSF-rv	9,255	7,532	2,830	—

Table 6.3: Statistics of all the learned KBs

6.1.3.3 Quality of Entity Linking

We mentioned in Section 6.1.2.3 the lacking of both music-specific EL tools as well as benchmarking datasets. For this reason, we performed a set of experiments to select the best-suited EL tool for the music domain, among some of the best known and reputed. Specifically, we perform evaluation experiments on DBPEDIA Spotlight, TAGME and BABELFY.

As of now, most EL systems *speak their own language*, partially due to the fact that they perform entity disambiguation with different KBs as reference. Since their output is heterogeneous in format, performing a comparison between them is not straightforward. In order to evaluate the aforementioned EL systems, we used ELVIS[8] [Oramas et al., 2016], an EL integration tool which provides a common output for different EL system. In addition, we created a dataset of annotated musical entities and applied both quantitative and qualitative evaluations in order to verify which system performs better with musical entities, and is more suitable for our task.

Let us begin the evaluation of the EL approach by describing the collection and preparation of our evaluation data. The result of this process is an *ad-hoc* gold standard dataset used to evaluate the different EL systems, with the Songfacts corpus (Section 6.1.3.1) as our testbed. In Songfacts, each document univocously refers to one single song. In addition, we have information about artist and song names at our disposal. We used this information to obtain the MUSICBRAINZ ID for songs and artists. In MUSICBRAINZ, artist and song items sometimes have information about their equivalent WIKIPEDIA page. We leveraged this information, when available, to obtain their corresponding DBPEDIA URIs. Finally, we obtained a mapping with DBPEDIA of 7,691 songs and 3,670 artists. From the DBPEDIA resources of each song, we gathered their corresponding album name and URI, if available, obtaining information of about 2,092 albums. Then, for every document, we looked for exact string matches of the reported song, and its related album and artist names. Every detected entity is thus annotated with its DBPEDIA URI. At the end of this process, the newly created gold standard dataset contains 6,052 documents where 17,583 sentences are annotated with the following entities: 5,981 Song, 12,137 Artist and 1,722 Album entities. As mentioned in Section 6.1.2.3, there are typical cases of ambiguity in musical entities where songs, artists and albums can potentially share the same name. Therefore, we manually corrected the entities detected in 212 documents where this kind of ambiguity was present.

The three EL systems under review provide their own confidence measure. Hence, we evaluated their output filtering out the entities with a confidence measure below to a certain threshold θ. We run the evaluation for different values of θ,

[8] https://github.com/sergiooramas/elvis

	Album		Artist		Song		Macro Average		
	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	F-measure
Babelfy	0.93	0.28	0.98	0.55	0.96	0.31	0.96	0.38	0.54
Tagme	0.75	0.69	0.97	0.77	0.65	0.71	0.79	0.72	**0.76**
Spotlight	0.80	0.52	0.94	0.83	0.59	0.42	0.78	0.59	0.67

Table 6.4: Precision and recall of the EL Systems considered

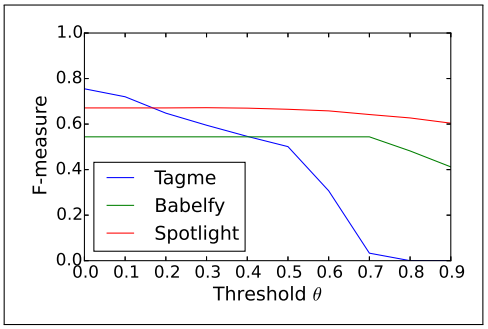


Figure 6.5: F-measure of the EL systems at different confidence thresholds

ranging from 0 to 0.9 in bins of 0.1. After evaluating on the gold dataset, the best results in terms of F-measure were obtained by all the systems at $\theta = 0$ (see Figure 6.5), which means that there is no need to apply any filtering process based on the EL system own confidence score. Detailed results on the run of every system at $\theta = 0$ are shown in Table 6.4. We used macro-average Precision and Recall measures, i.e. we averaged their values from the three sets of entities.

We may conclude from these results that Babelfy is the system with highest Precision on musical entities. However, its recall is lower than the other systems under consideration, and specifically with respect to Tagme, which in turn, shows much lower precision. DBpedia Spotlight, on the other hand, achieves a similar precision score as Tagme, but with a slightly lower recall.

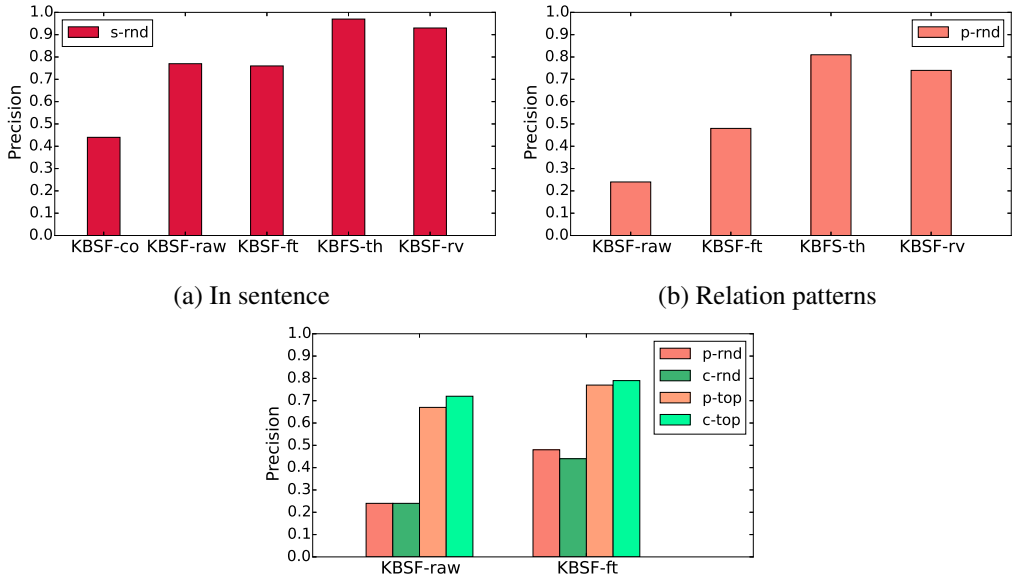
This evaluation experiment is only focused on measuring the precision in the annotation of entities present in the gold standard. However, since all possible entities in a document may be not annotated, we also report on specific types of false positives which emerged during a qualitative inspection of classification results. For example, a frequent error that is not being evaluated concerns cases in which a text span not annotated in the ground truth is identified incorrectly as an entity by any system. Therefore, to complement the evaluation, we listed the most frequently identified entities by each system (see Table 6.5). As we can see,

Babelfy and Tagme are misidentifying common words as entities very frequently, whereas DBpedia Spotlight is not doing so. These errors may propagate to the rest of the IE pipeline, penalizing the accuracy of the final KB. Although a filtering process could be applied to filter out misidentified entities by computing their tf-idf score in each document, we opted for using DBpedia Spotlight, as it has shown pretty good performance, its output does not require any further processing, and it is released as open source, which means that there are no limitations on the number of queries.

System	Song	Album	Artist
Babelfy	**Carey**	**Debut**	John_Lennon
	Stephen	**Song_For**	Eminem
	Rap_Song	**Sort_Of**	Paul_McCartney
	Singing_This_Song	**First_Song**	Bob_Dylan
	A_Day_in_the_Life	**Debut_Album**	Drake
Tagme	**The_Word**	**Up!**	John_Lennon
	The_End	**When_We_On**	**The_Notorious_B.I.G.**
	If	**U**p	**Do**
	Once	**Together**	Paul_McCartney
	For_You	**By_the_Way**	**Neil_Young**
Spotlight	Sexy_Sadie	The_Wall	Madonna
	Helter_Skelter	Let_It_Be	Eminem
	Cleveland_Rocks	Born_This_Way	Rihanna
	Stairway_to_Heaven	Thriller	John_Lennon
	Minnie_the_Moocher	Robyn	Britney_Spears

Table 6.5: Top-5 most frequent entities by type and tool.

IE evaluation is a difficult task in restricted domains, where ground truth data is usually scarce, and also semantic relations between entities may vary in terms of correctness over time. Also, correct relations may be linguistically flawed, i.e. not fluent. Previous approaches assessed automatically extracted relations in terms of correctness according to human judgement [Fader et al., 2011, Mausam et al., 2012]. Additionally, a finer grained analysis is carried out in [Banko et al., 2007], adding a prior step in which relations are judged as being *concrete* or *abstract*.



(a) In sentence

(b) Relation patterns

(c) Patterns and clusters

Figure 6.6: Precision of relations at sentence (*s*), relation pattern (*p*) and cluster pattern (*c*) levels in top (*top*) and random (*rnd*) samples of relations

For this experiment, we made use of extensive human input and asked two experts in Computational Linguistics to evaluate the *top 100* scoring relations as yielded by our weighting policy (Section 6.1.2.7), as well as a random sample of 100 relations. The original sentence from which the relations came from was available to them. This was done for all the KBs produced by our pipeline and for KBSF-rv. Cohen's kappa coefficient ranged from 0.60 to 0.81, which is generally considered as *substantial* agreement.

In Figures 6.6a and 6.6b, where we compare random samples from each KB, we observe a gradual improvement of the quality of relations as the different modules of our implementation are incorporated. The difference between these figures is that in the former, a relation is deemed correct if it has extracted a relation *expressed in the original sentence*, whereas the latter figure reports numbers on whether the extracted relation pattern was correct, i.e. if it *meant* the same as it was intended in the source sentence. We may infer from these results that co-occurrence between entities does not guarantee an explicit relation, whereas the presence of a path between two entities over a sentence dependency tree, without any other entity mention in between, generally suggests a monsemous and unambiguous relation.

It is remarkable how well REVERB performs (Figure 6.6b), only being surpassed by the KB resulting from our most sophisticated implementation of MKB.

We note that the good results of the REVERB extractor are also due to the semantic processing of our system, which is forcing REVERB to select good candidates as relation arguments. Recall that the difference between KBSF-ft and KBSC-th is the inclusion of the *scoring* module, and the increase in Precision confirms that incorporating *statistical evidence contributes to better relations*.

This is further confirmed in the results showcased in Figure 6.6c, where we provide a comparison between top 100 relations according to our ranking policy against a random sample. Note that *in all* KBS*, highly scoring relations are more often marked as correct*, which constitutes additional support for the contribution of the scoring module. Together with the quality of the relation pattern, this figure shows the quality of the cluster pattern associated with the evaluated relations. We observe that cluster patterns inferred in our clustering module have similar quality than relation patterns in the random sample, and slightly better in the top 100 sample. This result implies that the scoring module is rewarding good clusters.

Next, we present results on the **coverage of the extracted KB**. With this experiment, we aim to compare the coverage of music relations in our KBS with respect to other resources requiring human intervention, such as DBPEDIA, MUSICBRAINZ, as well as resources created fully automatically. For the latter, we considered DEFIE as our closest competitor due to several methodological similarities (e.g., the use of dependency parsing, EL and RE over shortest paths).

We selected all triples in KBSF-th whose domain and range entities could be mapped to both DBPEDIA and MUSICBRAINZ. As our extracted KB has only MusicBrainz ID of entities of types MusicalArtist and Song, the set of triples to evaluate is restricted to relations between them. Since entities in DEFIE are disambiguated against BABELNET ids, we mapped all DBPEDIA uris to their corresponding BABELNET id, which yielded a subset of 3,633 triples. From here, we selected all possible domain-range entity pairs, and retrieved from the other KBS all triples with the same pairs, and counted them. The procedure to do so on DBPEDIA was via SPARQL queries. We discarded triples with predicate *wikiPageWikiLink*, as this predicate means an unlabeled relation. However, the mapping with MUSICBRAINZ was not trivial. MUSICBRAINZ is not a KB of triples, but a relational database. Entities are stored in tables, and relations between entities are represented in a set of tables of relations, having one table for each possible relation. The entities in the studied set of triples were only of type MusicalArtist and Song. However, an entity of type Song in KBSF-th can be related to either a Recording or a Work entity in MUSICBRAINZ (see Section 6.1.2). Therefore, for the analysis of relations involving a Song entity, we obtained the equivalent Recording and Work MUSICBRAINZ entities, and looked up relations where any of them where present.

Mapping results are shown in Table 6.6. Let us highlight the fact that most semantic relations encoded in KBSF-th are novel, as they were not found in any

of the other resources we compared against. In the overlapping cases, most of the times the relation labels were semantically equivalent, and often the relation label of KBSF-th triples was more specific than the ones retrieved from other KBs (e.g. *frontman* and *member of*). It could be argued that with our proposed approach, it is possible to find complementary information about musical entities which is not previously defined in general-purpose KBs.

	KBSF-th	MusicBrainz	DBpedia	DefIE
Relation instances	3,633	1,535	1,240	456

Table 6.6: Number of triples with labeled relations in the different KBs for the same set of domain-range entity pairs

6.1.3.4 Interpretation of Music Recommendations

The main aim of this experiment is to evaluate the suitability of KBSF-th to explain relations between songs, and study their impact on user's experience in music recommendation. Since our aim is not to measure the performance of a recommender system, we implemented a baseline recommender approach. Recommendations are based on the concept of song similarity, which exploits the graph-based structure of our KB, following [Oramas et al., 2015b].

We designed the experiment as an online survey, where the participant is first asked to select 5 songs from different artists of his/her choice. From each selected song, the system randomly selects 3 recommendations among the list of its top-10 most similar songs. One of them is shown together with an explanation in natural language (the source text), another with an explanation based on relation patterns, and finally the third one appears without explanation. Participants can listen to all songs with an embedded player. After listening to the recommendation and reading the explanation attached to it, participants were asked to rate each recommendation from 1 to 5 (1 being worst), and to mention whether they were familiar or not with the recommended songs (see Figure 6.7).

The experiment involved 35 participants, 28 males and 7 females, ranging from 26 to 38 years old and with different musical background and listening habits. Most of the participants said that they had previous experience with recommendation systems. A total of 525 answers (corresponding to individual song recommendations) were collected. In 38% of the cases, the user was familiar with the recommended songs.

The average rating of recommendations with natural language explanations is slightly higher (3.20 ± 1.29) than recommendations without explanations (3.08 ± 1.35),

or with explanations based on relation labels (3.04±1.34). In addition, for musically educated individuals, recommendations of unfamiliar songs, whether accompanied with or without explanations, have similar average rating (2.87 and 2.95 respectively). However, for untrained users, recommendations with explanations have a remarkable higher average rating (2.93) than without them (2.36). Thus, we can infer that the introduction of explanations in recommender systems improves the user experience of musically untrained subjects when discovering songs.

We also asked the subjects to select among a set of adjectives those that better described the recommendation experience. The general trend was to rate positively the experiment. Most users rated the experience as *enjoyable* (40%), followed by *useful* (31%) and enriching (29%). Negativity was much lower in general, with *confusing* being the most voted (17%), followed by *complicated* and *too geeky* (8% in both cases). This suggests that the introduction of explanations generated from our MKB in the recommendations was in general a satisfactory experience to users.

Figure 6.7: User interface for the music recommendation experiment.

6.1.4 Conclusion

We have presented an NLP pipeline that learns a Knowledge Base in the music domain taking raw text collections as input. It combines methods easily applicable to a general purpose application with domain-specific heuristics which are designed to exploit particularities of the domain.

The result of applying our approach over a dataset of stories about songs is a new Music Knowledge Base, which encodes semantic relations among musical entities. Our method relies on the syntactic structure (defined via dependency parsing) of sentences and the use and adaptation of music-specific heuristics for both EL and RE. In addition, we include modules for semantic clustering and pattern scoring, aimed at the efficient removal of noisy relations. Our modular evaluation shows that our RE module is able to capture a highly precise and compact set of weighted triples, and demonstrates the positive impact of the novel scoring metric we introduced. Moreover, we have shown that a high percentage of the knowledge encoded in our MKB is not present in other KBs, both general and domain-specific. Finally, regarding extrinsic evaluation, the experiment on recommendation interpretation confirms that explanations based on the learned KB are positively regarded by the users.

6.2 KB-Unify: Knowledge Base Unification via Sense Embeddings and Disambiguation

So far, in this dissertation we have discussed extensively methods to formalize knowledge from corpora. Either in the form of identifying relevant snippets of text (definitions), inferring semantic relations (hypernymy), constructing full-fledged lexical taxonomies, or creating domain-specific KBs from raw text. However, one challenge remains when these and other systems have to "speak to each other". There are plenty of knowledge extraction systems which have shown high quality performance in unrestrained settings, however they are mostly constrained by the fact that their reference ontological structure is created *ad-hoc* (e.g. in NELL), or they may lack one (e.g. ReVerb). This is an enormous bottleneck when it comes to leveraging information obtained from heterogenous systems. In this context, what follows is a description of our contribution for the seamless integration of the output of OIE systems, a system called **KB-Unify**. This is a unification algorithm that contains, into one single knowledge repository, disambiguated output in the form of semantic triples (argument, relation, argument) coming from systems that may or may note provide a disambiguated (linked) output.

6.2.1 Introduction

The breakthrough of the OIE paradigm has opened up a research area where Web-scale unconstrained IE systems are developed to acquire and formalize large quantities of knowledge. However, while successful, to date most state-of-the-art OIE systems have been developed with their own type inventories, and no portable ontological structure. In fact, OIE systems can be very different in nature. Early

approaches [Etzioni et al., 2008, Wu and Weld, 2010, Fader et al., 2011] focused on extracting a large number of relations from massive unstructured corpora, mostly relying on dependencies at the level of surface text. Systems like [Carlson et al., 2010] combine a hand-crafted taxonomy of entities and relations with self-supervised large-scale extraction from the Web, but they require additional processing for linking and integration [Dutta et al., 2014].

More recent work has focused, instead, on deeper language understanding, especially at the level of syntax and semantics [Nakashole et al., 2012, Moro and Navigli, 2013]. By leveraging semantic analysis, knowledge gathered from unstructured text can be adequately integrated and used to enrich existing knowledge bases, such as YAGO, FREEBASE [Bollacker et al., 2008] or DBPEDIA. A large amount of reliable structured knowledge is crucial for OIE approaches based on distant supervision [Mintz et al., 2009, Riedel et al., 2010], even when multi-instance multi-learning algorithms [Surdeanu et al., 2012] or matrix factorization techniques [Riedel et al., 2013, Fan et al., 2014] come into play to deal with noisy extractions. For this reason a recent trend of research has focused on KB completion [Nickel et al., 2012, Bordes et al., 2013], exploiting the fact that distantly supervised OIE and structured knowledge can complement each other. However, the majority of integration approaches nowadays are not designed to deal with many different resources at the same time.

Hence, we propose an approach where the key idea is to bring together knowledge drawn from an arbitrary number of OIE systems, regardless of whether these systems provide links to some general-purpose inventory, come with their own ad-hoc structure, or have no structure at all. Knowledge from each source, in the form of ⟨subject, predicate, object⟩ triples, is disambiguated and linked to a single large sense inventory. This enables us to discover alignments at a semantic level between relations from different KBs, and to generate a unified, fully disambiguated KB of entities and semantic relations. KB-UNIFY achieves state-of-the-art disambiguation and provides a general, resource-independent representation of semantic relations, suitable for any kind of KB.

6.2.1.1 An Example

Different OIE systems may encode the same relation with a slightly different word choice. This can be caused by being used to process different corpora, their scoring algorithm, or the quality of any library or text processing tool used (e.g. POS taggers or parsers). Let us look at a few illustrative sample relations, used to refer to entities or concepts which are known for "dropping" (we leave the interpretation open enough to acomodate different senses for this word). In ReVerb, for instance, we find triples such as ⟨*educational standard*, *has dropped below*, *acceptable level*⟩ or ⟨*temperatures*, *get below*, *freezing*⟩, while in WiseNet, there

exists the triple ⟨*outside temperature*, *was below the*, *freezing*⟩.

It would be desirable, for an automatically generated KB, to consider such rich and fine-grained relations as the ones acquired by these (and other) OIE systems, and unify them into one *relation synset* which could be defined as the accumulation of several verbalizations used for referring to the same fact. This unified knowledge can further be exploited to improve ontologies or directly in inference and semantic systems which today are expected to be aware of world knowledge (e.g. the fact that temperatures, prices or educational standards, to name extremely unrelated concepts, can "drop below", "go down" or "get below" a certain degree, a certain level, or below a certain threshold).

KB-UNIFY explicitly addresses this challenging problem, with the added value of not requiring an input system to be previously disambiguated against any reference ontology or sense inventory (e.g. ReVerb relations, which are at the text string level, are perfectly valid). Following our running example, KB-UNIFY encoded as one *relation synset* the following relations (coming from ReVerb and WiseNet)[9]: *have drop below*, *can go below*, *stay below*, *can fall below*, *hover just above*, *reach below*, *climb above*, *range from below*, *be expect to remain below*, *was below the*, *may drop below*, *get below*. Note that the *climb above* relation is incorrectly included in this synset. This is explained by the tendency of antonyms to co-occur in the same context, and thus distributional approaches (such as the one we follow in KB-UNIFY) may tend to assign them similar vectors, as we discussed in Chapter 1.

After having provided a motivation and a working example on how the result of our approach can be further leveraged for providing better quality information to intelligent systems, we proceed in what follows to describe our method for constructing KB-UNIFY.

6.2.2 Knowledge Base Unification: Overview

KB-UNIFY takes as input a set of KBs $\mathbf{K} = \{KB_1, ..., KB_n\}$ and outputs a single, unified and fully disambiguated KB, denoted as $\mathbf{KB}^*$. For our purposes we can define a KB KB_i as a triple $\langle E_i, R_i, T_i \rangle$, where E_i is a set of entities, R_i is a set of semantic relations, and T_i is a set of triples (facts) $\langle e_d, r, e_g \rangle$ with subject and object e_d, $e_g \in E_i$ and predicate $r \in R_i$. Depending on the nature of each KB_i, entities in E_i might be disambiguated and linked to an external inventory (e.g. the entity *Washington* linked to the Wikipedia page GEORGE WASHINGTON), or unlinked and only available as ambiguous mentions (e.g. the bare word *washington* might refer to the president, the city or the state). We can thus partition $\mathbf{K}$ into a subset of linked resources $\mathbf{K}_D$, and one of unlinked resources $\mathbf{K}_U$. In order

[9]Note that relations are lemmatized in ReVerb.

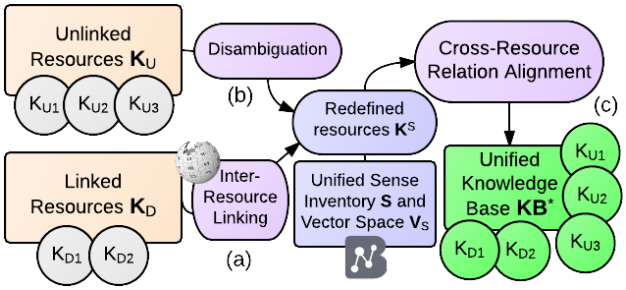


Figure 6.8: Unification algorithm workflow

to align very different and heterogeneous KBs at the semantic level, KB-UNIFY exploits:

- A unified sense inventory S, which acts as a superset for the inventories of individual KBs. We choose BabelNet for this purpose: by merging complementary knowledge from different resources (e.g. Wikipedia, WordNet, Wikidata and Wiktionary, among others), BabelNet provides a wide coverage of entities and concepts whilst at the same time enabling convenient inter-resource mappings for KB_i in $\mathbf{K}_D$. For instance, each Wikipedia page (or Wikidata item) has a corresponding synset in BabelNet, which enables a one-to-one mapping between BabelNet's synsets and entries in, e.g., DBPEDIA or FREEBASE;

- A vector space model V_S that enables a semantic representation for every item in S. Current distributional models, like word embeddings, are not suitable to our setting: they are constrained to surface word forms, and hence they inherently retain ambiguity of polysemous words and entity mentions. We thus leverage SENSEMBED (cf. Chapter 3).

Figure 6.8 illustrates the workflow of our KB unification approach. Entities coming from any $KB_i \in \mathbf{K}_D$ can be directly (and unambiguously) mapped to the corresponding entries in S via BabelNet inter-resource linking (Figure 6.8(a)): in the above example, the entity *Washington* linked to the Wikipedia page GEORGE WASHINGTON is included in the BabelNet synset $\texttt{Washington}_{bn}$. In contrast, unlinked (and potentially ambiguous) entities need an explicit disambiguation step (Figure 6.8(b)) connecting them to appropriate entries, i.e. synsets, in S: this is the case, in the above example, for the ambiguous mention *washington* that has to be linked to either the president, the city, the state or any other entity named

washington which may or may not be included in BabelNet[10]. Our approach, thus, comprises two successive stages:

- A **disambiguation** stage (Section 6.2.3) where all $KB_i \in \mathbf{K}$ are linked to S, either by inter-resource mapping (Figure 6.8(a)) or disambiguation (Figure 6.8(b)), and all E_i are merged into a unified set of entities E^*. As a result of this process we obtain a set $\mathbf{K}^S$ comprising all the KBs in $\mathbf{K}$ redefined using the common sense inventory S;

- An **alignment** stage (Section 6.2.7, Figure 6.8(c)) where, for each pair of KBs $KB_i^S, KB_j^S \in \mathbf{K}^S$, we compare any relation pair $\langle r_i, r_j \rangle$, $r_i \in R_i^S$ and $r_j \in R_j^S$, in order to identify cross-resource alignments and merge relations sharing equivalent semantics into relation clusters (*relation synsets*). This process yields a unified set of relation synsets R^*. The overall result is $\mathbf{KB}^* = \langle E^*, R^*, T^* \rangle$, where T^* is the set of all disambiguated triples redefined over E^* and R^*.

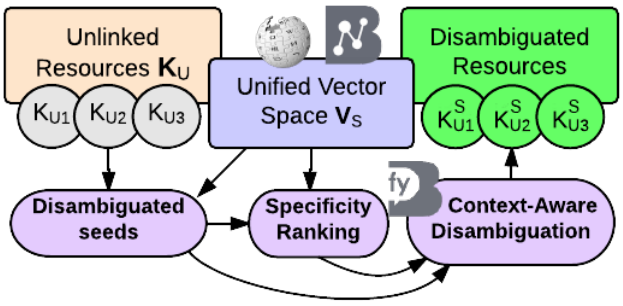


Figure 6.9: Disambiguation algorithm workflow

6.2.3 Disambiguation

In the disambiguation phase (Figure 6.8(b)), all $KB_i \in \mathbf{K}_U$ are linked to the unified sense inventory S and added to the set of redefined KBs $\mathbf{K}^S$. As explained in Section 6.2.2, while each KB in $\mathbf{K}_D$ can be unambiguously redefined via BabelNet inter-resource links and added to $\mathbf{K}^S$, KBs in $\mathbf{K}_U$ require an explicit disambiguation step. Given a $KB_i \in \mathbf{K}_U$, our disambiguation module (Figure 6.9) takes as input its set of unlinked triples T_i and outputs a set $T_i^S \subseteq T_i$ of disambiguated triples with subject-object pairs linked to S. The triples in T_i^S, together with their

[10]Modeling unseen entities and incorporating them to a reference ontology or KB remains a task for future work.

corresponding entity sets and relation sets, constitute the redefined KB_i^S which is then added to $\mathbf{K}^S$. However, applying a straightforward approach that disambiguates all triples in isolation might lead to very imprecise results, due to the lack of available context for each individual triple. We thus devised a disambiguation strategy that comprises three stages:

1. We identify a set of high-confidence seeds from T_i (Section 6.2.4), i.e. triples $\langle e_d, r, e_g \rangle$ where subject e_d and object e_g are highly semantically related, and disambiguate them using the senses that maximize their similarity in our vector space V_S;

2. We use the seeds to generate a ranking of the relations in R_i according to their degree of specificity (Section 6.2.5). We represent each $r \in R_i$ in our vector space V_S and assign higher specificity to relations whose arguments are closer in V_S;

3. We finally disambiguate the remaining non-seed triples in T_i (Section 6.2.6) starting from the most specific relations, and jointly using all participating argument pairs as context.

6.2.4 Identifying Seed Argument Pairs

The first stage of our disambiguation approach aims at extracting reliable seeds from T_i, i.e. triples $\langle e_d, r, e_g \rangle$ where subject e_d and object e_g can be confidently disambiguated without additional context. In order to do this we leverage the sense embeddings associated with each candidate disambiguation for e_d and e_g, and perform the same disambiguation strategy as in Section 5.1.4, obtaining disambiguated triples $\langle s_d^*, r, s_g^* \rangle$. The cosine similarity value associated with $\langle v_d^*, v_g^* \rangle$ represents the disambiguation confidence ζ_{dis}. We rank all such triples according to their confidence, and select those above a given threshold δ_{dis}. The underlying assumption is that, for high-confidence subject-object pairs, the embeddings associated with the correct senses s_d^* and s_g^* will be closest in V_S compared to any other candidate pair. Intuitively, the more the relation r between e_d and e_g is semantically well defined, the more this assumption is justified. As an example, consider the triple $\langle$*Armstrong*, *worked for*, *NASA*$\rangle$: among all the possible senses for *Armstrong* (the astronaut, the jazz musician or the cyclist) and *NASA* (the space agency, the racing organization or the Swedish band) we expect the vectors corresponding to the astronaut and the space agency to be closest in the vector space model V_S.

6.2.5 Relation Specificity Ranking

The assumption that, given an ambiguous subject-object pair, correct argument senses are the closest pair in the vector space (Section 6.2.4) is easily verifiable for general relations (e.g. *is a*, *is part of*). However, as a semantic relation becomes specific, its arguments are less guaranteed to be semantically related (e.g. *is a professor in the university of*) and a disambiguation approach based exclusively on similarity is prone to errors. On the other hand, specific relations tend to narrow down the scope of possible entity types occurring as subject and object. In the above example, *is a professor in the university of* requires entity pairs with professors as subjects and cities as objects. Our disambiguation strategy should thus vary according to the specificity of the relations taken into account. In order to consider this observation in our disambiguation pipeline, we first need to estimate the degree of specificity for each relation in the relation set R_i of the target KB to be disambiguated. Given R_i and a set of seeds from the previous stage (Section 6.2.4), we apply a specificity ranking policy and sort relations in R_i from the most general to the most specific. We compute the generality $Gen(r)$ of a given relation r by looking at the spatial dispersion of the sense embeddings associated with its seed subjects and objects. Let $\mathbf{v}_D$ ($\mathbf{v}_G$) be the set of sense embeddings associated with the domain (range) seed arguments of r. For both $\mathbf{v}_D$ and $\mathbf{v}_G$, we compute the corresponding centroid vectors μ_D and μ_G as:

$$\mu_k = \frac{1}{|\mathbf{v}_k|} \sum_{v \in \mathbf{v}_k} \frac{v}{\|v\|}, \qquad k \in \{D, G\} \tag{6.3}$$

Then, the variances σ_D^2 and σ_G^2 are given by:

$$\sigma_k^2 = \frac{1}{|\mathbf{v}_k|} \sum_{v \in \mathbf{v}_k} \left(1 - cos\left(v, \mu_k\right)\right)^2 \tag{6.4}$$

with $k \in \{D, G\}$ as before. We finally compute $Gen(r)$ as the average of σ_D^2 and σ_G^2. The result of this procedure is a *relation specificity ranking* that associates each relation r with its generality $Gen(r)$. Intuitively, we expect more general relations to show higher variance (hence higher $Gen(r)$), as their subjects and objects are likely to be rather disperse throughout the vector space; instead, arguments of very specific relations are more likely to be clustered together in compact regions, yielding lower values of $Gen(r)$ (see Section 6.2.8.2 for an evaluation of this approach).

6.2.6 Disambiguation with Relation Context

In the third step, both the specificity ranking and the seeds are exploited to disambiguate the remaining triples in T_i. To do this we leverage the EL and WSD system

BABELFY [Moro et al., 2014] (introduced in Section 3.2.1). As we observe in Section 6.2.8.2, specific relations impose constraints on their subject-object types and tend to show compact domains and ranges in the vector space. Therefore, given a triple $\langle e_d, r, e_g \rangle$, knowing that r is specific enables us to put together all the triples in T_i where r occurs, and use them to provide meaningful context for disambiguation. If r is general, instead, its subject-object types are less constrained and additional triples do not guarantee to provide semantically related context.

At this stage, our algorithm takes as input the set of triples T_i, along with the associated disambiguation seeds (Section 6.2.4), the specificity ranking (Section 6.2.5) and a specificity threshold δ_{spec}. T_i is first partitioned into two subsets: T_i^{spec}, comprising all the triples for which $Gen(r) < \delta_{spec}$, and $T_i^{gen} = T_i \setminus T_i^{spec}$. We then employ two different disambiguation strategies:

- For each distinct relation r occurring in T_i^{spec}, we first retrieve the subset $T_{i,r}^{spec} \subset T_i^{spec}$ of triples where r occurs, and then disambiguate $T_{i,r}^{spec}$ as a whole with BABELFY. For each triple in $T_{i,r}^{spec}$, context is provided by all the remaining triples along with the disambiguated seeds extracted for r.

- We disambiguate the remaining triples in T_i^{gen} one by one in isolation with BABELFY, providing for each triple only the predicate string r as additional context.

6.2.7 Cross-Resource Relation Alignment

After disambiguation (Section 6.2.3) each KB in $\mathbf{K}$ is linked to the unified sense inventory S and added to $\mathbf{K}^S$. However, each $KB_i^S \in \mathbf{K}^S$ still provides its own relation set $R_i^S \subseteq R_i$. Instead, in the unified $\mathbf{KB}^*$, relations with equivalent semantics should be considered as part of a single relation synset even when they come from different KBs. Therefore, at this stage, we apply an alignment algorithm to identify pairs of relations from different KBs having equivalent semantics. We exploit the fact that each relation r is now defined over entity pairs linked to S, and we generate a semantic representation of r in the vector space V_S based on the centroid vectors of its domain and range. Due to representing the semantics of relations on this common ground, we can compare them by computing their domain and range similarity in V_S. We first consider each $KB_i^S \in \mathbf{K}^S$ and, for each relation r_i in R_i^S, we compute the corresponding centroid vectors $\mu_d^{r_i}$ and $\mu_g^{r_i}$ using formula 6.3. Then, for each pair of KBs $\langle KB_i^S, KB_j^S \rangle \in \mathbf{K}^S \times \mathbf{K}^S$, we compare all relation pairs $\langle r_i, r_j \rangle \in R_i^S \times R_j^S$ by computing the cosine similarity between domain centroids s_D and between range centroids s_G:

$$s_k = \frac{\mu_k^{r_i} \cdot \mu_k^{r_j}}{\|\mu_k^{r_i}\| \, \|\mu_k^{r_j}\|} \tag{6.5}$$

	$\mathbf{K}_U$		$\mathbf{K}_D$	
	NELL	REVERB	PATTY	WISENET
# relations	298	1,299,844	1,631,531	245 935
# triples	2,245,050	14,728,268	15,802,946	2,271 807
# entities	1,996,021	3,327,425	1,087,907	1,636 307

Table 6.7: Statistics on the input KBs for KBU

where μ_k^r denotes the centroid associated with relation r and $k \in \{D, G\}$. The average of s_D and s_G gives us an *alignment confidence* ζ_{align} for the pair $\langle r_i, r_j \rangle$. If confidence is above a given threshold δ_{align} then r_i and r_j are merged into the same relation synset. Relations for which no alignment is found are turned into singleton relation synsets. As a result of this alignment procedure we obtain the unified set of relations R^*.

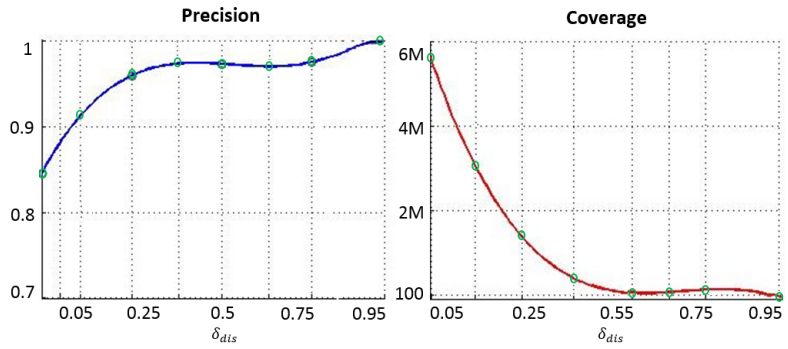
6.2.8 Evaluation

The setting for our experimental evaluation was the following:

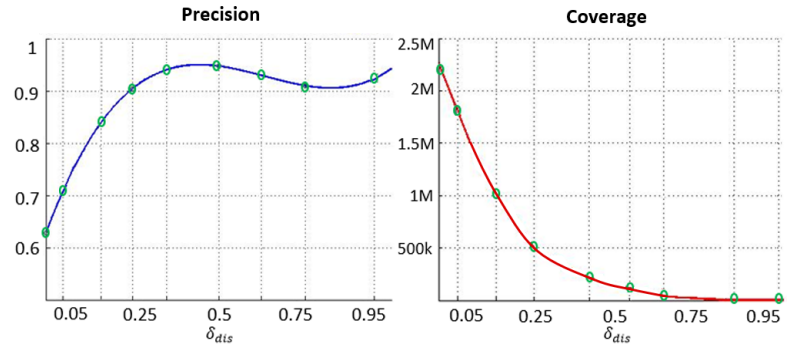
- We used BabelNet 3.0 as our unified sense inventory for the unification procedure as well as the underlying inventory for both BABELFY and SENSEMBED. Currently, BabelNet contains around 14M synsets and represents the largest single multilingual repository of entities and concepts;
- We selected PATTY [Nakashole et al., 2012] and WISENET [Moro and Navigli, 2013] as linked resources. We used PATTY with FREEBASE types and pattern synsets derived from Wikipedia, and WISENET 2.0 with Wikipedia relational phrases;
- We selected NELL [Carlson et al., 2010] and REVERB [Fader et al., 2011] as unlinked resources. We used KB beliefs updated to November 2014 for the former, and the set of relation instances from ClueWeb09 for the latter.

Comparative statistics in Table 6.7 show that the input KBs are rather different in nature: NELL is based on 298 predefined relations and contains beliefs for about 2 million entities. The distribution of entities over relations is however very skewed, with 80.33% of the triples being instances of the `generalizations` relationship. In contrast, REVERB contains a highly sparse relation set (1,299,844 distinct relations) and more than 3 million distinct entities. PATTY features the

largest (and, together with WISENET, sparsest) set of triples, with 1,631,531 distinct relations and less than 10 triples per relation on average.



(a)



(b)

Figure 6.10: Precision (left) and coverage (right) of disambiguated seeds at different values of δ_{dis} for (a) the whole set of triples in PATTY and (b) the subset of ambiguous triples

6.2.8.1 Disambiguation

We tested our disambiguation approach experimentally in terms of both disambiguated seed quality and overall disambiguation performance. We created a development set by extracting a subset of 6 million triples from the largest linked KB in our experimental setup, i.e. PATTY. Triples in PATTY are automatically linked to YAGO, which is in turn linked to WordNet and DBPEDIA. Since both resources are also linked by BabelNet, we mapped the original triples to the BabelNet sense inventory and used them to tune our disambiguation module. We also provide two baseline approaches: (1) direct disambiguation on individual triples with BABELFY alone (without the seeds) and (2) direct disambiguation of the seeds only (without BABELFY).

	SENSEMBED			Baseline		
ζ_{dis}	0.5-0.7	0.7-0.9	0.9-1.0	0.5-0.7	0.7-0.9	0.9-1.0
PATTY	.980	.980	1.000	.793	.780	1.000
WISENET	.958	.960	.973	.726	.786	.791
NELL	.955	.995	1.000	.800	.770	.885
REVERB	.930	.940	.950	.775	.725	.920

Table 6.8: Disambiguation precision for all KBs

	$\delta_{spec} = 0.8$		$\delta_{spec} = 0.5$		$\delta_{spec} = 0.3$	
	all	o-s	all	o-s	all	o-s
PATTY	62.15	26.60	52.49	24.06	40.75	21.41
WISENET	60.00	37.46	54.44	22.26	53.58	16.62
NELL	76.97	62.98	50.95	20.71	44.70	4.36
REVERB	41.20	38.57	25.14	23.70	13.37	12.75

Table 6.9: Coverage results (%) for all KBs considering both the "only seeds" approach alone (o-s), and combined with Babelfy (all).

We tuned our disambiguation algorithm by studying the quality of the disambiguated seeds (Section 6.2.4) extracted from the surface text triples of PATTY. Figure 6.10 shows precision and coverage for increasing values of the confidence threshold δ_{dis}. We computed precision by checking each disambiguated seed against the corresponding linked triple in the development set, and coverage as the ratio of covered triples. We analyzed results for both the whole set of triples in PATTY (Fig. 6.10a) and the subset of ambiguous triples (Fig. 6.10b), i.e. those triples whose subjects and objects have at least two candidate senses each in the BabelNet inventory. In both cases, precision of disambiguated seeds increases rapidly with δ_{dis}, stabilizing above 90% with $\delta_{dis} > 0.25$. Coverage displays the opposite behavior, decreasing exponentially with more confident outcomes, from 6 million triples to less than a thousand (for seeds with confidence $\delta_{dis} > 0.95$). As a result, we chose $\delta_{dis} = 0.25$ as optimal threshold value throughout the rest of the evaluations.

In addition, we manually evaluated the disambiguated seeds extracted from

both linked KBs (PATTY and WISENET) and unlinked KBs (NELL and REVERB). For each KB, we extracted up to three random samples of 150 triples according to different levels of confidence ζ_{dis}: the first sample included extraction with $0.5 \leq \zeta_{dis} < 0.7$, the second with $0.7 \leq \zeta_{dis} < 0.9$, and the third with $\zeta_{dis} \geq 0.9$. Each sample was evaluated by two human judges: for each disambiguated triple $\langle e_d, r, e_g \rangle$, we presented our judges with the surface text arguments e_d, e_g and the relation string r, along with the two BabelNet synsets corresponding to the disambiguated arguments s_d^*, s_g^*, and we asked whether the association of each subject and object with the proposed BabelNet synset was correct. We then estimated precision as the average proportion of correctly disambiguated triples. For each sample we compared disambiguation precision using SENSEMBED, as in Section 6.2.4, against the first baseline with BABELFY alone. Results, reported in Table 6.8, show that our approach consistently outperforms the baseline and provides high precision over all samples and KBs.

We then evaluated the overall disambiguation output after specificity ranking (Section 6.2.5) and disambiguation with relation context using BABELFY (Section 6.2.6). We analyzed three configurations of the disambiguation pipeline, namely $\delta_{spec} \in \{0.8, 0.5, 0.3\}$. We ran the algorithm over both linked and unlinked KBs of our experimental setup, and computed the coverage for each KB as the overall ratio of disambiguated triples. Results are reported in Table 6.9 and compared to the coverage obtained from the disambiguated seeds only: context-aware disambiguation substantially increases coverage over all KBs. Table 6.9 also shows that a restrictive δ_{spec} results in lower coverage values, due to the increased number of triples disambiguated without context.

	KB-UNIFY		Dutta et al.	Baseline
	all	only seeds	($\alpha = 0.5$)	
Precision	.852	**.957**	.931	.749
Recall	**.875**	.117	.799	.608
F-score	**.864**	.197	.857	.671

Table 6.10: Disambiguation results over NELL gold standard

Finally, we evaluated the quality of disambiguation on a publicly available dataset [Dutta et al., 2014] comprising manual annotations for NELL. This dataset provides a gold standard of 1,200 triples whose subjects and objects are manually assigned a proper DBpedia URI. We again used BabelNet's inter-resource links to express DBpedia annotations with our sense inventory and then sought, for each annotated triple in the dataset, the corresponding triple in our disambiguated

version of NELL with $\delta_{dis} = 0.25$ and $\delta_{spec} = 0.8$. We then repeated this process considering only the disambiguated seeds instead of the whole disambiguation pipeline. In line with [Dutta et al., 2014], we computed precision, recall and F-score for each setting. Results are reported in Table 6.10 and compared against those of [Dutta et al., 2014] and against our first baseline with BABELFY alone. KB-UNIFY achieves the best result, showing that a baseline based on state-of-the-art disambiguation is negatively affected by the lack of context for each individual triple. In contrast, an approach that relies only on the disambiguated seeds affords very high precision, but suffers from dramatically lower coverage.

6.2.8.2 Specificity Ranking

We evaluated the specificity ranking (Section 6.2.5) generated by KB-UNIFY for all KBs of our experimental setup. First of all, we empirically validated our scoring function $Gen(r)$ over each resource: for each relation we computed the average cosine similarity among all its domain arguments $\bar{s}_D$ and among all its range arguments $\bar{s}_G$. We then plotted the average $\bar{s}$ of $\bar{s}_D$ and $\bar{s}_G$ against $Gen(r)$ for each relation r (Figure 6.11). The resulting trend seems to confirm our intuition, introduced in Section 6.2.5, since the average similarity among domain and range arguments decreases for increasing values of $Gen(r)$, indicating that more general relations allow less semantically constrained subject-object types.

	NELL	REVERB	PATTY	WISENET
Precision	.660	.715	.625	.750
Cohen's kappa	-	.430	.620	.600

Table 6.11: Specificity ranking evaluation

We then used human judgement to assess the quality of our specificity rankings. First, each ranking was split into four quartiles, and two human evaluators were presented with a sample from the top quartile (i.e. a relation falling into the most general category) and a sample from the bottom quartile (i.e. a relation falling into the most specific category). We shuffled each relation pair, showed it to our human judges, and then asked which of the two relations they considered to be the more specific. Ranking precision was computed by considering those pairs where human choice agreed with the ranking. Finally, we computed inter-annotator agreement on each specificity ranking (except for NELL, due to the small sample size) with Cohen's kappa coefficient [Cohen, 1968]. Results for each ranking are reported in Table 6.11, while some examples of general and specific relations for each KB are shown in Table 6.12. Disagreement between

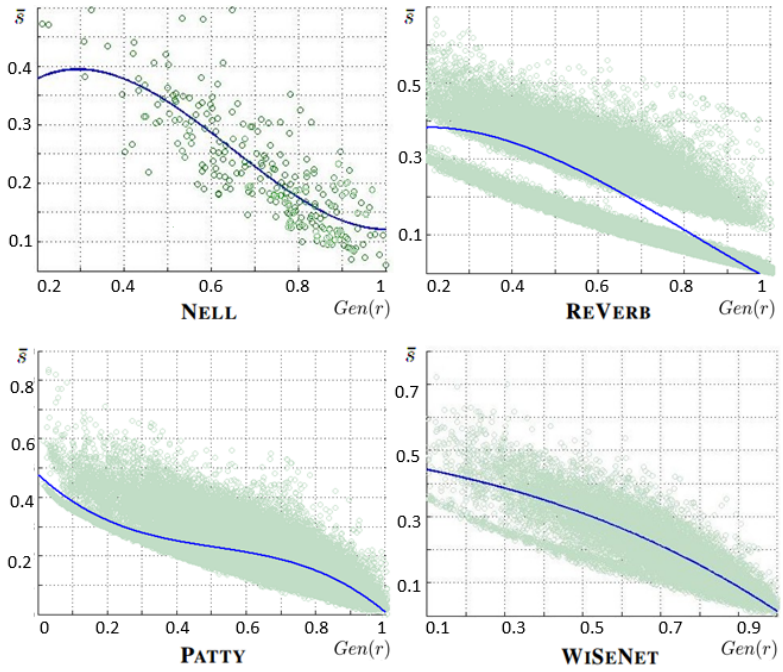


Figure 6.11: Average argument similarity against $Gen(r)$

human choice and ranking is higher in NELL (where the set of relations is quite small compared to other KBs) and in PATTY (due to a sparser set of relations, biased towards very specific patterns). Inter-annotator agreement is instead lower for REVERB, where unconstrained Web harvesting often results in ambiguous relation strings.

6.2.8.3 Alignment

Due to the novelty of our approach, and hence the lack of widely accepted gold standards and testbeds, we evaluated our cross-resource relation alignment algorithm (Section 6.2.7) by exploiting human judgement once again. Given the results of Section 6.2.8.1, we considered the top 10k frequent relations for each KB and ran the algorithm over each possible pair of KBs with two different configurations: $\delta_{align} = 0.7$ and $\delta_{align} = 0.9$. From each pair of KBs $\langle KB_i, KB_j \rangle$ we obtained a list of candidate alignments, i.e. pairs of relations $\langle r_i, r_j \rangle$ where $r_i \in KB_i$ and $r_j \in KB_j$.

From each list we then extracted a random sample of 150 candidate alignments. We showed each alignment[11] $\langle r_i, r_j \rangle$ to two human judges, and asked

[11] In the case of relation synsets, such as PATTY and WISENET, we selected up to three random relation strings from each synset.

	NELL
High $Gen(r)$	`agent created` `at location`
Low $Gen(r)$	`person in economic sector` `restaurant in city`
	REVERB
High $Gen(r)$	`is for` `is in`
Low $Gen(r)$	`enter Taurus in` `carry oxygen to`
	PATTY
High $Gen(r)$	`located in` `later served to`
Low $Gen(r)$	`starting pitcher who played` `league coach for`
	WISENET
High $Gen(r)$	`include` `is a type of`

Low $Gen(r)$	`lobe-finned fish lived during`
	`took part in the Eurovision`
	`contest`

Table 6.12: Examples of general and specific relations for all KBs

whether r_i and r_j represented the same relation. The problem was presented in terms of paraphrasing: for each pair, we asked whether exchanging r_i and r_j within a sentence would have changed that sentence's meaning. In line with Section 6.2.8.2 we computed precision based on the agreement between human choice and automatic alignments. Results are reported in Table 6.13. Our alignment algorithm shows high precision in all pairings where $\delta_{align} = 0.9$. Alignment reliability decreases for lower δ_{align}, as relation pairs where r_i is a generalization of r_j (or vice versa) tend to have similar centroids in V_S. The same holds for pairs where r_i is the negation of r_j (or vice versa). Even though we could have utilized measures based on relation string similarity [Dutta et al., 2015] to reduce wrong

	PATTY-WISENET		PATTY-REVERB		NELL-REVERB	
δ_{align}	0.7	0.9	0.7	0.9	0.7	0.9
Prec.	.68	**.80**	.58	**.74**	.61	**.75**
# Align.	128k	1.2k	47k	643	2.6k	88
	PATTY-NELL		WISENET-NELL		WISENET-REVERB	
δ_{align}	0.7	0.9	0.7	0.9	0.7	0.9
Prec.	.66	**1.00**	.70	**.84**	.59	**.87**
# Align.	2.6k	57	381	34	9.9k	169

Table 6.13: Cross-resource alignment evaluation

alignments in these cases, by relying on a purely semantic criterion we removed any prior assumption on the format of input KBs. Some examples of alignments are shown in Table 6.14.

To conclude, we report statistics regarding the unified **KB*** produced from the initial set of resources in our experimental setup (cf. Section 6.2.8). We validated our thresholds for high-precision, and selected $\delta_{dis} = 0.25$, δ_{spec} = 0.8 and δ_{align} = 0.8. Our alignment algorithm produced 56,673 confident alignments, out of which 2,207 relation synsets were derived, with an average size of 16.82 individual relations per synset. As a result, we obtained a unified **KB*** comprising 24,221,856 disambiguated triples defined over 1,952,716 distinct entities and 2,675,296 distinct relations.

6.2.9 Conclusion

In this section, we described KB-UNIFY, a novel, general approach for disambiguating and seamlessly unifying KBs produced by different OIE systems. KB-UNIFY represents entities and relations using a shared semantic representation, leveraging a unified sense inventory together with a semantically-enhanced vector space model and a disambiguation algorithm. This enables us to disambiguate unlinked resources (like NELL and REVERB) with high precision and coverage, and to discover relation-level cross-resource alignments effectively. One of the key features of our strategy is its generality: by representing each KB on a common ground, we need no prior assumption on the nature and format of the knowledge it encodes. We tested our approach experimentally on a set of four very different KBs, both linked and unlinked, and we evaluated disambiguation and alignment results extensively at every stage, exploiting both human evaluations and public

PATTY-WISENET		ζ_{align}
portrayed	's character	0.84
debuted in	first appeared in	0.86
PATTY-REVERB		ζ_{align}
language in	is spoken in	0.81
mostly known for	plays the role of	0.70
NELL-REVERB		ζ_{align}
bookwriter	is a novel by	0.88
personleadscity	is the mayor of	0.60
NELL-PATTY		ζ_{align}
worksfor	was hired by	0.72
riveremptiesintoriver	tributary of	0.89
NELL-WISENET		ζ_{align}
animaleatfood	feeds on	0.72
teamhomestadium	play their home games at	0.88
REVERB-WISENET		ζ_{align}
has a selection of	offers	0.82
had grown up in	was born and raised in	0.85

Table 6.14: Examples of cross resource relation alignments

gold standard datasets (when available).

6.3 ColWordNet: Extending WordNet with Fine-Grained Collocational Information via Supervised Distributional Learning

The importance of WordNet as a key knowledge enabler is indisputable. In this dissertation we have where WordNet played a key role. Given its usefulness, but also its inherent limitations, there is a strong line of research in NLP concerned

with the improvement of WordNet. One of its main drawbacks is that it is not updated frequently, and thus it omits many lemmas and senses, such as those from domain specific lexicons, creative slang usages, or those for technology or entities that came into recent existence (e.g., selfie, mp3) [Jurgens and Pilehvar, 2016]. Another limitation is that it does not account for collocational information, i.e., idiosyncratic binary lexical co-occurrences. As a standalone research topic, collocations have been the focus of a substantial amount of work, e.g. for automatically retrieving them from corpora [Choueka, 1988, Church and Hanks, 1990, Smadja, 1993, Kilgariff, 2006, Evert, 2007, Pecina, 2008, Bouma, 2010, Gao, 2013], and for their semantic classification according to different typologies [Wanner et al., 2006, Gelbukh and Kolesnikova., 2012, Moreno et al., 2013, Wanner et al., 2016]. However, to the best of our knowledge, no previous work attempted the automatic enrichment of WordNet with collocational information. The only related attempt consisted in designing a schema for the manual inclusion of lexical functions from Explanatory Combinatorial Lexicology (ECL) [Mel'čuk, 1996] into the Spanish EuroWordNet [Wanner et al., 2004]. We propose to bridge this gap by introducing **ColWordNet** (**CWN**), an automatic extension of WordNet with collocational information, which is obtained thanks to leveraging sense-level embeddings and the semantic relations holding between the two components of a collocation, namely the base and the collocate (e.g. *rain* and *heavy* respectively, in the collocation "heavy rain").

6.3.1 Background

Collocations are restricted lexical co-occurrences of two syntactically related lexical items, the base and the collocate. In a collocation, the base is freely chosen by the speaker, while the choice of the collocate depends on the base; see, e.g., [Cowie, 1994, Mel'čuk, 1996, Kilgariff, 2006] for a theoretical discussion. For instance, in the collocations *take* [*a*] *step*, *solve* [*a*] *problem*, *pay attention*, *deep sorrow*, and *strong tea*, the bases are *step*, *problem*, *attention*, *sorrow* and *tea*, and *take*, *solve*, *pay*, *deep* and *strong* are their respective collocates.

Besides a syntactic dependency, between the base and the collocate a semantic relation holds. Some of these semantic relations, such as 'intense', 'weak', 'perform', 'cause', etc. can be found across a large number of collocations. For instance, an 'intense' *applause* is a *thundering applause*, an 'intense' *emotion* is *deep*, 'intense' *rain* is *heavy*, and so on. In our experiments, we focused on a subset of eight prominent semantic collocation relations (or categories), which are listed in the first column of Table 6.15. These semantic categories are a generalization of the *lexical functions* (LFs) from ECL already used in [Wanner et al., 2004]. We have decided to use somewhat more generic categories instead of LFs because, on the one hand, some of the LFs differ only in terms of their syntactic structure

(i.e. they capture the same semantic relation), and, on the other hand, LFs pose a great challenge for annotation due to their syntactic granularity (for example, certain semantic relations such as 'perform' may be further distinguished across three different finer-grained LFs depending on their role in a surface-syntactic representation of the sentence). To sum up our linguistic motivation, let us note that in the specific set of categories we address in this experiment, we aim at capturing two of the most studied groups of LFs, namely intensifiers (noun+adjective, as in 'deep' *commitment*) and collocations triggered by semi-auxiliary verbs (also called suppport or light verbs), such as 'give' [an] *order* [Melčuk, 1998].

As for external resources, in our work we take advantage of BABELNET and SENSEMBED. We used SENSEMBED for automatically disambiguating our training data, and as our **bases model**; and (3) the SW2V (*Senses and Words to Vectors*) vector space model [Mancini et al., 2016] as our **collocates model**. The SW2V vector space is moeled with a source web corpus of 3 billion words [Han et al., 2013],[12]. Similarly to SENSEMBED, this model is based on a predisambiguation of text corpora using BabelNet as sense inventory. However, unlike SENSEMBED, which learns vector representations for individual word senses, this provides fine-grained information in the form of both plain text words and synsets in a shared vector space. We used the algorithm of [Mancini et al., 2016][13] for training word and synset embeddings in the same vector space. This approach modifies the objective function of Word2Vec[14] so that words and senses can be learned jointly in a single training.

6.3.2 Methodology

In this section, we provide a detailed description of the algorithm behind the construction of CWN. The system takes as input the WordNet lexical database and a set of collocation lists pertaining to predefined semantic categories, and outputs CWN. First, we collect training data and perform automatic disambiguation. Then, we use this disambiguated data for training a linear *transformation matrix* from the base vector space, i.e., SENSEMBED, to the collocate vector space, i.e., SHAREDEMBED. Finally, we exploit the WordNet taxonomy to select input base collocates to which we apply the transformation matrix in order to obtain a sorted list of candidate collocates.

[12] ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/

[13] Available at `http://lcl.uniroma1.it/sw2v`

[14] We used the Continous Bag-Of-Words (CBOW) model with standard hyperparameters: 300 dimensions and a window size of 8 words.

Sem. Category	Example	# instances
'intense'	*absolute certainty*	586
'weak'	*remote chance*	70
'perform'	*give chase*	393
'begin to perform'	*take up a chase*	79
'increase'	*improve concentration*	73
'decrease'	*limit* [*a*] *choice*	73
'create', 'cause'	*pose* [*a*] *challenge*	195
'put an end'	*break* [*the*] *calm*	79

Table 6.15: CWN semantic categories and size of training set

6.3.2.1 Disambiguation and Training

As is common in previous work on semantic collocation classification [Moreno et al., 2013, Wanner et al., 2016], our training set consists of a list of manually annotated collocations. For this purpose, we randomly selected nouns from the Macmillan Dictionary and manually classified their corresponding collocates with respect to their semantic categories.[15] Note that there may be more than one collocate for each base. Since collocations with different collocate meanings are not evenly distributed in language (e.g., we may tend to use more often collocations conveying the idea of 'intense' and 'perform' than 'begin to perform'), the number of instances per category in our training data also varies significantly (see Table 6.15).

Our training dataset consists at this stage of pairs of plain words, with the inherent ambiguity this gives raise to. We surmount this problem by applying a disambiguation strategy based on the notion that, from all the available senses for a collocation's base and collocate, their correct senses are those which are most similar. We follow the **L2S** disambiguation strategy (cf. Section 3.2.2). Let us recall that the main idea of this strategy is to, for a given pair of semantically related words, obtain the set of associated SensEmbed vectors to each word, and then assign the pair of vectors that maximizes cosine similarity between them.

This disambiguation process yields a set of disambiguated pairs **D**, where each pair is denoted as $\langle v'_b, v'_c \rangle$, which constitutes the input for the next module of the pipeline, the learning of a *transformation matrix* aimed at retrieving WordNet

[15] We do not consider phrasal verb collocates, e.g. *stand up*, *give up* or *calm down*.

synset collocates for any given WordNet synset base. In a similar fashion as in Section 4.2.3.3, we learn a linear transformation from v'_b to v'_c, aiming at reflecting an inherent condition of collocations. Since collocations are a linguistic phenomenon that is more frequent in the narrative discourse than in formal essays, they are less likely to appear in an encyclopedic corpus (recall that SENSEMBED vectors, which we use, are trained on the English Wikipedia). This motivates the use of SENSEMBED (denoted as $\mathcal{S}$) as ou *base space*, and our SHAREDEMBED $\mathcal{X}$ as the *collocate model*, as it was trained over more varied language such as blog posts or news items.

Then, we construct our linear transformation model as follows: For each disambiguated collocation $\langle l'_b, l'_c \rangle \in \mathbf{D}$, we first retrieve the corresponding base vectors v'_b. Next, we exploit the fact that $\mathcal{X}$ contains both BabelNet synsets and words, and derive for each l'_c two items, namely the vectors associated to its lexicalization (word-based) and its BabelNet synset. For example, for the training pair $\langle \texttt{ardent\_bn:00097467a}, \texttt{desire\_bn:00026551n} \rangle \in \mathbf{D}$, we learn two linear mappings, namely $\texttt{ardent\_bn:00097467a} \mapsto \texttt{desire}$ and $\texttt{ardent\_bn:00097467a} \mapsto \texttt{bn:00026551n}$. We opt for this strategy, which doubles the size of the training data in most lexical functions (depending on coverage), due to the lack of resources of manually-encoded classification of collocations. By following this strategy we obtain an extended training set $\mathbf{D}* = \{\vec{b}_i, \vec{c}_i\}_{i=1}^{n}$ ($\vec{b}_i \in \mathcal{X}$, $\vec{c}_i \in \mathcal{S}, n \geq |\mathbf{D}|$). Then, we construct a *base matrix* $\mathbf{B} = \left[\vec{b}_1 \dots \vec{b}_n\right]$ and a collocate matrix $\mathbf{C} = [\vec{c}_1 \dots \vec{c}_n]$ with the resulting set of training vector pairs. We use these matrices to learn a linear transformation matrix $\Psi \in \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{X}}$, where $d_\mathcal{S}$ and and $d_\mathcal{X}$ are, respectively, the number of dimensions of the base vector space (i.e., SENSEMBED) and the collocate vector space (SHAREDEMBED).[16] The transformation matrix is learned by minimizing the least squared distance between each base and collocate pair, similar as the approach we described in Section 4.2.3.3.

Having trained Ψ, the next step of the pipeline is to apply it over a subset of WordNet's base concepts and their hyponyms. For each synset in this branch, we apply a scoring and ranking procedure which assigns a *collocates-with* score. If such score is higher than a predefined threshold, tuned over a development set, this relation is included in CWN.

6.3.2.2 Retrieving and Sorting WordNet Collocate Synsets

During the task of enriching WordNet with collocational information, we first gather a set of base WordNet synsets by traversing WordNet hypernym hierarchy starting from those base concepts that are most fit for the input semantic cate-

[16] In our setting the numbers of dimensions are $d_\mathcal{S} = 400$ and $d_\mathcal{X} = 300$.

gory[17]. Then, the *transformation matrix* is used to find candidate WordNet synset collocates (mostly verbs or adjectives) for each base WordNet synset.

WordNet synsets are mapped to BabelNet synsets, which in turn map to as many vectors in SENSEMBED as their associated lexicalizations. Formally, given a base synset b, we apply the transformation matrix to all the SENSEMBED vectors $\mathbf{V}_b = \{v_b^1, ..., v_b^n\}$ associated with its lexicalizations. For each $v_b^i \in \mathbf{V}_b$, we first get the vector $\psi_b^i = v_b^i \Psi$ obtained as a result of applying the transformation matrix and then we gather the subset $W_b^i = w_b^1 \dots w_b^{10} (\mathrm{w}_b^j \in \mathcal{X})$ of the top ten closest vectors by cosine similarity to ψ_b^i in the SHAREDEMBED vector space $\mathcal{X}$. Each $\vec{w}_b^{i,j}$ is ranked according to a scoring function $\lambda(\cdot)$, which is computed as follows[18]: $\lambda(w_b^{i,j}) = \frac{cos(\psi_b^i, w_b^{i,j})}{j}$. This scoring function takes into account both the cosine similarity as well as the relative position[19] of the candidate collocate with respect to other neighbors in the vector space. Apart from sorting the list of candidate collocates, this scoring function is also used to measure the confidence of the retrieved collocate synsets in CWN.

6.3.3 Evaluation

We evaluate CWN both intrinsically and extrinsically. Our intrinsic evaluation consists of a manual scoring of the correctness of the newly introduced relations. Extrinsic evaluation assesses the quality of CWN as an input resource for introducing collocational information into a word embeddings model.

6.3.3.1 Intrinsic: Precision of Collocate Relations

Sampling and evaluation are carried out as follows. First, for each semantic category, we retrieve 50 random bases included in the aforementioned base concepts and their hyponym branch. This results in an evaluation set $Test$ of 800 collocations, as for each base we retrieve the 5 highest scoring candiates. These collocations are evaluated in terms of correctness, i.e., if the associated synset is an appropriate collocate for the input base. Note that not all bases in the test set may be suitable for the given semantic category, and that is why we also perform an evaluation on the test data restricted to only those bases manually selected for being suitable for having at least one collocate. We denote the restricted test data as $Test*$. For example, the base synset `putt.n.01` defined as *hitting a golf ball*

[17]These are: For 'intense' and 'weak', `attitude.n.01`, `feeling.n.01` and `ability.n.02`. For the rest of them, we select `cognition.n.01`, `act.n.02` and `action.n.01`.

[18]If w_b^j appears in a different W_b^j set ($j \neq i$), its scores are averaged.

[19]Position is arguably an important factor as there may be dense areas where cosine similarity alone may not reflect entirely the fitness of a candidate.

that is on the green using a putter does not admit any 'decrease' collocate, and therefore its collocations are not considered in $Test*$.

Since our algorithm returns a list of candidate collocate synsets for an input base synset, the task naturally becomes that of a ranking problem, and therefore ranking metrics such as Precision@K (P@K), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) are appropriate for evaluating this experiment, in a very similar fashion as the experimental setup of our TAXOEMBED approach for hypernym discovery, described and evaluated in Chapter 4. These measures provide insights on e.g. how often valid collocates were retrieved in the first positions of the rank (MRR), and if there were more than one valid collocate, whether this set was correctly retrieved, (MAP and R-P). In Table 6.16 we provide a detailed summary of the performance of our system (CWN), as compared with a competitor unsupervised baseline which exploits word analogies (as in $\vec{man} - \vec{king} + \vec{woman} = \vec{queen}$) [Rodríguez-Fernández et al., 2016]. This baseline, which we deploy on the SHAREDEMBED space, takes as input a prototypical collocation of a given semantic category (e.g. *thunderous applause* for 'intense') and an input base, and collects the top 10 Nearest Neighbours (NNs) to the vector resulting of the aforementioned analogy operation. Due to the difficulty of the task, and the restriction it imposes for collocates to be disambiguated synsets rather than any text-based word, the unsupervised approach fails short when compared to our supervised method, which is capable to find more and better disambiguated collocates.

Note that for half of the semantic categories under evaluation, our approach correlated well with human judgement, with the highest ranking candidates being more often correct than those ranked lower. This is the case of 'put an end', 'decrease', 'create/cause' and 'weak'. In fact, it is in 'put an end', where our system achieves the highest MRR score, which we claim to be the most relevant measure, as it rewards cases where the first ranked returned collocation is correct without measuring in the retrieved collocates at other positions. Moreover, let us highlight the importance of two main factors. First, the need for a well-defined semantic relation between bases and collocates. It has been shown in other tasks that exploit linear transformations between embeddings models that even for one single relation there may be clusters that require certain specificity accounting, for instance, for the *domain* or underlying *semantics* of the data [Fu et al., 2014]. Second, the importance of having a reasonable amount of training pairs so that the model can learn the idiosyncrasies of the semantic relation that is being encoded (e.g., [Mikolov et al., 2013b] report a major increase in performance as training data increases in several orders of magnitude). This is reinforced in our experiments, where we obtain the highest MAP score for 'intense', the semantic category for which we have the largest training data available.

	'intense'				**'perform'**				**'put an end'**				**'increase'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.35	**0.46**	0.15	0.16	0.20	**0.36**	0.05	0.08	0.15	**0.50**	0.05	0.14	0.15	**0.42**
P@5	0.03	0.30	0.43	**0.57**	0.06	0.06	0.13	**0.23**	0.02	0.03	0.12	**0.40**	0.04	0.11	0.18	**0.51**
MRR	0.05	0.41	0.48	**0.65**	0.18	0.19	0.32	**0.59**	0.07	0.12	0.20	**0.68**	0.07	0.21	0.22	**0.65**
MAP	0.05	0.45	0.48	**0.64**	0.15	0.18	0.32	**0.59**	0.07	0.12	0.19	**0.64**	0.07	0.20	0.22	**0.64**
	'decrease'				**'create/cause'**				**'weak'**				**'begin to perform'**			
	Baseline		CWN		Baseline		CWN		Baseline		CWN		Baseline		CWN	
	Test	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**	*Test*	*Test**
P@1	0.00	0.00	0.30	**0.46**	0.05	0.16	0.10	**0.50**	0.00	0.00	0.10	**0.22**	0.00	0.00	0.00	0.00
P@5	0.02	0.03	0.19	**0.29**	0.04	0.13	0.04	**0.20**	0.02	0.03	0.04	**0.08**	0.03	0.07	0.02	**0.20**
MRR	0.02	0.04	0.39	**0.61**	0.07	0.25	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**
MAP	0.02	0.03	0.38	**0.58**	0.06	0.20	0.10	**0.50**	0.03	0.04	0.01	**0.22**	0.05	0.12	0.04	**0.41**

Table 6.16: Comparative evaluation of CWN and an unsupervised baseline.

6.3.3.2 Extrinsic evaluation: Retrofitting Vector Space Models to CWN

We complement our manual evaluation with an extrinsic experiment, where we assess the extent to which our newly generated lexical resource can be used to *introduce collocational sensitivity* to a generic word embeddings model, that is, to draw closer in the space vectors that are related by collocational information of a certain type, as defined in a complementary lexicon or semantic resource[20]. To this end, we extract collocation clusters by extracting all the synsets associated lemmas (e.g. for *heavy.a.01 rain.n.01*, we would extract the cluster [*heavy*, *rain*, *rainfall*]). These are used as input for the Retrofitting algorithm [Faruqui et al., 2015][21]. This algorithm takes as input a vector space and a semantic lexicon which may encode any semantic relation, and puts closer in the vector space words that are related in the lexicon.

Previous approaches have encoded semantic relations by introducing some kind of bias into a vector space model [Yu et al., 2015, Pham et al., 2015, Mrkšić et al., 2016, Nguyen et al., 2016]. For instance, [Yu et al., 2015] encode term-hypernym relations by grouping together terms and their hypernyms, rather than semantically related items. In this way, their *biased* model puts closer to *jaguar* terms like *animal* or *vehicle*, while an unbiased model would put nearby terms such as *lion*, *bmw* or *jungle*. We aim at introducing a similar bias, but in terms of collocational information. This is achieved, for each lexical function and each

[20]We use the Google News pre-trained Word2Vec vectors, available at `code.google.com/archive/p/word2vec/`, as input for retrofitting.

[21]We used the code available at `https://github.com/mfaruqui/retrofitting`

	'intense'			**'weak'**			**'perform'**			**'create/cause'**		
	correct	*dist.*	*diff.*	*correct*	dist.	*diff.*	*correct*	*dist.*	*diff.*	*correct*	*dist.*	*diff.*
original	0.22	0.04	+0.18	0.17	0.05	+0.12	0.15	0.05	+0.10	0.17	0.06	+0.11
retrofitted	0.27	0.06	**+0.21**	0.19	0.06	**+0.13**	0.25	0.11	**+0.14**	0.28	0.12	**+0.16**

Table 6.17: Comparison of collocational sensitivity between original and retrofitted embeddings models over four semantic categories.

synset in CWN, by obtaining its top 3 collocate candidates and incorporate information on their *collocationality* into the model.

6.3.3.2.1 Collocational Sensitivity In this experiment, we assess the extent to which a retrofitted model with collocational bias is able to discriminate between a correct collocation and a random combination of the same base with an unrelated collocate. To this end, we manually constructed two datasets, one for *noun+adjective* ('intense' and 'weak' semantic categories) and one for *noun+verb* combinations, which we evaluate on the two most productive semantic categories, namely 'perform' and 'create/cause'. These datasets consist of 50 bases, each base with one correct collocate according to the Macmillan Collocations Dictionary, accompanied by four *distractor* (*dist.* in Table 6.17) collocates. For instance, given the correct 'perform' collocation *make a pledge*, we expect our 'perform'-wise retrofitted model to increase the score in $\vec{make} + \vec{pledge}$ substantially more than a combination $\vec{distractor} + \vec{pledge}$. For each evaluated semantic category, we computed the average increase of the cosine similarity between all correct collocations and all distractors (*diff.* in Table 6.17). As shown in Table 6.17, there is a consistent increase over the four evaluated semantic categories, namely 'intense', 'weak', 'perform' and 'create/cause'. This proves the potential of our retrofitted model to discern between correct and wrong collocates. In the following section, we explore the possibility to use this vector space for finding collocates giving a base as input.

6.3.3.2.2 Exploring Nearest Neighbours for Collocate Discovery Inspired by Yu et al.'s work on introducing hypernymic bias into a word embeddings model, we explore the extent to which our retrofitted models can be used to discover *alternative collocates* given the composition of the words involved in a collocation as input. In order to discover these collocates, we compose the base and the collocate by averaging their respective word embeddings and retrieve its closest words in the vector space according to cosine similarity. In Table 6.18, we show a sample of five NNs for several input adjective+noun collocations. These

	'intense'			'weak'	
	original	retrofitted		original	retrofitted
ferocious + hatred	**vicious**	fierce	dim +light	bright	**faint**
	fury	**fearsome**		dimmed	**unaccented**
	ferocity	fury		dimmer	**dense**
	savage	hate		dimming	bright
	hostility	**savage**		lights	centaur
intense + sympathy	fierce	**considerable**	mild + comment	milder	**modest**
	empathy	**tremendous**		NamedEntity	**meek**
	admiration	**enormous**		NamedEntity	NamedEntity
	anger	encouragement		NamedEntity	NamedEntity
	grudging respect	**immense**		NamedEntity	NamedEntity
sheer + delight	amazement	**immense**	modest + progress	progress	**mild**
	sheer unadulterated	**colossal**		pro gress	**meek**
	sheer joy	delectation		**Modest**	dissatisfaction
	joy	disgust		NamedEntity	progess
	astonishment	**stupendous**		strides	**slight**

Table 6.18: Comparison of the five NNs of six sample adj+noun collocations between a generic word embeddings model and a *retrofitted* version with semantic collocation information ('intense' and 'weak'). Note the increase in plausible collocates in retrofitted models (in bold). NamedEntity refers to noisy entities appearing among the top 5 NNs.

examples reveal how the vector space model retroffited using our collocations tends to bring closer in the space modifiers (i.e., collocates), providing an interesting method for automatic collocation discovery. Despite its simplicity, this collocational discovery approach extracts a considerable amount of suitable fine-grained collocates for a given base. For example, given the collocation *intense sympathy*, the retrofitted space extracts *considerable*, *tremendous*, *enormous* and *immense* as candidate collocates of intensity among the five nearest neighbours. As future work we plan to further exploit and evaluate the impact of this property.

6.3.4 Conclusions and Future Work

We have described a system for an automatic enrichment of the WordNet lexical database with fine-grained collocational information, yielding a resource called ColWordNet (CWN). Our approach is based on the intuition that there is a linear transformation in vector spaces between bases and collocates of the same *semantic category*, e.g. between *heavy* and *rain*, or between *ardent* and *desire*. We have exploited sense-based embedding models to train an algorithm designed to retrieve valid collocates for a given input base. This pipeline is carried out at the *sense* level (rather than the word level), by leveraging models which use BabelNet as a reference sense inventory. We evaluated CWN both intrinsically and extrinsi-

cally, and verified that our algorithm is able to encode fine-grained *collocates-with* relations at synset level.

6.4 Savana: Enriching the Spanish Snomed via Dependency Parsing and Distributional Semantics

Up until this far, we have presented a number of methods for formalizing knowledge in various forms (definitions, hypernymic relations, and lexical taxonomies), and have explored three approaches for improving knowledge representation both in a general (KB-U), and a specific (MKB) sense. We have also targeted a specific semantic relation in CWN. We complement these contributions with an experiment which poses two major challanges. First, it is carried out in a language other than English, which results in less availability of resources and software. And second, it aims at improving a medical terminological database, which due to the specificity and idiosyncrasy of the domain, makes it more difficult than modeling other less variable domains.

Thus, in this section we present the first approach (to the best of our knowledge) that attempts to extend the Spanish version of the Snomed Clinical Terms medical terminology [Spackman et al., 1997].

6.4.1 Motivation and Background

Among the many fields of knowledge that are sensitive to the dramatic changes ignited by the advent of the Information Age, the medical domain is probably one of the most prolific. There is a considerable amount of research focused on aggregating knowledge contained in medical research papers [Giuliano et al., 2006, Rindflesch et al., 2000, Pustejovsky et al., 2001, Subramaniam et al., 2003, Donaldson et al., 2003], also in Spanish [Bedmar et al., 2008, Gálvez, 2012], and machine learning is gaining popularity as well in the medical field as decision support tools[22]. In this context, the scenario of a comprehensive Evidence-Based Medicine [Kumar, 2011] seems to be more plausible than ever. However, one of the great challenges for enabling data-driven support to clinical decisions is making sense of unstructured information appearing in Clinical Health Records (CHR). These are documents where doctors take notes on a patient's medical condition, his or her progress, and suggest possible medication and treatment. They are a rich source of information because they provide personalized empirical data on treat-

[22] see e.g. the recent sucessful case of Watson in predicting cancer's treatments `https://futurism.com/ibms-watson-ai-recommends-same-treatment-as-doctors-in-99-of-cancer-cases/`

ment and evolution of medical conditions, and hence this type of document is receiving interest from the NLP community as enabler not only for medical support systems but also for its potential as training and evaluation data for Machine Learning algorithms in the field of Bioinformatics.

Examples of the interaction between NLP and CHRs include MEDEX [Xu et al., 2010], a system for extracting medication information from clinical narratives, or a system for drug reaction event extraction [Santiso et al., 2016]. However, and despite their potential, CHRs pose great challenges for automatic processing, as they are often unstructured, ill-defined and arduous to analyse at scale [Iqbal et al., 2015].

In this context, Medical Terminological Databases (MTDs) play a crucial role, as they provide a structured ground where medical concepts and their relations are encoded by medical experts and can be used as a benchmark for developing algorithms that leverage medical concept extraction to some extent. One of the best known MTDs is SnomedCT, which is part of the Unified Medical Language System (UMLS) [Bodenreider, 2004]. One of the main drawbacks of MTDs is that creating and maintaining them manually is arduous. More importantly, keeping them up to date is not possible considering the amount of novel information that is generated daily. Furthermore, even if they are manually created, there is certain discussion even on their quality, since it is difficult to control the fitness of every single addition to the database [Morrey et al., 2009] (cf. Chapter 1).

In this chapter, we propose to bridge the gap between unstructured medical knowledge stored arbitrarily in CHRs, on one hand, and the automatic maintaining of MTDs, on the other. In the remainder of this paper, we first describe SAVANA, a Biomedical Information Extraction system, which we run on a large collection of CHRs. In a second phase, SAVANA's predictions are presented to medical practitioners, who validate novel associations between SnomedCT concepts and their lexicalizations (i.e. the way they are expressed in free text). We exploit the combination of SAVANA and the validation stage to obtain a validation dataset of nearly 500 novel medical terms in Spanish, on which we evaluate several unsupervised systems aimed at finding, for each candidate novel term, its best point of attachment in the Spanish SnomedCT Database. These systems are based on both syntactic and semantic properties. Our results suggest that this is a promising direction for performing large-scale medical terminology extraction for Spanish, along with its *semantification*.

6.4.1.1 Brief background of MTDs

The availability of MTDs is in constant growth. Examples range from well-established collaborative efforts like UMLS [Bodenreider, 2004], umbrella terminologies for multilingual resources such as SnomedCT [Spackman et al., 1997],

or even the CIE database (*Clasificación Internacional de Enfermedades*), published by the Panamerican Health Organization (*Organización Panamericana de la Salud*). In addition, general purpose resources are increasingly playing more important roles in biomedical NLP tasks, as is the case of Wikipedia, which has been exploited for identifying medical disorders in CHRs [Bodnari et al., 2013].

The medical domain has also received attention in terms of automatically expanding existing resources. Prominent examples include (1) The development of novel MTDs from Wikipedia [Pedro et al., 2008]; (2) Enriching SnomedCT terminology with associated definitions [Ma and Distel, 2013]; and in multilingual settings, (3) Expansion of SnomedCT in Swedish by processing CHRs [Henriksson et al., 2013].

6.4.2 Savana

We use SAVANA, a Biomedical Information Extraction System[23], integrated in several public and private healthcare institutions in Spain, for obtaining and validating ground truth data. The SAVANA algorithm is designed to retrieve prominent biomedical information from CHRs in the Spanish language. It does so by combining in its pipeline modules for, among others, sentence segmentation, tokenization, spell checking, acronym detection and expansion, negation identification, and a multi-dimensional ranking scheme which combines linguistic knowledge, statistical evidence, and state-of-the-art continuous vector representations of words and documents in the biomedical domain learned via shallow neural networks. We run SAVANA over several thousand CHRs, and ask medical practitioners to validate matches of SAVANA's association between a mention of a medical concept in text, and an existing SnomedCT entry, by means of a web interface (Figure 6.12). The subset of the Spanish SnomedCT branch on which we run our experiments contains over 401,126 concepts, which are linked by means of 2,722,877 hypernymic (is-a) relations. The validation procedure may yield *novel terminology* in terms of either novel lexicalizations for an existing term (synonyms), or novel terms which can be attached to a more general SnomedCT concept (hyponyms). In this experiment, we are interested in the latter case: Finding the best point of attachment for novel concepts, rather than finding additional ways of expressing the same idea. At validation stage, if human experts consider that a concept identified by SAVANA has a meaning which is missing in SnomedCT, this concept makes it to our ground truth novel terminology, and hence will constitute the testbed for the experiments we describe in Section 6.4.3. We collect gold standard data of up to 492 novel terms, with an average of 3.2 hypernymic relations encoded by human experts. There was no restriction in the type of concept

[23] http://www.savanamed.com/

Figure 6.12: A snapshot of the validation web interface. Let us highlight how the validation procedure allows the medical expert to assign to the SnomedCT concept `departamento de pediatría`, a novel lexicalization in the context of CHRs, namely the string `su pediatra`.

to be included. Therefore, this dataset includes diverse terms which are related to infrastructure, e.g. SERVICIO DE ODONTOLOGÍA $\rightarrow$ {servicio hospitalario}[24], or actual medical conditions, e.g. GONARTROSIS$\rightarrow$ {trastorno de la rodilla, enfermedad de la rodilla}. In the following section, we describe the experiments carried out to discover the most appropriate hypernym for each of the 492 novel terms we incorporated to SnomedCT thanks to combining the SAVANA algorithm with an expert validation stage.

6.4.3 Enriching SnomedCT

In this section we describe the SnomedCT enrichment experiments. Given a novel term, we aim at finding its best point of attachment, expressed as its closest hypernym. Our approach is unsupervised and hence requires no prior annotation or training. Moreover, we do not exploit any web or Wikipedia-based textual evidence (which we may investigate in future work). However, we do leverage two main resources in our experiments, which are described briefly.

- For syntactic parsing, we use a transition-based parser based on the parsing technology included in the Mate framework [Bohnet, 2010].

[24] As usual during this dissertation, we denote is-a relations between terms and sets of hypernyms as $term \rightarrow \{hypernym_1 \cdots hypernym_n\}$.

- For computing similarities between concepts, we exploit word embeddings derived from training a shallow neural net model [Mikolov et al., 2013c] with the *word2vec* tool, implemented in *gensim*[25]. The model used for our experiments comes from a 2015 dump of the Spanish Wikipedia preprocessed and lemmatized with Freeling [Atserias et al., 2006b]. Our model is 300-dimensional, and is trained using the skip-gram with negative sampling algorithm, using a minimum count of 10 for each word.

Having described the two main technological pivots of our approach, let us describe each of the systems evaluated:

- **Substring***[26] This is a substring inclusion baseline which, for each novel term, assigns as term hypernyms all Snomed concepts that are subsumed in the novel term. For example, given the unseen concept GONARTROSIS, candidate hypernyms are ARTROSIS and ARTROSIS (TRASTORNO). Note that this approach fails short when dealing with longer and more complex terminology, as in the case of the concept NO OTROS HÁBITOS TÓXICOS, where incorrect hypernyms are captured, such as TOS or OTRO.

- **Head Fuzzy Match*** Multiword terms (mwt) may be generalized via their syntactic dependencies. For example, given the novel concept INSUFICIENCIA CARDÍACA CONGESTIVA LEVE, after dependency parsing we are able to isolate INSUFICIENCIA as the mwt's head. This configuration of our approach collects all Snomed concepts of which this head is substring. In this example, we would correctly match INSUFICIENCIA CARDÍACA, but also generate false positives such as INSUFICIENCIA HEPÁTICA or INSUFICIENCIA RESPIRATORIA TIPO 2.

- **Head Exact Match** This is a restricted version of *Head Fuzzy Match*, in which in most cases we only obtain one candidate, i.e. the Snomed concept which matches exactly the out-of-vocabulary (OOV) term's head. For instance, for the concept NO OTROS HÁBITOS TÓXICOS, the retrieved candidate would be the Snomed concept HÁBITO.

- **Distributional** The first of our distributional approaches, exploiting word embeddings, stems from the intuition that similar concepts may occur in similar contexts. This property has been confirmed to hold in many semantic relations [Mikolov et al., 2013c, Mikolov et al., 2013a]. In this configuration of our system, given a term t consisting of a set of words $\{w_i, ...w_n\}$

[25] radimrehurek.com/gensim/models/word2vec.html

[26] We distinguish baseline systems with *.

(after stopword removal), we compute the centroid vector μ of the set of associated word vectors $\overrightarrow{w} \in t$. We obtain $\mu(t)$ as follows:

$$\mu(t) = \frac{1}{|t|} \sum_{\overrightarrow{w} \in t} \frac{\overrightarrow{w}}{||\overrightarrow{w}||} \tag{6.6}$$

We perform the same operation on all candidate Snomed concepts. Specifically, we obtain, given a Snomed terminology $\mathcal{S}$, for each Snomed term $t_s \in \mathcal{S}$, its corresponding centroid vector $\mu(t_s)$. Then, our algorithm returns as best match the Snomed concept maximizing the semantic similarity between t and t_s, denoted as $\text{SIM}(t, t_s)$, and computed via cosine score as follows:

$$\text{SIM}(t, t_s) = \frac{\mu(t) \cdot \mu(t_s)}{\mu(t)\, \mu(t_s)} \tag{6.7}$$

This operation yields a ranked list of candidates by score, where score is the cosine score above, and thus the predicted candidate is the term t_s with the highest similarity with the input term t.

- **DistDep** Our last system is performed with the **DistDep** system, which combines head word lookup with similarities derived from word embeddings. It simply consists in comparing the vector associated to the head node (as extracted in any of the **Head**-based approaches) of the novel term with the centroid of all available concepts in $\mathcal{S}$, and keeping as best match the highest scoring candidate.

6.4.4 Evaluation

6.4.4.1 Distance-based Evaluation

The evaluation of a system's performance in terms of its ability to attach novel terminology to an existing knowledge repository is traditionally performed by considering distance between reference nodes and predicted nodes, i.e. the best point of attachment, and the decision made by the system. While evaluation metrics exist for computing semantic similarity over lexical databases like WordNet, these are not suitable in our case because our branch of Snomed is designed in a slightly different fashion, as it can be considered a multiroot directed acyclic graph (DAG), and hence in many cases, given two concepts, there is no *least common subsumer* other than the root node. For instance, the path between TRASTORNO CON TALLA BAJA and PRUEBA DE VARIANTE DE HEMOGLOBINA includes one of

the root nodes in the SnomedCT taxonomy, namely SNOMED CLINICAL TERMS (ENERO 2014).

This makes metrics like Wu&Palmer Similarity [Wu and Palmer, 1994], which considers lowest common subsumers in their similarity computation, unsuitable. For this reason, we propose a distance metric for evaluation purposes sensitive to terminological databases shaped as DAGs rather than trees (like WordNet). We account for the fact that there may be several valid points of attachment and hence compute an average of node-based shortest path $sp(\cdot)$ over all predicted candidates and all gold standard nodes.

Let $\mathcal{G}$ be the set of gold standard points of attachment to a novel term t, and $\mathcal{P}$ the set of predictions generated by a system. We define an Error-Score function E that, given a novel term t, computes the average shortest path of all predicted points of attachment $p \in \mathcal{P}$:

$$E(t) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{\sum_{g \in \mathcal{G}} sp(p, g)}{|\mathcal{G}|} \tag{6.8}$$

In addition, we report a second evaluation based on whether a system is able to capture all the gold standard points of attachment, regardless of additional incorrect predictions. This is performed only on those systems which return a *set of candidates*, which is not the case of the distributional systems **Distributional** and **DistDep**. We included evaluation under this Recall score, which we denote as $R(t)$, as we foresee a real world scenario where human post-edition of false positives may be less time-consuming than finding in SnomedCT the best points of attachment for each novel term. We simply set $R(t) = 1$ if a given approach is able to cover, with its *n* predictions, all the possible gold standard attachments, and $R(t) = 0$ otherwise, and average results over the total prediction sets. We provide the evaluation results for both criteria in Table 6.19 (note the N/A value for Recall in systems that *do not* return a set of candidates). The two main conclusions that can be drawn from our experimental results are that, first, leveraging similarities derived from word embeddings improve the performance of MTD enrichment systems, and second, that exploiting a greedy approach of fuzzy syntactic head matching is a reasonable strategy for increasing recall.

Finally, we plotted the performance in Error-Score of our proposed systems (not baselines) over all the novel terms present in the evaluation data. We can observe that the two distributional systems based on word embeddings show a similar behaviour, much better in general than the third best system, **Head Exact** (Figure 6.13).

	Error-Score	Recall
Substring*	8.51	26%
Head Fuzzy	7.07	**84**%
Head Exact	4.72	13%
Distributional	**3.34**	N/A
DistDep	3.36	N/A

Table 6.19: Evaluation results for our proposed systems in terms of average performance of all its predictions (Error-Score), and Recall. Note the N/A values for distributional systems, for which only the best candidate is considered for evaluation.

Novel Term	$\mathcal{G}$	**Novel PoA** (fp)	**Correctness**
dermatitis seborreica leve	eccema seborreico	dermatitis	Yes
servicio de cardiología pediátrica	servicio hospitalario	servicio de cardiología	Yes
artrosis cervical	artrosis	linfadenopatía cervical	No
talla baja idiopática	trastorno con baja estatura	al examen: estatura baja	No

Table 6.20: Illustrative cases where some of the novel concepts discovered by our approach, and evaluated as false positive (fp) by the automatic criteria, were considered correct in a second pass by human domain experts.

6.4.4.2 Human Evaluation

We assume in our automatic evaluation that human experts in the biomedical domain will provide a solid ground truth against which system predictions can be evaluated. However, given the size of SnomedCT, our system may provide correct points of attachment for novel terminology which were not included in the first place, and this is penalized in the automatic evaluation. For this reason, we presented human experts with the set difference between the sets of gold and predicted points of attachment, and asked them to label them as correct or incorrect. We find an average of 27% correctness over all systems, which suggests that certain cases of *false positives* were actually correct predictions and hence were valid inclusions of novel terms along with their associated hypernymic relations. We illustrate a few cases of *false positives* in Table 6.20 together with their correctness according to a domain expert.

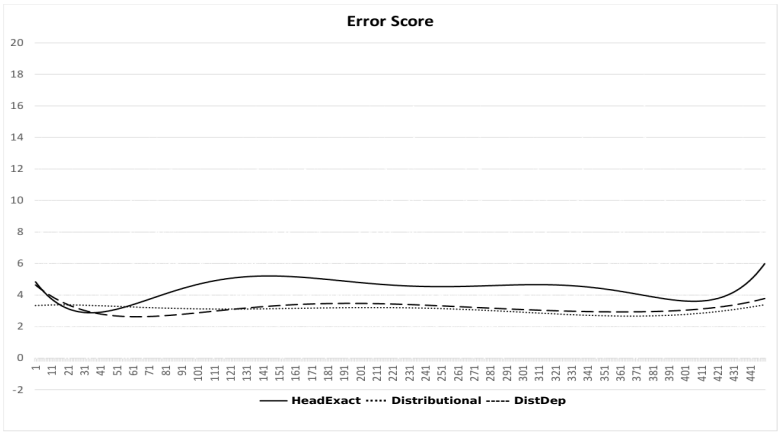


Figure 6.13: Unnormalized (range $[0, +\infty[$) error-Score results (y axis) for the three systems we propose, namely **Distributional** (dotted), **DistDep** (dashed), and **Head Exact Match** (line), over the whole test terminology (x axis).

6.4.5 Conclusion

The rapidly growing interplay between Artificial Intelligence and healthcare is producing innovative assistive technologies (e.g. adaptive and rehabilitative devices), as well as medical support systems which leverage large quantities of heterogeneous data. Among the latter, let us highlight SAVANAMED, which thanks to the SAVANA algorithm, provides a real-time medical support system by making sense of textual information present in CHRs. In this paper, we described how the SAVANA algorithm, backed up by a validation stage carried out by medical practitioners, was used to produce a ground truth for evaluating a system in the task of MTD enrichment. We evaluated several systems against this data and found that combining linguistic information derived from syntactic dependencies, as well as similarities computed over word produces the best results. To the best of our knowledge, both SAVANA and the MTD enrichment system are the first systems of their kind developed for the Spanish language.

Chapter 7

LANGUAGE RESOURCES AND SOFTWARE

Owing to the very nature of this dissertation, which has a strong focus on the automatic creation, enrichment and extension of knowledge resources, in this chapter we present and describe a set of assets associated with the experiments described and evaluated so far. We release datasets for the use of the research community, as well as a number of software applications in the hope to foster research in this area of NLP and AI. Specifically, we accompany this thesis with: (1) Train and test corpora and precomputed definition-wise noun phrase frequencies derived from **SequentialDE** (in Catalan), which can be further leveraged for DE in the Catalan language; (2) A `python` toolkit that implements a lightweight version of **WeakDE**, along with train, development and test datasets; (3) A `python` API for **TaxoEmbed**, along with associated training and evaluation data; (4) Automatically constructed taxonomies in several domains of knowledge, generated by the **ExTaSem!** taxonomy learning system; (5) Different resources associated with **KB-Unify**, including disambiguated output from OIE systems, pairwise alignment and the final **KB-Unify** resource; (6) **ColWordNet**, the collocational extension of WordNet, as well as a `python` API for enabling custom experiments; and (7) Several versions of our **Music Knowledge Base**.

7.1 SequentialDE: Datasets description

We release train and test datasets with definitions and distractors in the Catalan language, extracted from the Catalan Wikicorpus[1]. The only difference between them is that the test data has undergone a manual validation of a portion of the data (it remains as future work to comprehensively evaluate the quality of the test

[1]Data available at `bitbucket.org/luisespinosa/catalande`

set). These corpora are provided in CoNLL format (tab-separated files, where each row corresponds to one word, and columns represent features). Columns provide information derived from a preprocessing stage, as the result of running the Freeling [Atserias et al., 2006a] linguistic workbench on them. For example, given a definition such as:

> Un metabòlit és qualsevol molècula utilitzada o produïda durant el metabolisme

the processed version contains the information shown in Table 7.1.

Un	un	DI0MS0	0.95	b_def
metabòlit	metabòlit	NCMS000	1.00	i_def
és	ser	VSIP3S0	1.00	i_def
qualsevol	qualsevol	DI0CS0	0.99	i_def
molècula	molècula	NCFS000	1.00	i_def
utilitzada	utilitzar	VMP00SF	1.00	i_def
o	o	CC	0.99	i_def
produïda	produir	VMP00SF	1.00	i_def
durant	durant	SPS00	1.00	i_def
el	el	DA0MS0	0.99	i_def
metabolisme	metabolisme	NCMS000	0.99	i_def

Table 7.1: Example of a definition sentence, preprocessed with Freeling. It contains surface form information (column 1), lemma (column 2), part of speech (column 3), probability for such part of speech (column 4), as well as the classification label for the token (column 5).

The train corpus we release contains 195,071 sentences, with 111,470 definitions. Likewise, the test set contains 4,281 sentences, out of which 2,825 are definitions. Let us recall that the process of constructing these datasets follows the idea introduced in [Navigli and Velardi, 2010], in that non definitions or distractors are "syntactically plausible false definitions", or contain an explicit mention of a term (the title of the Wikipiedia page in which the sentence appear), as in the following example:

> **Definition**: "El **dadaisme**, també conegut com a moviment dadà, va ser un moviment intel·lectual, literari i estètic d'avantguarda, desenvolupat entre el 1916 i el 1925, precedent immediat del surrealisme.

Non Definition: "A Catalunya, la relació amb el **dadaisme** va ser molt directa, a causa d'un grup d'artistes d'avantguarda que per mitjà les Galeries Dalmau van donar a conèixer les obres del moviment (...)".

Finally, we complement these corpora with additional support data in the hope that it can constitute a valuable resource for feature engineering. This data is divided in three releases: (1) frequency list of all tokens in the training set that appear in *definiendum* position; (2) complementary frequency list for tokens appearing in *definiens* position; and (3) frequency distribution of all noun phrases (including multiword expressions as detected by Freeling) both at definiendum and definiens position.

7.2 DefExt: Definition Extraction Tool

In GlobaLex (2016) we presented **DefExt**, a lightweight `python` implementation of **WeakDE** (Section 3.4). The main idea behind DefExt is to allow any user to quickly extract features useful for the DE task (and to easily extend the initial feature set), and to run a bootstrapping algorithm which iteratively identifies highly confident definitions, removes them from the target data, retrains and applies the model again as many times as the user decides (or until convergence). A summary of the workflow of DefExt is illustrated in Figure 7.1.

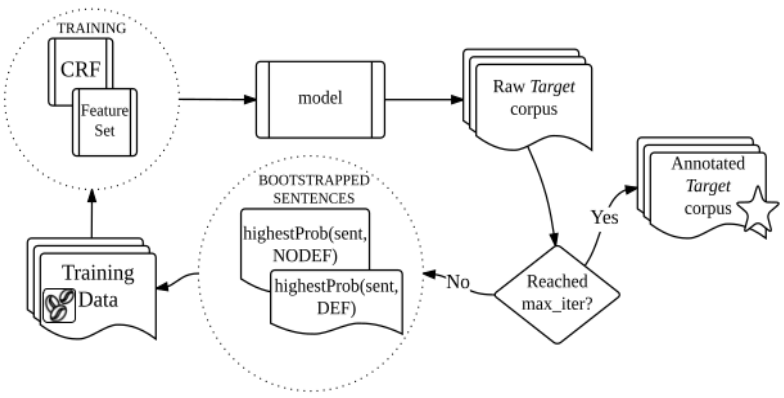


Figure 7.1: Workflow of DefExt, a lightweight implementation of the WeakDE bootstrapping algorithm.

In the repository[2], we make available raw and processed training data from the WCL dataset [Navigli and Velardi, 2010], as well as access to preprocessed versions of the ACL-ARC corpus [Bird et al., 2008].

7.3 TaxoEmbed domains and associated datasets

In addition to the experiments reported in Section 4.2, we also provide the research community with the following datasets and software[3]:

- **Domain Clustering** of BabelNet synsets according to the approach described in [Camacho-Collados et al., 2016]. This synset-level clustering is essential for training what [Fu et al., 2014] named *piecewise linear transformation* matrix, in our case operating at the sense level. The current release contains more than 1.6M BabelNet synsets in a tab-separated file associated to at least one domain of knowledge (e.g. *art* or *physics*).

- **Wikidata Hypernym Branch**, which contains one hyponym-hypernym pair per line in a tab separated file (first column including the hyponym, and second column, the hypernym). In the current release, we provide a total of 5.3M term-hypernym pairs at the synset level.

- **KB-Unify isa-0.9 Branch**, where we release an automatically constructed dataset with term-hypernym pairs in the same format as the Wikidata pairs. Note that in this case, this is an automatic mapping from NELL to BabelNet, and even if the disambiguation threshold is set to 0.9, the veracity of these relations was not manually assessed. Our release contains 13.5M term-hypernym pairs.

- **Taxoembed Python API**, which can be used to load text, synset or sense-level training data, and executes the pipeline described in [Espinosa-Anke et al., 2016]. It also includes a script that can be run in interactive mode, which prompts the user for an input BabelNet synset, and returns its most likely hypernym in a predefined embeddings model. Finally, reproducibility allowing potential extensions is also possible, for example by improving the construction of BabelNet domains [Camacho-Collados and Navigli, 2017].

[2] `bitbucket.org/luisespinosa/defext`

[3] Available at `bitbucket.luisespinosa.com/taxoembed`

7.4 ExTaSem!: Evaluation Data and Taxonomies

In addition to the experiments we report in Chapter 5, where we described experiments evaluating the performance of EXTASEM!, we release taxonomies for the following domains: AI, Chemical, Equipment, Food, Science and Terrorism[4]. Each taxonomy comes in two formats: (1) Easy to read `html` files (with hyperlinks to disambiguated nodes to their corresponding BabelNet page, where possible), along with the score provided by the system (Section 7.4.1); and (2) CSV files formatted so that they can be opened and inspected with the `gephi` graph visualization tool[5]. Visualizations for manual inspection of semantic clusters, along with a discussion on how these could be beneficial for NLP, are provided in Section 7.4.2.

7.4.1 `HTML` Taxonomies

In Figure 7.2 we show a screenshot of an `html` taxonomy in the EQUIPMENT domain, one of the benchmarking domains in TexEval [Bordea et al., 2015]. An example of the capability of EXTASEM! to encode novel taxonomic relations can be found in the example concerning the BabelNet synset "bn:02795723n" (edge id 12). This synset corresponds to NETRA, an eye diagnostic device[6]. In the original BabelNet page, this concept is not associated to any hypernym (no relation in Wikidata or Wikipedia, other than it belongs to the *Sensors* and *Optical metrology* categories). In EXTASEM!, however, the encoded hypernym is as informative as "mobile eye diagnostic device".

Finally, the sample taxonomy also provides examples of the hypernym decomposition module (Section 5.1.3). For instance, in edge id 31, the taxonomic relation is `digital datalink system`→`datalink system`.

7.4.2 Visualizing and inspecting semantic clusters

It is possible to visualize EXTASEM! taxonomies, and use these visualizations to manually inspect clusters or sub-domains. We illustrate this potential via the `gephi` visualization tool, and use the `food` domain as a use case.

A general overview of a domain taxonomy can be useful to examine structural properties such average depth, presence of cycles or whether all components are

[4]Data available at `bitbucket.org/luisespinosa/extasem`

[5]`gephi.org`

[6]`http://babelnet.org/synset?word=bn%3A02795723n&lang=EN&details=1&orig=bn%3A02795723n`

ExTaSem! - EQUIPMENT taxonomy

Edge id	Hyponym	Hypernym	Weight
0	bn:00165672n	manufacturer	0.14167099485
1	bn:03474130n	standardized_test_method	0.150335282341
2	bn:00015425n	device	0.149265181551
3	picture_acquisition	acquisition	0.152410950427
4	century_device	device	0.140701825973
5	bn:00466634n	secure_storage_container	0.146874450983
6	tube_device	device	0.45855871566
7	bn:00042837n	craft	0.137445775333
8	electricity	equipment	0.323243441683
9	long-range_airborne_navigation_system	airborne_navigation_system	0.786135133152
10	bn:00071301n	protective_equipment	0.389367779677
11	physical_box	box	0.137858761397
12	bn:02795723n	mobile_eye_diagnostic_device	0.140701825973
13	wheeled_military_vehicle	military_vehicle	0.833543921557
14	delivery_vehicle	vehicle	0.351062552162
15	philatelic_material	material	0.300122102494
16	bn:00784257n	british_vehicle_maker	0.139514705634
17	bn:03572568n	vehicle	0.145423213555
18	motor_vehicle	vehicle	1.28838362763
19	bn:03447125n	british_truck_manufacturer	0.154739802574
20	vehicle_manufacturing_company	manufacturing_company	0.549388676145
21	rigid_cylindrical_container	cylindrical_container	0.137259713806
22	german_multinational_manufacturer	multinational_manufacturer	0.301700362068
23	specialized_firefighting_apparatus	firefighting_apparatus	0.412788332358
24	bn:00637982n	apparatus	0.190992337435
25	bn:00582812n	offroad_technology	0.184552407954
26	drive_all-terrain_vehicle	all-terrain_vehicle	0.410919036303
27	bn:02204625n	apparatus	0.199289984941
28	instrumentation_system	system	0.49773246947
29	commodity_cargo	cargo	0.384569614004
30	bn:01226418n	large_dump_truck	0.177329924695
31	digital_datalink_system	datalink_system	0.303126088794
32	bn:01070752n	sensor	0.156248186261
33	bn:01045186n	device	0.177998106522
34	bn:03350131n	optical_device	0.136744046678
35	sleeveless_garment	garment	0.334957794943

Figure 7.2: Sample of the html page for the *equipment* domain taxonomy.

connected. In addition, well defined domains seem to show a higher proportion of semantic clusters (i.e. a higher number of well defined sub-domains), and visualization can constitute a straightforward evaluation tool, complementary to specific metrics based on evaluation of (possibly hierarchical) clusters. In fact, evaluation reports of TexEval include qualitative discussion based on taxonomy visualization. In the specific case of the `food` taxonomy we automatically built, it can be clearly seen from Figure 7.3 that there are well defined clusters.

An interesting aspect of these taxonomies is that they combine highly specific concepts at the text level, with many disambiguated nodes (against the BabelNet semantic network). See, for example, a zoom-in of the "cake" cluster in

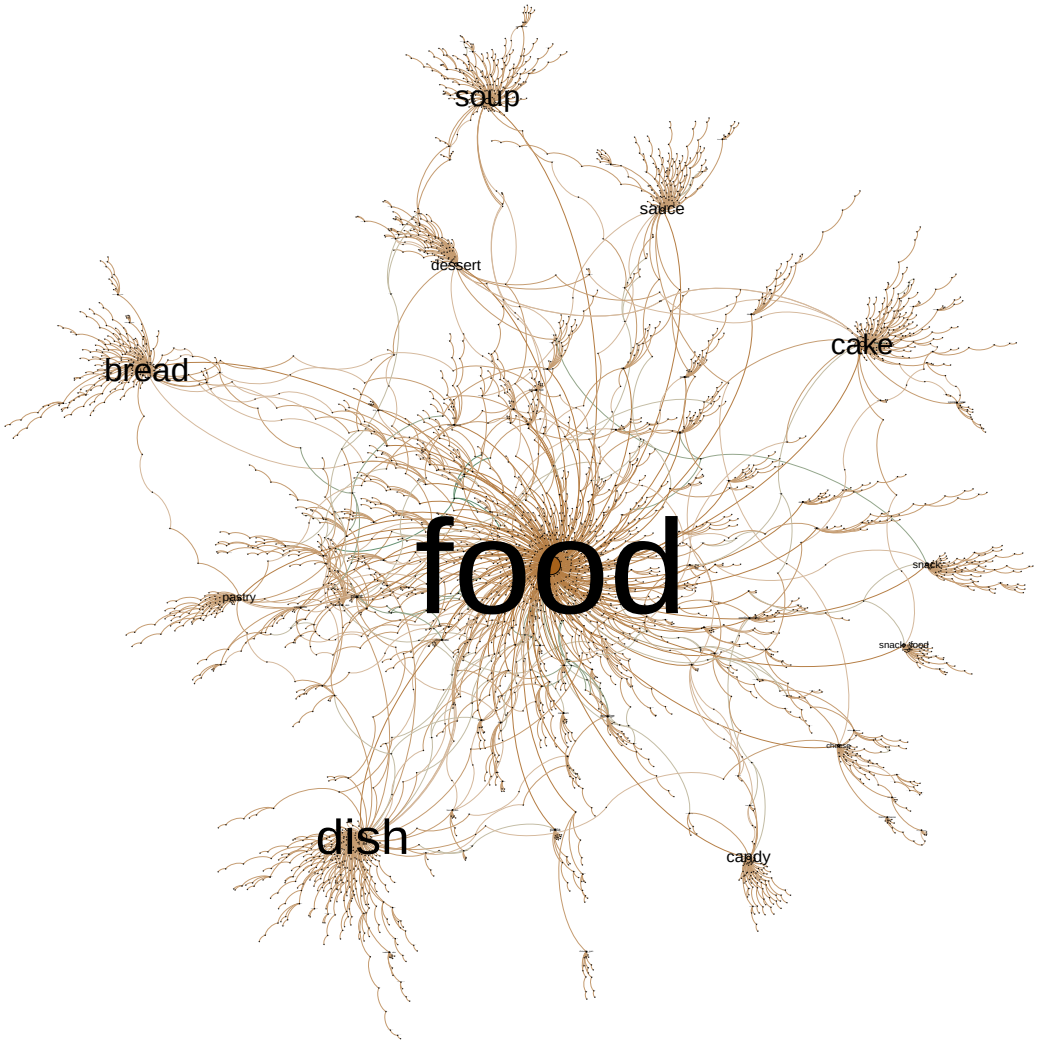


Figure 7.3: Food taxonomy generated with ExTaSem! The node at the center is the root node `food`. The density of the graph has been scaled down for illustrative purposes.

our working example (Figure 7.4). In the current version of BabelNet, the concept for 'wedding cake' (BabelNet id "bn:00013073n", highlighted in green in the figure) only has one (hypernymic) relation with the 'cake' concept. Thanks to EXTASEM!, we introduce four intermediate nodes between 'wedding cake' and 'cake', namely 'rich cake', 'multi-layered cake', 'decorated cake' and 'traditional cake'.

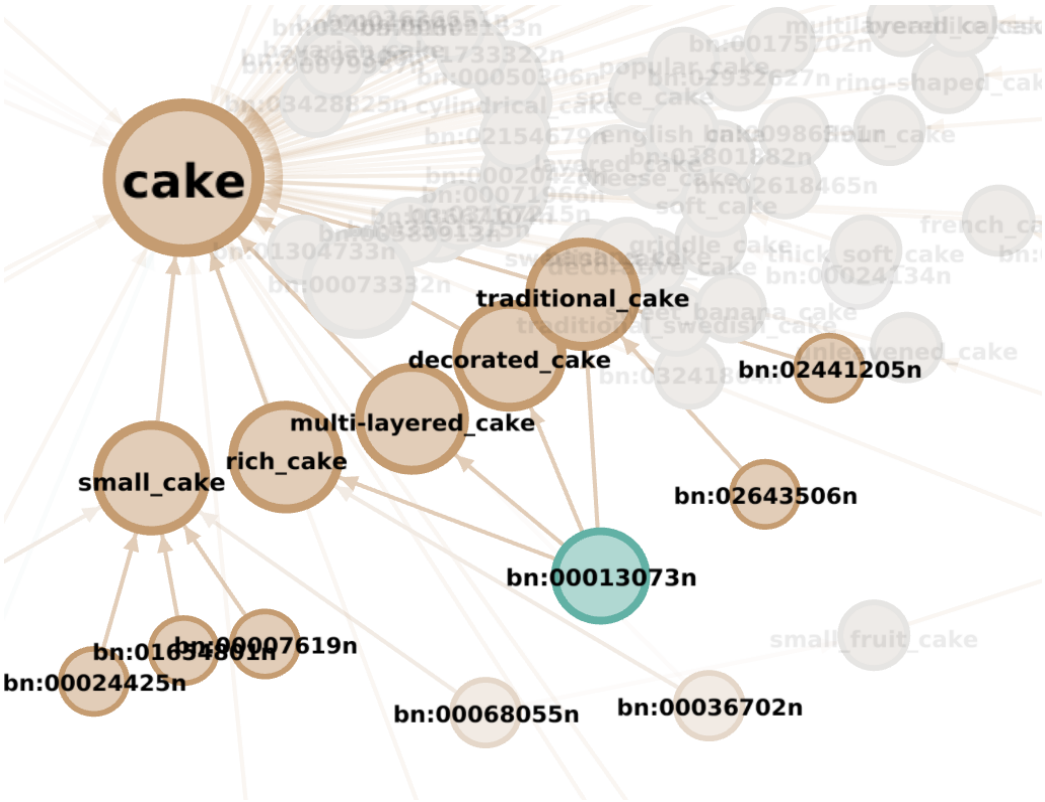


Figure 7.4: A highlight of the cluster for the concept "cake". Note the mixture between disambiguated concepts (BabelNet synsets) and specific terms such as "small cake" or "multi-layered cake".

Finally, by resizing nodes by degree, it is possible to manually inspect the most prominent sub-domains (or semantic clusters) generated by our algorithm. An additional visualization of these clusters is shown in Figure 7.5, where generic food-related terms such as "dish", "candy", "snack", "sauce", "bread" or "dessert" appear. This kind of information may be useful for ontology engineering, WSD, semantic search, or any task requiring some kind of hierarchical modeling of a target domain of knowledge.

Based on how useful this information can be for discovering patterns in lexical taxonomies, we encourage the research community in taxonomy and ontology learning to incorporate to evaluation procedures visualization-based assessments. It is important not only to have many correct relations, but to have them integrated in a homogeneous knowledge graph whose most salient entities can be

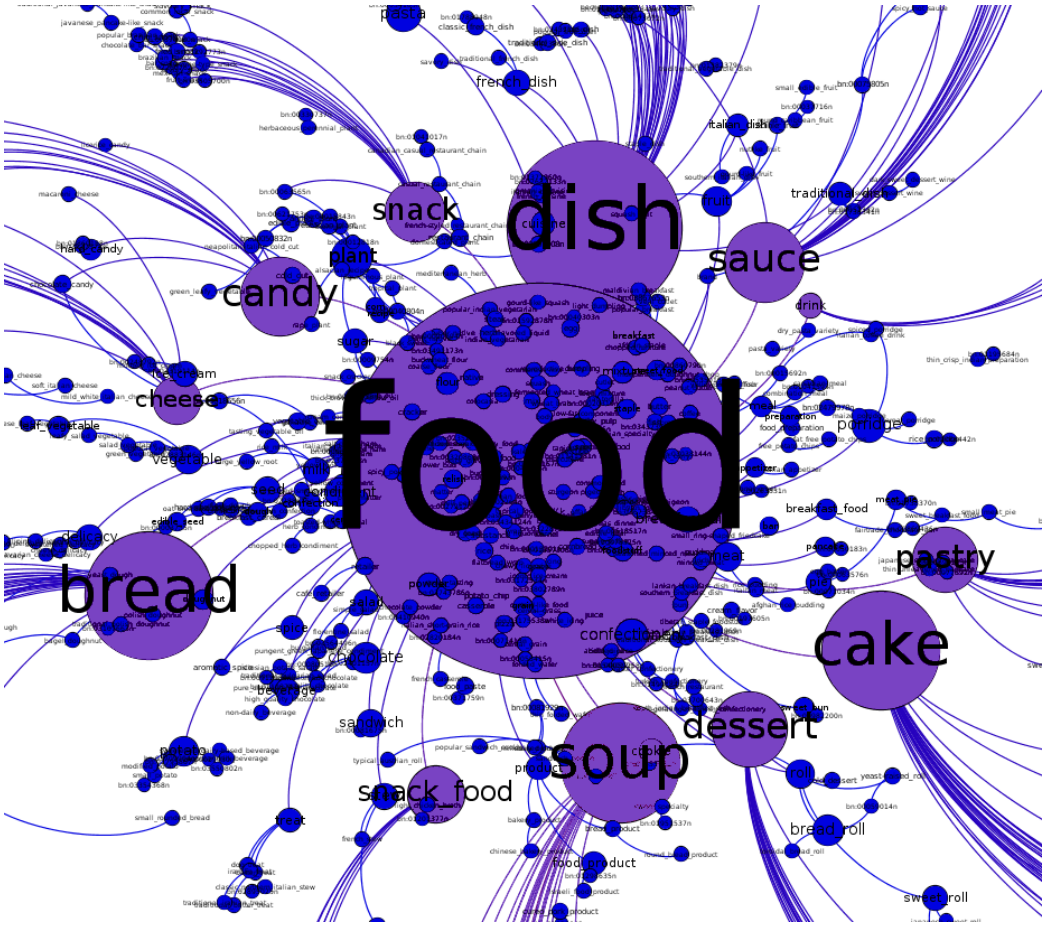


Figure 7.5: Most prominent clusters in the ExTaSem! food taxonomy. Many node labels have been removed for clarity.

clearly identified. It would make little sense to have a taxonomy on the food domain without top concepts such as the ones highlighted in Figure 7.5. In addition, visualization techniques also provide the means to easily inspect potential disconnected nodes, presence of cycles, and to get the gist of the structural properties of the graph, which can be complementary with quantitative metrics.

Let us point to another interesting scenario, this time involving the `chemical` domain. In TexEval 2015, this domain proved to be the most difficult to model by all participating systems (the best system achieved an average Cumulative F&M score of 0.23, more than 20 points lower than the following domain). The chemical domain is extremely technical, difficult to model, and with abnormally long and convoluted concepts (e.g., *glutamic acid-2,3,3,4,4-d5*, a type of molecular entity). By inspecting the result of ExTaSem! on this domain (Figure 7.6), we observe that rather than a root concept *chemical*, it may be desirable to evaluate

a `chemical` taxonomy learning system on subdomains such as *chemical compound* or even *organic compound* (which is defined as any chemical compound containing carbon), which have shown to emerge (degree-wise) as highly prominent concepts.

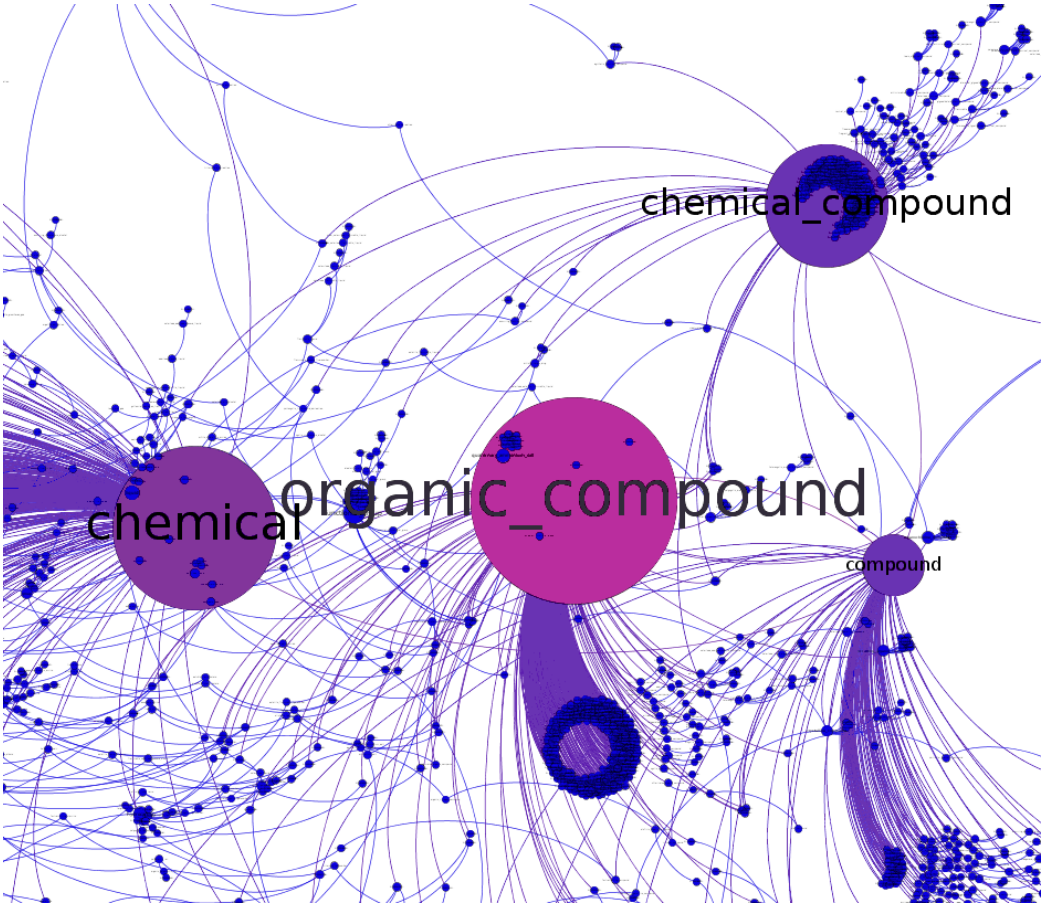


Figure 7.6: Illustration of how non-root concepts in the chemical domain, such as "organic compound" or "chemical compound" rival frequency and popularity of the term "chemical" as yielded by the ExTaSem! taxonomy.

Finally, our last argument for incorporating visualization-based techniques in taxonomy evaluation concerns *error analysis*. In very large domains (the `food` taxonomy we have discussed in this section contains 3930 edges, for example), the potentially correct domain-specific taxonomic models is very large. By being able to "hover" over a taxonomy, a human expert may detect inconsistencies easier than

going over a sample of randomly selected edges (which was the choice in TexEval 2015), or by looking at edges involving popular nodes (which are more likely to be correct, as generic or top concepts occur more frequently in corpora and thus there is extensive textual evidence for defining their position in a taxonomy). As an example, consider Figure 7.4, where one may argue that *rich cake* is an incorrect hypernym for the concept *wedding cake*. In fact, this node comes from extracting the hypernym from the following Wiktionary definition: "*rich* or highly ornamented **cake**, to be distributed to the guests at a wedding, or sent to friends after the wedding". This kind of qualitative analysis can be arguably carried out more effectively with the aid of proper visualizations for lexical taxonomies.

7.5 KB-U: Disambiguated and aligned OIE systems

We release the KB-Unify associated dataset as follows[7]:

- **KB-Unify**: The KB resulting from the disambiguation and alignment algorithm.
- **KB-Unify Alignments**: Final pairwise alignments between the KBs described in Section 6.2 In the current 1.0 release of KB-Unify, these KBs are NELL, ReVerb, PATTY and WiseNet.
- **Disambiguated KBs**: We also provide a disambiguated version of each of the "unlinked" KBs that are part of KB-Unify 1.0 (namely NELL and ReVerb[8]).
- **Evaluation Data**: We also release the evaluation data that accompanies the experiments reported in Section 6.2.8. Specifically, these evaluation datasets are used to assess the modules on automatic *disambiguation*, *specificity* and *alignment*.

7.6 CWN: Data and API

In the CWN project (described in Chapter 6), we aimed at providing the research community with a useful extension of WordNet which included collocational information. This information was automatically obtained thanks to a distributional

[7] Available in the following url: `http://lcl.uniroma1.it/kb-unify/`

[8] We include both disambiguated versions of ReVerb's output, one resulting from running its pipeline on Wikipedia, and another one derived from its execution over the ClueWeb corpus [Callan et al., 2009].

pipeline based on sense-level embeddings. We complement those experiments with one additional contribution, namely a Python API that replicates the whole CWN pipeline, allowing for any user to replicate our results, and more importantly, use the implementation with custom data, generating *subsets* of WordNet (e.g. only with collocations of intensity, or introducing collocations only for hyponyms of 'sorrow'). Finally, in addition to the CWN resource and its associated API, we also make available pre-trained vectors retrofitted [Faruqui et al., 2015] with collocational information for the four semantic categories evaluated in the original CWN publication[9].

7.7 MKB: Music Knowledge Base

The last release accompanying this dissertation is derived from the experiments described in Chapter 6, where we constructed a full-fledged Music Knowledge Base (MKB) from scratch. We release several versions of our automatically constructed MKB[10], in addition to the evaluation data used to assess the quality of the extraction algorithm. The dataset derived from our best configuration is a graph of 11,010 nodes disambiguated against DBpedia and MusicBrainz (where possible), connected by 11,835 relations expressed in natural language. In addition, evaluation data used to validate the quality of the extracted relations is also provided. In Figure 7.7, we show a sentence, two identified entities, and a relation expressed among them. The evaluator then is asked to mark whether there is a relation expressed between them in the sentence or the pattern extracted. We also provide the extractions performed by ReVerb.

7.7.1 The MKB Dataset

All versions of MKB are released in `json` format. Each entry is a relation, which contains the following fields:

- **clustered**: How many relation clusters contain this relation.
- **dep_freq**: Frequency of the dependeny relation between domain and range.
- **dep_path**: Part of speech path in the dependency tree between domain and range (e.g. sbj-obj).

[9] Data and code available at `bitbucket.org/luisespinosa/cwn`
[10] Available at `http://mtg.upf.edu/download/datasets/kbsf`

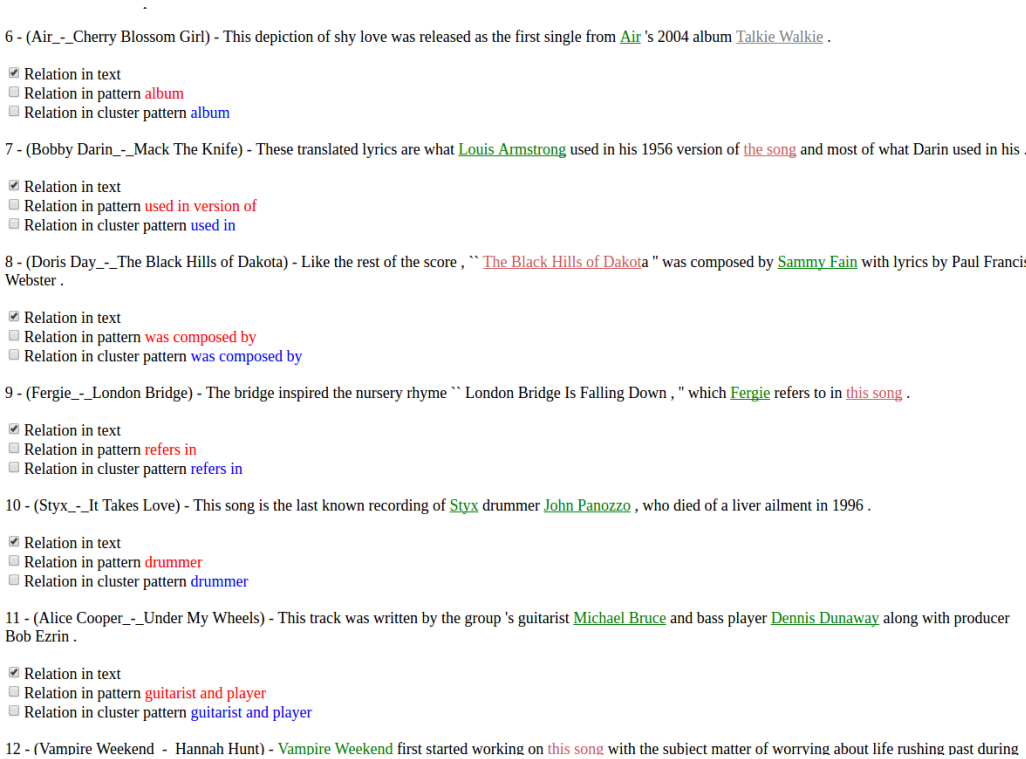


Figure 7.7: Screenshot of the evaluation procedure for the extracted relations of MKB.

- **dep_path_cluster**: Part of speech cluster that subsumes this relation (e.g. 'nmod')[11].
- **domain**: First argument of the relation, disambiguated against DBpedia, where possible (e.g. `dbpedia.org/resource/Ride_(Ciara_song)`).
- **domain_mbid**: MusicBrainz id of the domain.
- **domain_offset**: Character offset of the domain.
- **domain_tfidf**: tf*idf score of the domain (considering as individual documents each biography).
- **domain_type**: Music type of the domain (e.g. Song)[12].
- **id**: Relation id.

[11]There are entries for these attributes for part-of-speech and surface form.
[12]The same attributes are included for the entity appearing in range position.

- **num_neighbours**: Number of nodes at a distance of one edge from both domain and range.
- **num_nodes**: Number of nodes encoded by this relation.
- **num_paths_in_path**: Number of subsumed relation paths contained in the current relation.
- **score**: A score provided by our relation weighting policy.
- **sentence**: The original sentence from which this relation was obtained, e.g. "The song features Ciara 's fellow Atlantan , Ludacris".
- **score**: A score provided by our relation weighting policy.

Similarly as in the EXTASEM! release (Section 7.6), a `gephi` compatible file is released for MKB, allowing visualization of artist clusters. This KB has the particularity that it integrates in one single unified resource entities musical entities detected in the Songfacts corpus, and which may be encoded in DBpedia, MusicBrainz, or both. We also release evaluation data, illustrated in Figure 7.7.

Chapter 8

CONCLUSIONS

The vision behind this dissertation was to provide solid proof that combining what [Hovy et al., 2013] called *structured resources* like dictionaries, along with *unstructured* resources (e.g. vector space models, or simply text corpora), it is possible to achieve very competitive results in several NLP tasks, while at the same time providing frameworks for the creation and extension of knowledge repositories. Motivated by the good results the interplay between structured and unstructured resources had shown in the past, in this thesis we aimed at contributing to the current state of the art in NLP from several standpoints, namely: (1) developing and evaluating different strategies for sentence representation, WSD, and entity linking, leveraging techniques derived from lexical semantics; (2) we have improved tasks like Definition or Hypernym Extraction, and have coined a novel task called *Hypernym Discovery*, which parts ways from previous and less realistic approaches to inferring hypernymy based on binary classification given a term-hypernym candidate pair; (3) we have achieved the best results in a number of subtasks and knowledge domains related to taxonomy learning, and have proposed directions for future work in taxonomy evaluation, and (4) we have developed systems for automatic KB enrichment for compositional meaning as well as knowledge representation in restricted domains.

In addition, and in the hope that this thesis contributes significantly to improving knowledge-based Artificial Intelligence and NLP systems, we provide a large number of automatically generated (and thoroughly evaluated) datasets, both domain-specific and generic. It does not seem to be adventurous to claim that the trend today in NLP is the exploitation of huge amounts of data with minimal annotation, as the field of ML is advancing at such a fast pace that there seems to be the case that the more data, the better, even if it is noisy, as long as it is enough to be overall useful for any statistical model. The dominance of approaches based on neural architectures, which do not require a feature engineering step in the "traditional" machine learning way, seems to have closed a few doors for many

knowledge-based applications. However, the knowledge and capacity for reasoning that humans exhibit is still an asset which we cannot obviate, and for this reason, it seems obvious that any approach capable to take advantage of both *big data* and human expertise will have an edge versus purely statistical models.

Furthermore, this dissertation is also a call for commitment to the NLP community in that data and software must be not only easily accessible, but well documented, to foster collaboration between institutions and to make seemingly distant worlds like lexicography or knowledge engineering, and NLP or ML, to interact more often. We expect that, by incorporating human knowledge to highly sophisticated technical approaches, artificial agents will learn faster (e.g. learning better paraphrases, generalizations or multiword expressions); but also lexicographers and professionals in the digital humanities field will benefit dramatically from all this exciting technology and data that is swarming research centers.

Moreover, in this thesis we have come to the following **key findings**, which we list and describe as follows:

- DE systems largely improve with syntactic and distributional information.
- It is possible to improve the quality of Hypernym Discovery systems by aggregating seemingly noisy information coming from OIE systems. There is, however, a notable difference across domains depending on how over (or under) represented they are in standard OIE triple collections.
- In tasks concerning the modeling of linear transformations between semantically related linguistic items using the *translation matrix* approach introduced in [Mikolov et al., 2013b], the *quality* of the data and the homogeneity of the semantic relation that is to be modeled rivals importance with the size of the data used.
- The taxonomy learning task benefits substantially by introducing definitional information, on one hand, as well as our novel domain-pertinence distributional scheme for candidate taxonomic paths.
- Domain-specific terminological databases and KBs can be extended and improved by means of the combination of syntactic, statistical and distributional information.

With regards to the **limitations** of the approaches developed and evaluated throughout this dissertation, we have identified the following. First, with regard to DE, we have not explored neural approaches to classification [Ling et al., 2016] or generation [Noraset et al., 2016]. Moreover, it remains for a future challenge to combine our two **DependencyDE** and **SemanticDE** systems, so that the interplay between linguistic and distributional information is quantitatively explored.

Second, previous research in the Hypernym Discovery/Detection task has extensively leveraged *hypernymic* embeddings models [Yu et al., 2015], supervised approaches that learn distributional relations from large datasets of term-hypernym pairs [Roller et al., 2014] and train linear models for prediction, or introducing neural approaches for modeling is-a relations [Shwartz et al., 2016]. We acknowledge that these are areas where we have not delved in detail, and thus our experiments may be improved by considering the above contributions. Third, while the resources we have automatically created have gone through meticulous evaluation, we feel that their usefulness still remains to be assessed, ideally in semantically motivated downstream tasks. For instance, it would be interesting to evaluate the improvement of a natural language generation system when leveraging ColWordNet or KB-UNIFY. Fourth, our experiments in Section 6.2 are inevitably derived from a "closed world assumption", as in these experiments BabelNet constitutes the reference sense inventory for concepts and entities. In cases where new information or knowledge appears, it remains as an open avenue for future work how to decide whether these should be incorporated in a KR, and how. And fifth, we feel that there are unadressed issues in the evaluation of EXTASEM!, as it is unclear sometimes, for a given node, whether it "deserves" to be included in a taxonomy as full entity, or rather as a property of an existing concept (e.g. should we include the *small cake* node in a *food* taxonomy, or would it be better to include *small* as a property of the key concept *cake*?).

Finally, as for **specific directions of future work**, this thesis opens specific lines for future directions in three main areas. First, the improvement of existing semantic networks and KBs by incorporating automatically gathered information from the web in various forms, from OIE systems to domain-specific collaboratively built resources (e.g. Songfacts). Second, we have reported good results in tasks related to computational lexicography, such as Definition Extraction, Hypernym Discovery or Collocation Acquisition. However, there is not one single system yet out there that automatically builds a full-fledged dictionary entirely from scratch, using as input unstructured corpora (e.g. a collection of papers). This is a very exciting opportunity that can complement very well the current state-of-the-art in Information Extraction in the area of scientific publishing. Finally, we have presented extensive results, evaluation and reflections on the Taxonomy Learning task. This is an area where there is not one single gold standard, as a domain of knowledge can be modeled in many different ways. For this reason, we find it essential that the community puts stronger emphasis on *automatic taxonomy evaluation*, so that it makes sense to carry out research on developing very large lexical taxonomies without having to rely on comparison against WordNet, or on manual evaluation of Precision at edge level, which by necessity is done only on a sample, and therefore its reliability is always subject to a certain random factor.

As a final conclusion, this thesis has taken the reader through experiments

on the interplay between lexicography and *knowledge-based* NLP. Bringing back the notion of the *virtuous cycle of NLP and lexicography*, which we introduced in Chapter 1, our main idea was to show empiric proof that lexicography can constitute a game-changer in NLP by providing high-quality knowledge. This knowledge, in addition, is no longer dependent on the efforts carried out by professional lexicographers, but rather, is growing at an increasingly fast pace due to the rapid growth of collaborative resources. On the other hand, moreover, we have released several resources for improving the workflow in lexicography, providing some kind of automation to tasks such as finding definitions in domain corpora, or *typing* novel concepts by means of finding their best associated (set of) hypernym(s). We hope that this contribution ignites further research where dictionaries, glossaries and other KRs are combined with cutting edge machine learning technology for making knowledge acquisition and formalization faster, easier and of better quality.

Bibliography

[Afzal et al., 2011] Afzal, N., Mitkov, R., and Farzindar, A. (2011). Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, Canadian AI'11, pages 32–43, Berlin, Heidelberg. Springer-Verlag.

[Aguilar et al., 2004] Aguilar, C., Alarcón, R., Rodríguez, C., and Sierra, G. (2004). Reconocimiento y clasificación de patrones verbales definitorios en corpus especializados. In Cabré, T., Estopà, R. & Tebé, C. *La terminología en el siglo XXI*, pages 259–269.

[Alarcón, 2009] Alarcón, R. (2009). *Descripción y evaluación de un sistema basado en reglas para la extracción automática de contextos definitorios*. PhD thesis, Universitat Pompeu Fabra.

[Alarcón et al., 2009] Alarcón, R., Sierra, G., and Bach, C. (2009). Description and evaluation of a definition extraction system for Spanish language. In *Proceedings of the 1st Workshop on Definition Extraction*, WDE '09, pages 7–13, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Alegria et al., 2013] Alegria, I., Cabezon, U., de Betono, U. F., Labaka, G., Mayor, A., Sarasola, K., and Zubiaga, A. (2013). Reciprocal enrichment between Basque Wikipedia and machine translation. In *The People's Web Meets NLP*, pages 101–118. Springer.

[Alfarone and Davis, 2015] Alfarone, D. and Davis, J. (2015). Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1434–1441. AAAI Press.

[Aman and Szpakowicz, 2008] Aman, S. and Szpakowicz, S. (2008). Using roget's thesaurus for fine-grained emotion recognition. In *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pages 312–318.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene Ontology: Tool for the unification of biology. *Nature genetics*, 25(1):25–29.

[Atserias et al., 2006a] Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., and Padró, M. (2006a). Freeling 1.3: Syntactic and semantic services in an open-source NLP library. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, volume 6, pages 48–55, Genoa, Italy. ELRA.

[Atserias et al., 2006b] Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., and Padró, M. (2006b). FreeLing 1.3: Syntactic and Semantic Services in an Open-Source NLP Library. In *Proceedings of LREC*, volume 6, pages 48–55.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

[Banko et al., 2007] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open Information Extraction for the web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 2670–2676.

[Bansal et al., 2014] Bansal, M., Burkett, D., De Melo, G., and Klein, D. (2014). Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051. Association for Computational Linguistics.

[Barnbrook, 2002] Barnbrook, G. (2002). *Defining Language: A local grammar of definition sentences*, volume 11. John Benjamins Publishing.

[Baroni et al., 2012] Baroni, M., Bernardi, R., Do, N.-Q., and Shan, C.-c. (2012). Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

[Baroni et al., 2014] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

[Baroni and Lenci, 2011] Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10. Association for Computational Linguistics.

[Bedmar et al., 2008] Bedmar, I. S., Martínez, P., and Samy, D. (2008). Detección de fármacos genéricos en textos biomédicos. *Procesamiento del lenguaje Natural*, 40:27–34.

[Béjoint, 1994] Béjoint, H. (1994). *Tradition and innovation in modern English dictionaries*. Oxford University Press, USA.

[Bird et al., 2008] Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco. European Language Resources Association (ELRA). ACL Anthology Identifier: L08-1005.

[Bodenreider, 2004] Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

[Bodnari et al., 2013] Bodnari, A., Deleger, L., Lavergne, T., Neveol, A., and Zweigenbaum, P. (2013). A supervised Named-Entity extraction system for medical text. In *CLEF (Working Notes)*.

[Boella and Di Caro, 2013] Boella, G. and Di Caro, L. (2013). Supervised learning of syntactic contexts for uncovering definitions and extracting hypernym relations in text databases. In *Machine learning and knowledge discovery in databases*, pages 64–79. Springer.

[Boella et al., 2014] Boella, G., Di Caro, L., Ruggeri, A., and Robaldo, L. (2014). Learning from syntax generalizations for automatic semantic annotation. *Journal of Intelligent Information Systems*, pages 1–16.

[Bohnet, 2010] Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring

human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

[Bordea et al., 2015] Bordea, G., Buitelaar, P., Faralli, S., and Navigli, R. (2015). SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado. Association for Computational Linguistics.

[Bordea et al., 2016] Bordea, G., Lefever, E., and Buitelaar, P. (2016). Semeval-2016 Task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[Bordes et al., 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. In *Advances in NIPS*, volume 26, pages 2787–2795.

[Borg and Rosner, 2007] Borg, C. and Rosner, M. (2007). Language technologies for an elearning scenario. *Proceedings of CSAW'07*, page 42.

[Borg et al., 2009] Borg, C., Rosner, M., and Pace, G. (2009). Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop in Definition Extraction*.

[Borsodi, 1967] Borsodi, R. (1967). *The definition of definition; a new linguistic approach to the integration of knowledge*. P. Sargent Boston,.

[Bouayad-Agha et al., 2014] Bouayad-Agha, N., Burga, A., Casamayor, G., Codina, J., Nazar, R., and Wanner, L. (2014). An exercise in reuse of resources: Adapting general discourse coreference resolution for detecting lexical chains in patent documentation. In *Proccedings of the Language Resources and Evaluation 775 Conference (LREC)*, pages 3214–3221.

[Bouma, 2010] Bouma, G. (2010). Collocation extraction beyond the independence assumption. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 109–114, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Bowker, 2003] Bowker, L. (2003). Specialized lexicography and specialized dictionaries. *A practical guide to lexicography*, 6:154.

[Briscoe, 1991] Briscoe, T. (1991). Lexical issues in natural language processing. In *Natural Language and Speech*, pages 39–68. Springer.

[Cai et al., 2009] Cai, P., Luo, H., and Zhou, A. (2009). Named entity recognition in italian using CRF. In *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy*.

[Callan et al., 2009] Callan, J., Hoy, M., Yoo, C., and Zhao, L. (2009). Clueweb09 data set.

[Camacho-Collados and Navigli, 2017] Camacho-Collados, J. and Navigli, R. (2017). BabelDomains: Clustering lexical resources by domain of knowledge. In *Proceedings of EACL*, page 223.

[Camacho-Collados et al., 2015] Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2015). NASARI: A Novel Approach to a Semantically-Aware Representation of Items. In *Proceedings of NAACL*, pages 567–577.

[Camacho-Collados et al., 2016] Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

[Carlson et al., 2010] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of AAAI*, pages 1306–1313.

[Celma and Herrera, 2008] Celma, Ò. and Herrera, P. (2008). A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender Systems*, pages 179–186. ACM.

[Celma and Serra, 2008] Celma, O. and Serra, X. (2008). FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics*, 6:250–256.

[Choueka, 1988] Choueka, Y. (1988). Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In *Proceedings of the RIAO*, pages 34–38.

[Chowdhury, 2003] Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1):51–89.

[Church and Hanks, 1990] Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

[Cohen, 1968] Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

[Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[Coster and Kauchak, 2011] Coster, W. and Kauchak, D. (2011). Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669. Association for Computational Linguistics.

[Coughlin, 1999] Coughlin, D. A. (1999). Natural language parser with dictionary-based part-of-speech probabilities. US Patent 5,878,386.

[Councill et al., 2008] Councill, I. G., Giles, C. L., and Kan, M.-Y. (2008). ParsCit: An open-source CRF reference string parsing package. In *LREC*.

[Cowie, 1994] Cowie, A. (1994). Phraseology. In Asher, R. and Simpson, J., editors, *The Encyclopedia of Language and Linguistics, Vol. 6*, pages 3168–3171. Pergamon, Oxford.

[Cowie, 2009] Cowie, A. P. (2009). *The Oxford history of English lexicography*. Oxford University.

[Crouch, 1988] Crouch, C. J. (1988). A cluster-based approach to thesaurus construction. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '88, pages 309–320, New York, NY, USA. ACM.

[Cui et al., 2005] Cui, H., Kan, M.-Y., and Chua, T.-S. (2005). Generic soft pattern models for definitional question answering. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 384–391. ACM.

[Curran and Moens, 2002] Curran, J. R. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9*, ULA '02, pages 59–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

[De Benedictis et al., 2013] De Benedictis, F., Faralli, S., and Navigli, R. (2013). Glossboot: Bootstrapping multilingual domain glossaries from the web. In

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, volume 1, pages 528–538. Association for Computational Linguistics (ACL).

[de Melo and Weikum, 2010] de Melo, G. and Weikum, G. (2010). Menta: Inducing multilingual taxonomies from Wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1099–1108. ACM.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

[Degórski et al., 2008] Degórski, L., Marcińczuk, M., and Przepiórkowski, A. (2008). Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

[Degtyarenko et al., 2008] Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008). Chebi: A database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl 1):D344–D350.

[Del Gaudio et al., 2013] Del Gaudio, R., Batista, G., and Branco, A. (2013). Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. *Natural Language Engineering*, pages 1–33.

[Del Gaudio and Branco, 2007] Del Gaudio, R. and Branco, A. (2007). Supporting e-learning with automatic glossary extraction: Experiments with Portuguese. In *RANLP Workshop: Natural Language Processing and Knowledge Representation for eLearning Environments*, pages 1–7.

[Del Gaudio and Branco, 2009] Del Gaudio, R. and Branco, A. (2009). Language independent system for definition extraction: First results using learning algorithms. In *Proceedings of the 1st Workshop on Definition Extraction*, Borovets, Bulgaria.

[Delli Bovi et al., 2015] Delli Bovi, C., Telesca, L., and Navigli, R. (2015). Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL*, 3:529–543.

[Demner-Fushman et al., 2009] Demner-Fushman, D., Chapman, W. W., and McDonald, C. J. (2009). What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–772.

[Domingos, 2007] Domingos, P. (2007). Toward knowledge-rich data mining. *Data Mining and Knowledge Discovery*, 15(1):21–28.

[Donaldson et al., 2003] Donaldson, I., Martin, J., De Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., et al. (2003). PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC bioinformatics*, 4(1):11.

[Dong et al., 2014] Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM.

[Dunmore, 2011] Dunmore, T. (2011). *Historical dictionary of soccer*. Scarecrow Press.

[Dutta et al., 2014] Dutta, A., Meilicke, C., and Ponzetto, S. P. (2014). A probabilistic approach for integrating heterogeneous knowledge sources. In *Proceedings of ESWC*, pages 286–301.

[Dutta et al., 2015] Dutta, A., Meilicke, C., and Stuckenschmidt, H. (2015). Enriching structured knowledge with open information. In *Proceedings of WWW*, pages 267–277.

[Dyer et al., 2015] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *ACL*.

[Espinosa-Anke et al., 2016] Espinosa-Anke, L., Camacho-Collados, J., Delli Bovi, C., and Saggion, H. (2016). Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of EMNLP*.

[Etzioni et al., 2008] Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open Information Extraction from the web. *Communications of ACM*, 51(12):68–74.

[Etzioni et al., 2011] Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam, M. (2011). Open Information Extraction: The second generation. In *IJCAI*, volume 11, pages 3–10.

[Etzioni et al., 2007] Etzioni, O., Reiter, K., Soderland, S., Sammer, M., and Center, T. (2007). Lexical translation with application to image search on the web. *Machine Translation Summit XI*.

[Evert, 2007] Evert, S. (2007). Corpora and Collocations. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics. An International Handbook*. Mouton de Gruyter, Berlin.

[Fader et al., 2011] Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for Open Information Extraction. In *Proceedings of EMNLP*, pages 1535–1545.

[Fahmi and Bouma, 2006] Fahmi, I. and Bouma, G. (2006). Learning to identify definitions using syntactic features. In *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*.

[Fan et al., 2014] Fan, M., Zhao, D., Zhou, Q., Liu, Z., Zheng, T. F., and Chang, E. Y. (2014). Distant Supervision for Relation Extraction with Matrix Completion. In *Proceedings of ACL*, pages 839–849.

[Faralli and Navigli, 2013] Faralli, S. and Navigli, R. (2013). Growing multi-domain glossaries from a few seeds using probabilistic topic models. In *EMNLP*, pages 170–181.

[Faruqui et al., 2015] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*, pages 1606–1615.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

[Ferragina and Scaiella, 2010] Ferragina, P. and Scaiella, U. (2010). Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM.

[Fillmore and Baker, 2001] Fillmore, C. J. and Baker, C. F. (2001). Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*.

[Flati et al., 2014] Flati, T., Vannella, D., Pasini, T., and Navigli, R. (2014). Two is bigger (and better) than one: the Wikipedia bitaxonomy project. In *ACL*.

[Forman, 2003] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305.

[Fountain and Lapata, 2012] Fountain, T. and Lapata, M. (2012). Taxonomy induction using hierarchical random graphs. In *Proceedings of NAACL*, pages 466–476. Association for Computational Linguistics.

[Fowlkes and Mallows, 1983] Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569.

[Friedland et al., 2004] Friedland, N. S., Allen, P. G., Witbrock, M. J., Matthews, G., Salay, N., Miraglia, P., Angele, J., Staab, S., Israel, D. J., Chaudhri, V. K., et al. (2004). Towards a quantitative, platform-independent analysis of knowledge systems. In *KR*, pages 507–515.

[Fu et al., 2014] Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (2014). Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*, volume 1.

[Gálvez, 2012] Gálvez, C. (2012). Reconocimiento y anotación de nombres de fármacos genéricos en la literatura biomédica. *Acimed*, 23(4):326–345.

[Gao, 2013] Gao, Z. (2013). Automatic Identification of English Collocation Errors based on Dependency Relations. *Sponsors: National Science Council, Executive Yuan, ROC Institute of Linguistics, Academia Sinica NCCU Office of Research and Development*, page 550.

[Gelbukh and Kolesnikova., 2012] Gelbukh, A. and Kolesnikova., O. (2012). *Semantic Analysis of Verbal Collocations with Lexical Functions*. Springer, Heidelberg.

[Gildea and Jurafsky, 2002] Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

[Giuliano et al., 2006] Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *EACL*, volume 18, pages 401–408. Citeseer.

[Granger, 1984] Granger, E. H. (1984). Aristotle on genus and differentia. *Journal of the History of Philosophy*, 22(1):1–23.

[Graupmann et al., 2005] Graupmann, J., Schenkel, R., and Weikum, G. (2005). The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. In *Proceedings of the 31st international conference on Very large data bases*, pages 529–540. VLDB Endowment.

[Grefenstette, 2015] Grefenstette, G. (2015). INRIASAC: Simple hypernym extraction methods. *Semeval-2015 task 17: Taxonomy extraction evaluation (texeval)*.

[Hacioglu, 2004] Hacioglu, K. (2004). Semantic role labeling using dependency trees. In *International Conference on Computional Linguistics (COLING)*.

[Hagberg et al., 2008] Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA.

[Han et al., 2013] Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). UMBC EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52.

[Havasi et al., 2007] Havasi, C., Speer, R., and Alonso, J. (2007). ConceptNet 3: A flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, pages 27–29. Citeseer.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545.

[Henriksson et al., 2013] Henriksson, A., Skeppstedt, M., Kvist, M., Duneld, M., and Conway, M. (2013). Corpus-driven Terminology Development: Populating Swedish SNOMED CT with Synonyms Extracted from Electronic Health Records. *ACL 2013*, page 36.

[Hindle and Rooth, 1993] Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational linguistics*, 19(1):103–120.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Hovy et al., 2002] Hovy, E., Hermjakob, U., Lin, C.-Y., and Ravichandran, D. (2002). Using knowledge to facilitate pinpointing of factoid answers. In *Proceedings of COLING*.

[Hovy et al., 2013] Hovy, E., Navigli, R., and Ponzetto, S. P. (2013). Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

[Howe et al., 2008] Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., et al. (2008). Big Data: The future of biocuration. *Nature*, 455(7209):47–50.

[Iacobacci et al., 2015] Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2015). SensEmbed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*, pages 95–105, Beijing, China.

[Iqbal et al., 2015] Iqbal, E., Mallah, R., Jackson, R. G., Ball, M., Ibrahim, Z. M., Broadbent, M., Dzahini, O., Stewart, R., Johnston, C., and Dobson, R. J. (2015). Identification of adverse drug events from free text electronic patient records and information in a large mental health case register. *PloS one*, 10(8):e0134208.

[Jarmasz and Szpakowicz, 2004] Jarmasz, M. and Szpakowicz, S. (2004). Roget's thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111.

[Jin et al., 2013] Jin, Y., Kan, M.-Y., Ng, J.-P., and He, X. (2013). Mining scientific terms and their definitions: A study of the ACL anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 780–790, Seattle, Washington, USA. Association for Computational Linguistics.

[Johnston, 1981] Johnston, R. J. (1981). Dictionary of human geography.

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

[Jurgens and Pilehvar, 2016] Jurgens, D. and Pilehvar, M. T. (2016). SemEval-2016 Task 14: Semantic Taxonomy Enrichment. *Proceedings of SemEval*, pages 1092–1102.

[Kezunovic et al., 2013] Kezunovic, M., Xie, L., and Grijalva, S. (2013). The role of Big Data in improving power system operation and protection. In *Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium*, pages 1–9. IEEE.

[Kilgariff, 2006] Kilgariff, A. (2006). Collocationality (and how to measure it). In *Proceedings of the Euralex Conference*, pages 997–1004, Turin, Italy. Springer-Verlag.

[Kiros et al., 2015] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

[Kit and Liu, 2008] Kit, C. and Liu, X. (2008). Measuring mono-word termhood by rank difference via corpus comparison. *Terminology*, 14(2):204–229.

[Klavans and Muresan, 2001] Klavans, J. and Muresan, S. (2001). Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of the AMIA Symposium*, page 324. American Medical Informatics Association.

[Kliegr, 2014] Kliegr, T. (2014). Linked Hypernyms: Enriching DBpedia with targeted hypernym discovery. *Web Semantics: Science, Services and Agents on the World Wide Web*.

[Kozareva and Hovy, 2010] Kozareva, Z. and Hovy, E. (2010). A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of EMNLP*, pages 1110–1118.

[Kozareva et al., 2009] Kozareva, Z., Hovy, E. H., and Riloff, E. (2009). Learning and evaluating the content and structure of a term taxonomy. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 50–57.

[Kumar, 2011] Kumar, D. (2011). The personalised medicine: A paradigm of evidence-based medicine. *Annali dell'Istituto superiore di sanità*, 47(1):31–40.

[Kusner et al., 2015] Kusner, M. J., Sun, Y., Kolkin, N. I., Weinberger, K. Q., et al. (2015). From word embeddings to document distances. In *ICML*, volume 15, pages 957–966.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *NAACL-HLT*.

[Laparra et al., 2010] Laparra, E., Rigau, G., and Cuadros, M. (2010). Exploring the integration of WordNet and FrameNet. In *Proceedings of the 5th Global WordNet Conference (GWC 2010), Mumbai, India*.

[Last et al., 2001] Last, J. M., Spasoff, R. A., Harris, S. S., and Thuriaux, M. C. (2001). *A dictionary of epidemiology*. International Epidemiological Association, Inc.

[Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

[Leacock et al., 1998] Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

[Leal et al., 2012] Leal, J. P., Rodrigues, V., and Queirós, R. (2012). Computing Semantic Relatedness using DBPedia. *1st Symposium on Languages, Applications and Technologies, SLATE 2012*.

[Levy et al., 2015] Levy, O., Remus, S., Biemann, C., Dagan, I., and Ramat-Gan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *NAACL 2015*, Denver, Colorado, USA.

[Lin et al., 2012] Lin, T., Etzioni, O., et al. (2012). No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903. Association for Computational Linguistics.

[Ling et al., 2016] Ling, W., Blunsom, P., Grefenstette, E., Hermann, K. M., Kočiský, T., Wang, F., and Senior, A. (2016). Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609, Berlin, Germany. Association for Computational Linguistics.

[Lonsdale et al., 2002] Lonsdale, D., Ding, Y., Embley, D. W., and Melby, A. (2002). Peppering knowledge sources with salt: Boosting conceptual content for ontology generation. In *Proceedings of the AAAI Workshop: Semantic Web Meets Language Resources*, pages 30–36.

[Luu Anh et al., 2015] Luu Anh, T., Kim, J., and Ng, S. (2015). Incorporating trustiness and collective synonym/contrastive evidence into taxonomy construction. In *Proceedings of EMNLP*, pages 1013–1022.

[Luu Anh et al., 2014] Luu Anh, T., Kim, J.-j., and Ng, S. K. (2014). Taxonomy construction using syntactic contextual evidence. In *Proceedings of EMNLP*, pages 810–819.

[Luu Anh et al., 2016] Luu Anh, T., Tay, Y., Hui, S. C., and Ng, S. K. (2016). Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *EMNLP*.

[Ma and Distel, 2013] Ma, Y. and Distel, F. (2013). Learning formal definitions for SNOMED CT from text. In *Artificial Intelligence in Medicine*, pages 73–77. Springer.

[Malaisé et al., 2004] Malaisé, V., Zweigenbaum, P., and Bachimont, B. (2004). Detecting semantic relations between terms in definitions. In *CompuTerm 2004 - 3rd International Workshop on Computational Terminology*, pages 55–62.

[Mancini et al., 2016] Mancini, M., Camacho-Collados, J., Iacobacci, I., and Navigli, R. (2016). Embedding words and senses together via joint knowledge-enhanced training. *arXiv prepring arXiv:1612.02703*.

[Matuszek et al., 2006] Matuszek, C., Cabral, J., Witbrock, M. J., and DeOliveira, J. (2006). An introduction to the syntax and content of Cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49.

[Mausam et al., 2012] Mausam, Schmitz, M., Bart, R., Soderland, S., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

[Mayr, 1982] Mayr, E. (1982). *The growth of biological thought: Diversity, evolution, and inheritance*. Harvard University Press.

[McHale, 1998] McHale, M. (1998). A comparison of WordNet and Roget's taxonomy for measuring semantic similarity. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, August 16, 1998, Montreal, Canada*. Association for Computational Linguistics, Morristown, NJ, USA.

[McRae et al., 2005] McRae, K., Cree, G. S., Seidenberg, M. S., and McNorgan, C. (2005). Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559.

[Medelyan et al., 2013] Medelyan, O., Witten, I. H., Divoli, A., and Broekstra, J. (2013). Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(4):257–279.

[Melčuk, 1998] Melčuk, I. (1998). Collocations and lexical functions. *Cowie, AP (ed.)*, pages 23–53.

[Mel'čuk, 1996] Mel'čuk, I. (1996). Lexical Functions: A tool for the description of lexical relations in the lexicon. In Wanner, L., editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. Benjamins Academic Publishers, Amsterdam.

[Mendes et al., 2011] Mendes, P. N., Jakob, M., García-silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*.

[Meyer and Gurevych, 2012] Meyer, C. M. and Gurevych, I. (2012). *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. Sylviane Granger and Magali Paquot, editors, Electronic Lexicography, chapter 13, pages 259-291. Oxford University Press.

[Meyer, 2001] Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. *Recent advances in computational terminology*, 2:279.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[Mikolov et al., 2013c] Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

[Miller, 1995] Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

[Moreno et al., 2013] Moreno, P., Ferraro, G., and Wanner, L. (2013). Can we determine the semantics of collocations without using semantics? In Kosem, I., Kallas, J., Gantar, P., Krek, S., Langemets, M., and Tuulik, M., editors, *Proceedings of the eLex 2013 conference*, Tallinn & Ljubljana. Trojina, Institute for Applied Slovene Studies & Eesti Keele Instituut.

[Moro and Navigli, 2013] Moro, A. and Navigli, R. (2013). Integrating syntactic and semantic analysis into the Open Information Extraction paradigm. In *Proceedings of IJCAI*, pages 2148–2154.

[Moro et al., 2014] Moro, A., Raganato, A., and Navigli, R. (2014). Entity Linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

[Morrey et al., 2009] Morrey, C. P., Geller, J., Halper, M., and Perl, Y. (2009). The neighborhood auditing tool: A hybrid interface for auditing the UMLS. *Journal of biomedical informatics*, 42(3):468–489.

[Morris and Hirst, 1991] Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, 17(1):21–48.

[Mrkšić et al., 2016] Mrkšić, N., Séaghdha, D. Ó., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H., and Young, S. (2016). Counter-fitting word vectors to linguistic constraints. *Proceedings of HLT-NAACL*.

[Müller and Gurevych, 2008] Müller, C. and Gurevych, I. (2008). Using Wikipedia and Wiktionary in domain-specific information retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 219–226. Springer.

[Muresan and Klavans, 2002] Muresan, A. and Klavans, J. (2002). A method for automatically building and evaluating dictionary resources. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

[Nakamura and Nagao, 1988] Nakamura, J. and Nagao, M. (1988). Extraction of semantic information from an ordinary english dictionary and its evaluation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 2*, COLING '88, pages 459–464, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Nakashole et al., 2012] Nakashole, N., Weikum, G., and Suchanek, F. M. (2012). PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of EMNLP-CoNLL*, pages 1135–1145.

[Nastase et al., 2010] Nastase, V., Strube, M., Börschinger, B., Zirn, C., and Elghafari, A. (2010). WikiNet: A very large scale multi-lingual concept network. In *LREC*.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

[Navigli and Ponzetto, 2012] Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

[Navigli and Velardi, 2010] Navigli, R. and Velardi, P. (2010). Learning word-class lattices for definition and hypernym extraction. In *ACL*, pages 1318–1327.

[Navigli et al., 2011] Navigli, R., Velardi, P., and Faralli, S. (2011). A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI*, pages 1872–1877.

[Navigli et al., 2010] Navigli, R., Velardi, P., and Ruiz-Martínez, J. M. (2010). An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of LREC'10*, Valletta, Malta.

[Nguyen et al., 2016] Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. (2016). Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proc. of ACL*, pages 454–459.

[Nickel et al., 2012] Nickel, M., Tresp, V., and Kriegel, H.-P. (2012). Factorizing YAGO: Scalable machine learning for Linked Data. In *Proceedings of WWW*, pages 271–280.

[Nielsen, 1994] Nielsen, S. (1994). *The bilingual LSP dictionary: principles and practice for legal language*, volume 24. Gunter Narr Verlag.

[Nielsen, 2010] Nielsen, S. (2010). Specialized translation dictionaries for learners. *Fuertes-Olivera, Pedro A.(ed.)*, pages 69–82.

[Niu et al., 2012] Niu, F., Zhang, C., Ré, C., and Shavlik, J. W. (2012). Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28.

[Nivre, 2005] Nivre, J. (2005). Dependency grammar and dependency parsing. Technical report, Växjö University.

[Noraset et al., 2016] Noraset, T., Liang, C., Birnbaum, L., and Downey, D. (2016). Definition modeling: Learning to define word embeddings in natural language. *arXiv preprint arXiv:1612.00394*.

[Oramas et al., 2016] Oramas, S., Espinosa-Anke, L., Sordo, M., Saggion, H., and Serra, X. (2016). ELMD: An automatically generated entity linking gold standard dataset in the music domain. In *Proccedings of the Language Resources and Evaluation Conference (LREC)*.

[Oramas et al., 2015a] Oramas, S., Gómez, F., Gómez, E., and Mora, J. (2015a). FlaBase: Towards the creation of a flamenco music knowledge base. In *Proceedings of the International Society for Music Information Retrieval Conference*.

[Oramas et al., 2015b] Oramas, S., Sordo, M., Espinosa-Anke, L., and Serra, X. (2015b). A Semantic-based Approach for Artist Similarity. In *Proceedings of the International Society for Music Information Retrieval Conference*, Málaga, Spain.

[Oramas et al., 2014] Oramas, S., Sordo, M., and Serra, X. (2014). Automatic Creation of Knowledge Graphs from Digital Musical Document Libraries. In *Conference in Interdisciplinary Musicology*, Berlin.

[Ostuni et al., 2015] Ostuni, V. C., Di Noia, T., Di Sciascio, E., Oramas, S., and Serra, X. (2015). A Semantic Hybrid Approach for Sound Recommendation. In *Proceedings of the 24th International Conference on World Wide Web (Companion Volume)*, pages 85–86. International World Wide Web Conferences Steering Committee.

[Palmer, 2006] Palmer, M. (2006). Data is the new oil. *ANA Marketing Maestros. CMO News*.

[Panchenko et al., 2016] Panchenko, A., Faralli, S., Ruppert, E., Remus, S., Naets, H., Fairon, C., Ponzetto, S. P., and Biemann, C. (2016). TAXI at SemEval-2016 Task 13: A taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. *Proceedings of SemEval*, pages 1320–1327.

[Pantel and Lin, 2002] Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

[Park et al., 2002] Park, Y., Byrd, R. J., and Boguraev, B. K. (2002). Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

[Passant, 2010] Passant, A. (2010). Dbrec: Music recommendations using DBpedia. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II*, volume 1380, pages 209–224. Springer.

[Pecina, 2008] Pecina, P. (2008). A Machine Learning Approach to Multiword Expression Extraction. In *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 54–57, Marrakech.

[Pedro et al., 2008] Pedro, V., Niculescu, S., and Lita, L. (2008). Okinet: Automatic extraction of a medical ontology from Wikipedia. In *WiKiAI08: a workshop of AAAI2008*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

[Penrose, 1956] Penrose, R. (1956). On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge Univ Press.

[Pham et al., 2015] Pham, N. T., Lazaridou, A., and Baroni, M. (2015). A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of ACL*, pages 21–26.

[Pilehvar and Collier, 2016] Pilehvar, M. T. and Collier, N. (2016). De-Conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*.

[Pilehvar et al., 2013] Pilehvar, M. T., Jurgens, D., and Navigli, R. (2013). Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of ACL*, pages 1341–1351, Sofia, Bulgaria.

[Ponzetto and Strube, 2008] Ponzetto, S. P. and Strube, M. (2008). WikiTaxonomy: A large scale knowledge resource. In *ECAI*, volume 178, pages 751–752.

[Przepiórkowski et al., 2007] Przepiórkowski, A., Spousta, M., Simov, K., Osenova, P., Lemnitzer, L., KuboÅ, V., and Wójtowicz, B. (2007). Towards the

automatic extraction of definitions in Slavic. In *Proceedings ofo the BSNLP workshop at ACL 2007*.

[Pustejovsky et al., 2001] Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M., and Cochran, B. (2001). Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of the Pacific symposium on biocomputing*, volume 7, pages 362–373.

[Raghupathi and Raghupathi, 2014] Raghupathi, W. and Raghupathi, V. (2014). Big data analytics in healthcare: Promise and potential. *Health Information Science and Systems*, 2(1):3.

[Ré et al., 2014] Ré, C., Sadeghian, A. A., Shan, Z., Shin, J., Wang, F., Wu, S., and Zhang, C. (2014). Feature engineering for knowledge base construction. *arXiv preprint arXiv:1407.6439*.

[Rebeyrolle and Tanguy, 2000] Rebeyrolle, J. and Tanguy, L. (2000). Repérage automatique de structures linguistiques en corpus : le cas des énoncés définitoires. *Cahiers de Grammaire*, 25:153–174.

[Reese et al., 2010] Reese, S., Boleda Torrent, G., Cuadros Oller, M., Padró, L., and Rigau Claramunt, G. (2010). Word-sense disambiguated multilingual wikipedia corpus. In *7th International Conference on Language Resources and Evaluation*.

[Reimer and Hahn, 1988] Reimer, U. and Hahn, U. (1988). Text condensation as knowledge base abstraction. In *Artificial Intelligence Applications, 1988., Proceedings of the Fourth Conference on*, pages 338–344. IEEE.

[Reiplinger et al., 2012] Reiplinger, M., Schäfer, U., and Wolska, M. (2012). Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 55–65, Jeju Island, Korea. Association for Computational Linguistics.

[Riedel et al., 2010] Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

[Riedel et al., 2013] Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL*, pages 74–84.

[Rindflesch et al., 2000] Rindflesch, T. C., Tanabe, L., Weinstein, J. N., and Hunter, L. (2000). EDGAR: Extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 517. NIH Public Access.

[Robinson, 1972] Robinson, R. (1972). *Definitions*. Oxford University Press.

[Rodríguez, 2004] Rodríguez, C. (2004). *Metalinguistic Information Extraction from Specialized Texts to Enrich Computational Lexicons*. PhD thesis, Universitat Pompeu Fabra.

[Rodríguez-Fernández et al., 2016] Rodríguez-Fernández, S., Carlini, R., Espinosa-Anke, L., and Wanner, L. (2016). Example-based acquisition of fine-grained collocation resources. In *Proceedings of LREC*, Portoroz, Slovenia.

[Roget, 1911] Roget, P. M. (1911). *Roget's Thesaurus of English Words and Phrases*. TY Crowell Company.

[Roller and Erk, 2016] Roller, S. and Erk, K. (2016). Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2163–2172.

[Roller et al., 2014] Roller, S., Erk, K., and Boleda, G. (2014). Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014*, Dublin, Ireland.

[Russell and Norvig, 1995] Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence - A Modern Approach: The intelligent agent book*. Prentice Hall series in artificial intelligence. Prentice Hall.

[Saggion, 2004] Saggion, H. (2004). Identifying definitions in text collections for question answering. In *Proceedings of LREC*.

[Saggion and Gaizauskas, 2004a] Saggion, H. and Gaizauskas, R. (2004a). Mining on-line sources for definition knowledge. In *17th FLAIRS*, Miami Bearch, Florida.

[Saggion and Gaizauskas, 2004b] Saggion, H. and Gaizauskas, R. (2004b). Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference*, pages 6–7.

[Sampson, 1990] Sampson, G. (1990). Review of computational lexicography for natural language processing by Bran Boguraev and Ted Briscoe - longman 1989. *Computational Linguistics*, 16(2):113–116.

[Santiso et al., 2016] Santiso, S., Casillas, A., Pérez, A., Oronoz, M., and Gojenola, K. (2016). Document-level adverse drug reaction event extraction on electronic health records in spanish. *Procesamiento del Lenguaje Natural*, 56:49–56.

[Santus et al., 2014] Santus, E., Lenci, A., Lu, Q., and Im Walde, S. S. (2014). Chasing hypernyms in vector spaces with entropy. In *EACL*, pages 38–42.

[Sarmento et al., 2006] Sarmento, L., Maia, B., Santos, D., Pinto, A., and Cabral, L. (2006). Corpógrafo V3 From terminological aid to semi-automatic knowledge engineering. In *5th International Conference on Language Resources and Evaluation (LREC'06)*, Geneva.

[Schlichtkrull and Alonso, 2016] Schlichtkrull, M. S. and Alonso, H. M. (2016). MSejrKu at SemEval-2016 Task 14: Taxonomy enrichment by evidence ranking. *Proceedings of SemEval*, pages 1337–1341.

[Schlippe et al., 2010] Schlippe, T., Ochs, S., and Schultz, T. (2010). Wiktionary as a source for automatic pronunciation extraction. In *Interspeech*, pages 2290–2293.

[Schubert, 2006] Schubert, L. (2006). Turing's dream and the knowledge challenge. In *Proceedings of the national conference on artificial intelligence*, volume 21, page 1534. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[Schwartz et al., 2015] Schwartz, R., Reichart, R., and Rappoport, A. (2015). Symmetric pattern based word embeddings for improved word similarity prediction. In *CoNLL*, volume 2015, pages 258–267.

[Seitner et al., 2016] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., and Ponzetto, S. (2016). A large database of hypernymy relations extracted from the web. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference, Portoroz, Slovenia*.

[Shwartz et al., 2016] Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[Sierra, 2009] Sierra, G. (2009). Extracción de contextos definitorios en textos de especialidad a partir del reconocimiento de patrones lingüísticos. *Linguamática*, 1(2):13–37.

[Sierra et al., 2006a] Sierra, G., Alarcón, R., Aguilar, C., and Barrón, A. (2006a). Towards the building of a corpus of definitional contexts. In *Proceeding of the 12th EURALEX International Congress, Torino, Italy*, pages 229–40.

[Sierra et al., 2006b] Sierra, G., Alarcón, R., Aguilar, C., Barrón, A., Benítez, V., and Baca, I. (2006b). Corpus de contextos definitorios: Una herramienta para la lexicografía y la terminología. *Ponencia presentada en el IX Encuentro Internacional de Lingüística en el Noroeste, Hermosillo, México, del*, pages 15–17.

[Sierra et al., 2006c] Sierra, G., Alarcón, R., and Aguilar, C. A. (2006c). Extracción automática de contextos definitorios en textos especializados. *Procesamiento del lenguaje natural*, (37):351–352.

[Sierra et al., 2003] Sierra, G., Medina, A., Alarcón, R., and Aguilar, C. A. (2003). Towards the extraction of conceptual information from corpora. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 691–697.

[Smadja, 1993] Smadja, F. (1993). Retrieving collocations from text: X-Tract. *Computational Linguistics*, 19(1):143–177.

[Snow et al., 2004] Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

[Snow et al., 2006] Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *Proceedings of COLING/ACL 2006*, pages 801–808.

[Soon et al., 2001] Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

[Sordo et al., 2015] Sordo, M., Oramas, S., and Espinosa-Anke, L. (2015). Extracting relations from unstructured text sources for music recommendation. In *Proceedings of Natural Language Processing and Information Systems (NLDB)*, pages 369–382.

[Spackman et al., 1997] Spackman, K. A., Campbell, K. E., and Côté, R. A. (1997). SNOMED RT: A reference terminology for health care. In *Proceedings*

of the AMIA annual fall symposium, page 640. American Medical Informatics Association.

[Speer et al., 2016] Speer, R., Chin, J., and Havasi, C. (2016). ConceptNet 5.5: An open multilingual graph of general knowledge. *AAAI Conference on Artificial Intelligence*.

[Štajner et al., 2015] Štajner, S., Béchara, H., and Saggion, H. (2015). A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Stevenson and Greenwood, 2006] Stevenson, M. and Greenwood, M. A. (2006). Comparing information extraction pattern models. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, IEBeyondDoc '06, pages 12–19, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Storrer and Wellinghoff, 2006] Storrer, A. and Wellinghoff, S. (2006). Automated detection and annotation of term definitions in German text corpora. In *Conference on Language Resources and Evaluation (LREC)*.

[Subramaniam et al., 2003] Subramaniam, L. V., Mukherjea, S., Kankar, P., Srivastava, B., Batra, V. S., Kamesam, P. V., and Kothari, R. (2003). Information extraction from biomedical literature: Methodology, evaluation and an application. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 410–417. ACM.

[Suchanek and Weikum, 2013] Suchanek, F. and Weikum, G. (2013). Knowledge harvesting in the Big-Data era. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 933–938. ACM.

[Suchanek et al., 2007] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *WWW*, pages 697–706. ACM.

[Sudo et al., 2003] Sudo, K., Sekine, S., and Grishman, R. (2003). An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (*ACL 2003*)*, Sapporo, Japan.

[Surdeanu et al., 2012] Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP-CoNLL*, pages 455–465.

[Swartz, 2002] Swartz, A. (2002). Musicbrainz: A semantic web service. *Intelligent Systems, IEEE*, 17(1):76–77.

[Szpektor et al., 2004] Szpektor, I., Tanev, H., Dagan, I., and Coppola, B. (2004). Scaling web-based acquisition of entailment relations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona.

[Tan et al., 2015] Tan, L., Zhang, H., Clarke, C., and Smucker, M. (2015). Lexical comparison between Wikipedia and Twitter corpora by using word embeddings. In *Proceedings of ACL (2)*, pages 657–661, Beijing, China.

[Trimble, 1985] Trimble, L. (1985). *English for Science and Technology: A Discourse Approach*. Cambridge Language Teaching Library.

[Turney and Pantel, 2010] Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

[Velardi et al., 2013] Velardi, P., Faralli, S., and Navigli, R. (2013). OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

[Velardi et al., 2008] Velardi, P., Navigli, R., and D'Amadio, P. (2008). Mining the web to create specialized glossaries. *IEEE Intelligent Systems*, 23(5):18–25.

[Voskarides and Meij, 2015] Voskarides, N. and Meij, E. (2015). Learning to explain entity relationships in knowledge graphs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 564–574.

[Vrandečić and Krötzsch, 2014] Vrandečić, D. and Krötzsch, M. (2014). Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

[Wanner et al., 2006] Wanner, L., Bohnet, B., and Giereth, M. (2006). Making sense of collocations. *Computer Speech and Language*, 20(4):609–624.

[Wanner et al., 2016] Wanner, L., Ferraro, G., and Moreno, P. (2016). Towards Distributional Semantics-based Classification of Collocations for Collocation Dictionaries. *International Journal of Lexicography, doi:10.1093/ijl/ecw002*.

[Wanner et al., 2004] Wanner, L., Ramos, M. A., and Martí, A. (2004). Enriching the Spanish EuroWordNet by collocations. In *LREC*.

[Ward and Barker, 2013] Ward, J. S. and Barker, A. (2013). Undefined by data: A survey of Big Data definitions. *arXiv preprint arXiv:1309.5821*.

[Westerhout, 2010] Westerhout, E. (2010). *Definition extraction for glossary creation: A study on extracting definitions for semi-automatic glossary creation in Dutch*. PhD thesis, University of Utrecht.

[Westerhout and Monachesi, 2007a] Westerhout, E. and Monachesi, P. (2007a). Combining pattern-based and machine learning methods to detect definitions for elearning purposes. In *Proceedings of RANLP 2007 Workshop Natural Language Processing and Knowledge Representation for eLearning Environments*.

[Westerhout and Monachesi, 2007b] Westerhout, E. and Monachesi, P. (2007b). Extraction of Dutch definitory contexts for elearning purposes. *Proceedings of the Computational Linguistics in the Netherlands (CLIN 2007), Nijmegen, Netherlands*, pages 219–34.

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[Wu and Weld, 2010] Wu, F. and Weld, D. S. (2010). Open Information Extraction using Wikipedia. In *Proceedings of ACL*, pages 118–127.

[Wu et al., 2016] Wu, L., Morstatter, F., and Liu, H. (2016). SlangSD: Building and using a sentiment dictionary of slang words for short-text sentiment classification. *arXiv preprint arXiv:1608.05129*.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

[Xu et al., 2010] Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R., and Denny, J. C. (2010). MedEx: A medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24.

[Xu and Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA. ACM.

[Xu et al., 2008] Xu, R., Supekar, K. S., Morgan, A., Das, A. K., and Garber, A. M. (2008). Unsupervised method for automatic construction of a disease dictionary from a large free text collection. In *AMIA*, pages 820–824.

[Yan et al., 2013] Yan, Y., Hashimoto, C., Torisawa, K., Kawai, T., Kazama, J., and De Saeger, S. (2013). Minimally supervised method for multilingual paraphrase extraction from definition sentences on the web. In *HLT-NAACL*, pages 63–73.

[Yang and Callan, 2009] Yang, H. and Callan, J. (2009). A metric-based framework for automatic taxonomy induction. In *Proceedings of ACL/IJCNLP*, pages 271–279. Association for Computational Linguistics.

[Yarowsky, 1992] Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 454–460. Association for Computational Linguistics.

[Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

[Yeh, 2000] Yeh, A. (2000). Comparing two trainable grammatical relations finders. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1146–1150. Association for Computational Linguistics.

[Yeh et al., 2009] Yeh, E., Ramage, D., Manning, C. D., Agirre, E., and Soroa, A. (2009). WikiWalk: Random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics.

[Yu et al., 2014] Yu, L., Hermann, K. M., Blunsom, P., and Pulman, S. (2014). Deep learning for answer sentence selection. *NIPS Deep Learning Workshop*.

[Yu et al., 2015] Yu, Z., Wang, H., Lin, X., and Wang, M. (2015). Learning term embeddings for hypernymy identification. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1390–1397. AAAI Press.

[Zesch et al., 2008] Zesch, T., Müller, C., and Gurevych, I. (2008). Using Wiktionary for computing semantic relatedness. In *AAAI*, volume 8, pages 861–866.

[Zhang et al., 2016] Zhang, C., Shin, J., Ré, C., Cafarella, M., and Niu, F. (2016). Extracting databases from dark data with DeepDive. In *Proceedings of the 2016 International Conference on Management of Data*, pages 847–859. ACM.

[Zhitomirsky-Geffet and Dagan, 2009] Zhitomirsky-Geffet, M. and Dagan, I. (2009). Bootstrapping distributional feature vector quality. *Computational linguistics*, 35(3):435–461.

[Zock and Bilac, 2004] Zock, M. and Bilac, S. (2004). Word lookup on the basis of associations: from an idea to a roadmap. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries*, pages 29–35. Association for Computational Linguistics.

[Zock and Schwab, 2008] Zock, M. and Schwab, D. (2008). Lexical access based on underspecified input. In *Proceedings of the workshop on Cognitive Aspects of the Lexicon*, pages 9–17. Association for Computational Linguistics.