Universität des Saarlandes

Doctoral Thesis

Constructing Lexicons of Relational Phrases

Adam Grycner

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken, März 2017

Degree Colloquium

Dean of the Faculty	Prof. Dr. rer. nat. Frank-Olaf Schreyer
Date	28 June, 2017
Place	Saarbrücken, Germany

Examination Board

Chairman	Prof. Dr. Gert Smolka
Supervisor and Reviewer	Prof. Dr.-Ing. Gerhard Weikum
Reviewer	Prof. Dr. Dietrich Klakow
Reviewer	Prof. Dr. Simone Paolo Ponzetto
Academic Assistant	Dr. Daria Stepanova

Abstract

Knowledge Bases are one of the key components of Natural Language Understanding systems. For example, DBpedia, YAGO, and Wikidata capture and organize knowledge about named entities and relations between them, which is often crucial for tasks like Question Answering and Named Entity Disambiguation. While Knowledge Bases have good coverage of prominent entities, they are often limited with respect to relations.

The goal of this thesis is to bridge this gap and automatically create lexicons of textual representations of relations, namely relational phrases. The lexicons should contain information about paraphrases, hierarchy, as well as semantic types of arguments of relational phrases.

The thesis makes three main contributions. The first contribution addresses disambiguating relational phrases by aligning them with the WordNet dictionary. Moreover, the alignment allows imposing the WordNet hierarchy on the relational phrases. The second contribution proposes a method for graph construction of relations using Probabilistic Graphical Models. In addition, we apply this model to relation paraphrasing. The third contribution presents a method for constructing a lexicon of relational paraphrases with fine-grained semantic typing of arguments. This method is based on information from a multilingual parallel corpus.

Kurzfassung

Wissensbanken sind Schlüsselkomponenten für sprachverarbeitende Systeme. Prominente Vertreter wie zum Beispiel DBpedia, Yago und Wikidata enthalten und organisieren Wissen über benannte Entitäten und deren Relationen zueinander. Das so strukturierte Wissen spielt oft eine zentrale Rolle für Aufgaben wie automatische Fragebeantwortung (engl. Question Answering) oder Disambiguierung von Entitäten. Wissensbanken haben eine gute Abdeckung an Entitäten, sind aber hinsichtlich Relationen oft limitiert.

Das Ziel dieser Dissertation ist es diese Lücke zu schließen und automatisch Lexika zu erstellen, die textuelle Repräsentationen von Relationen, so genannte relationale Phrasen, zur Verfügung stellen. Die Lexika sollten neben Informationen zu Paraphrasen und der Hierarchie relationaler Phrasen auch semantische Typisierung der Argumente einer Relation umfassen.

Diese Dissertation leistet dafür drei wesentliche Beiträge. Der erste Beitrag behandelt die Disambiguierung relationaler Phrasen durch Verknüpfung mit Einträgen des WordNet Lexikons. Diese Verknüpfung ermöglicht es die WordNet Hierarchie auf relationale Phrasen zu übertragen. Im zweiten Beitrag wird eine Methode zur Konstruktion eines Graphen aus Relationen mittels probabilistischer graphischer Modelle vorgeschlagen. Das erzeugte Modell wird darüber hinaus zur Paraphrasierung von Relationen angewandt. Der dritte Beitrag ist eine Methode zur Lexikonkonstruktion relationaler Paraphrasen mit feingranularer semantischer Typisierung der Argumente von Relationen. Diese Methode basiert auf Informationen aus multilingualen parallelen Korpora.

Acknowledgments

In the first place, I would like to thank my advisor, Prof. Gerhard Weikum. Without his outstanding guidance and consistent support, this dissertation would not have been possible. I would also like to thank him for the opportunities he provided. I feel honored to have had this possibility to work with him at the Max-Planck Institute for Informatics.

I wish to thank Prof. Lise Getoor, Jay Pujara, and Jimmy Foulds for the collaboration on the RELLY project. The time I spent during my research visit at the University of California in Santa Cruz is one of the most memorable moments of my PhD.

I thank all my colleagues from the D5 Databases and Information Systems group. In particular, I wish to thank Patrick Ernst for his tremendous help with the German translation of the abstract of the thesis. I also would like to thank Dhruv Gupta, Mohamed Gad-Elrab, Amy Siu, and Aleksandra Piwowarek for proofreading my thesis. Moreover, I would like to express my gratefulness to Mohamed Gad-Elrab, Mohamed Amir Yosef, and Stephan Seufert. You are amazing officemates.

I would also like to thank all my friends I have met during my time in Saarbrücken. Especially, I am very grateful to Konrad Jamrozik for the help he offered me. He had been supporting me since my first day in Saarbrücken. Additionally, I would like to thank Tomas Bastys, Marta Podgórska, Asia Biega, and Mateusz Malinowski for the great time I had with you.

Last but not least, I am very thankful to my father Janusz Grycner, as well as my siblings Alicja and Andrzej, for all of the support and trust they gave me during my PhD.

Contents

Chapter 1

Introduction

1.1 Motivation

For true Artificial Intelligence, a computer would need to be capable of understanding texts written in natural language. The Natural Language Understanding (NLU) community works on creating a system that could face this challenge. Often, the key components of NLU methods incorporate world knowledge from Knowledge Bases (KBs) to reason about the text. KBs such as YAGO (Suchanek et al., 2007), DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), or Wikidata (Vrandečić and Krötzsch, 2014) are rich in information on Named Entities, relations between them and their semantic types. This knowledge is a great asset when solving problems such as Named Entity Disambiguation, Question Answering, or Textual Entailment.

Particularly helpful is the information on relations between Named Entities. Knowing that the phrase "was born" corresponds to the KB relation *bornIn* could help answer the question "Where was Chopin born?" with the KB as a source of answers. With the knowledge that "'s birthplace" is a synonym of "was born," the paraphrased question "What's the birthplace of Chopin?" could be answered without directly knowing that "'s birthplace" corresponds to the KB relation *bornIn*. However, synonymy is not the only useful relationship between relations which could help in solving NLU problems.

Additionally, a comprehensive NLU system should know which relations are more specific and which relations are more general. The sentence "Maria Curie was a spouse of Pierre Curie" allows the system to answer the question "Who was Maria Curie married to?", because "was a spouse of" is a synonym of "was married to." In contrast, the sentence "Chopin was in a relationship with George Sand" should not be used to answer a question about Chopin's marital status. On the other hand, the aforementioned two sentences are sufficient to give an answer for the questions about Chopin's and Curie's romantic relationships.

While KBs have a good coverage of Named Entities, they are often limited with respect to relations. YAGO contains over 100 relations not organized into any hierarchy. In DBPedia and Freebase, the number of relations exceeds 5,000. Moreover, the relations are neither organized into clusters of synonyms nor into a hypernymy hierarchy. In Freebase and Wikidata the relations are organized into thematic groups (e.g., Person, Relationships, Sportspeople, Road). However, organizing relations into thematic groups have a limited impact on textual reasoning.

1.2 Challenges

To solve the problem of a limited number of relations in KBs, Open Information Extraction (OpenIE) (Banko et al., 2007) techniques were introduced. OpenIE methods acquire facts from texts written in natural language. Unlike standard algorithms for Knowledge Base Construction (KBC), OpenIE methods are not limited by a set of specified relations. Any word sequence can represent a relation. Apart from performing relation extraction, some of the OpenIE techniques organize the extracted relations into a taxonomy. However, the hierarchy construction and clustering methods presented in OpenIE have their limitations.

In PATTY (Nakashole et al., 2012), the clustering and hierarchy construction approaches are based on the approximate overlap and inclusion of sets of arguments of the relations. Two relations are clustered together if their sets of arguments, extracted from a text, are almost the same as shown in Figure 1.1. A similar approach was presented for the hierarchy construction of relations. Relation A is found to be more specific than relation B if the set of arguments of A is approximately included in the arguments set of B. This organization of phrases is depicted in Figure 1.2.

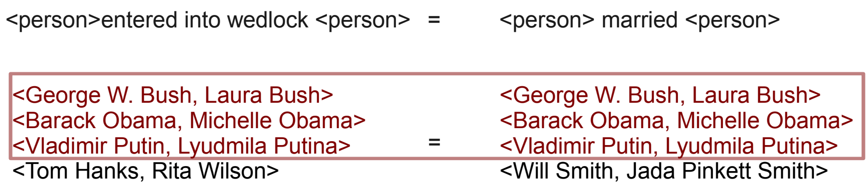


Figure 1.1: Example of arguments overlap

This approach has led to a hierarchy which is very sparse and consists of hundreds of thousands of root relations. For 350,000 relational phrases, only approximately 8,000 hierarchical links were found. This situation is caused by the sparsity of general relations in natural language texts. The system based on arguments comparison does not have enough data to reason about the general relations.

Another important problem related to relation hierarchy construction is scalability. There are hundreds of thousands of relations in texts and the system

<musician> composed <musical_composition> ⊂ <person> created <entity>

<Chopin, Funeral March>		<Chopin, Funeral March>
<Mozart, Eine kleine Nachtmusik>		<Mozart, Eine kleine Nachtmusik>
<Beethoven, Für Elise>	⊂	<Beethoven, Für Elise>
<Vivaldi, The Four Seasons>		<Shakespeare, Romeo and Juliet>
		<Picasso, Guernica>
		<Leonardo da Vinci, The Last Supper>
		<Gaudí, Sagrada Família>

Figure 1.2: Example of subsumption relationship

should be able to construct a hierarchy for such a large number of relations, and should take into consideration the global structure of the output hierarchy.

Finally, the system for organizing relations should be able to cope with relation ambiguity. The textual pattern "covered" could mean recording a new version of a song or creating an article about some event by a journalist. The meaning of this textual pattern is highly dependent on the context. Usually, these patterns are disambiguated by enriching them with semantic types of arguments. However, the type system should have sufficient granularity, so that the textual pattern could be properly disambiguated. High-level semantic types (*Person*, *Location*, *Organization*, *Event*) for relation arguments would not be specific enough to separate two meanings of the relation "cover."

1.3 Problem Statement

In this thesis, we concentrate on the problem of constructing a taxonomy of binary relational phrases. The taxonomy, similarly as in WordNet, should contain information about textual representations of relations, namely relational phrases, and relationships between them. Among those relationships, we focus on synonymy (finding relations of similar meaning) and hypernymy (detecting which relations are more specific and which relations are more general). Moreover, we address the problems of finding counterparts of relational phrases in other taxonomies (e.g., WordNet) and detecting fine-grained semantic types of relations' arguments.

1.4 Thesis Contribution

With this thesis we make the following contributions:

- **HARPY:** The first contribution of the thesis is HARPY (Grycner and Weikum, 2014) – a system for aligning relational phrases with WordNet verb senses. These alignments can help with building a hierarchy of relational phrases. Collections of relational paraphrases have been automatically constructed from large text corpora, as a WordNet counterpart for the realm of binary predicates and their surface forms. However, these resources fall short in their coverage of hypernymy links (subsumptions) among the synsets of phrases. This work closes this gap by computing a high-quality alignment between the relational phrases of the PATTY taxonomy, one of the largest collections of this kind, and the verb senses of WordNet. To this end, we devise judicious features and develop a graph-based alignment algorithm by adapting and extending the SimRank random-walk method. The resulting taxonomy of relational phrases and verb senses, coined HARPY, contains 20,812 synsets organized into a *Directed Acyclic Graph (DAG)* with 616,792 hypernymy links. Our empirical assessment indicates that the alignment links between PATTY and WordNet have high accuracy, with a *Mean Reciprocal Rank (MRR)* score 0.7 and *Normalized Discounted Cumulative Gain (NDCG)* score of 0.73. As an additional extrinsic value, HARPY provides fine-grained lexical types for the arguments of verb senses in WordNet.

- **RELLY:** The second contribution builds on top of HARPY. RELLY (Grycner et al., 2015) concentrates on the scalable construction of a high-precision graph of relational phrases. Relational phrases (e.g., "got married to") and their hypernyms (e.g., "is a relative of") are central for many tasks including Question Answering, Open Information Extraction, Paraphrasing, and Entailment Detection. This has motivated the development of several linguistic resources (e.g. DIRT, PATTY, and WiseNet) which systematically collect and organize relational phrases. These resources have demonstrable practical benefits, but are each limited due to noise, sparsity, or size. We present a new general-purpose method, RELLY, for constructing a large hypernymy graph of relational phrases with high-quality subsumptions using collective probabilistic programming techniques. Our graph induction approach integrates small high-precision knowledge bases together with large automatically curated resources, and reasons collectively to combine these resources into a consistent graph. Using RELLY, we construct a high-coverage, high-precision hypernymy graph consisting of 20,000 relational phrases and 35,000 hypernymy links. Our evaluation indicates a hypernymy link precision of 78%, and demonstrates the value of this resource for a document-relevance ranking task.

- **POLY:** Finally, the third contribution offers an algorithm for clustering of relational phrases using multilingual information (Grycner and Weikum, 2016). Language resources that systematically organize paraphrases for binary relations are of great value for various NLP tasks and have recently been advanced in projects like PATTY, WiseNet, and DEFIE. This work presents a new method for building such a resource and the resource itself, called POLY. Starting with a very large collection of multilingual sentences parsed into triples of phrases, our method clusters relational phrases using probabilistic measures. We judiciously leverage fine-grained semantic typing of relational arguments for identifying synonymous phrases. The evaluation of POLY shows significant improvements in precision and recall over the prior works on PATTY and DEFIE. An extrinsic use case demonstrates the benefits of POLY for Question Answering.

1.5 Thesis Outline

The outline of the rest of this dissertation is as follows. Chapter 2 introduces necessary concepts. It also gives an overview of tasks such as Relation Extraction, OpenIE, Relation Clustering, and Automatic Hierarchy Construction. Additionally, we discuss potential applications for lexicons of relational phrases. Chapter 3 presents HARPY, a system for finding alignments between relational phrases and WordNet verb senses. Chapter 4 introduces RELLY, a scalable system which uses Probabilistic Soft Logic (PSL) to generate a hierarchy of relational phrases. In Chapter 5, we investigate the applicability of PSL in the relation clustering domain. Chapter 6 presents POLY, a system and a resource which offers semantically typed clusters of relational phrases. The clustering and the semantic typing stages are influenced by multilingual information. Finally, Chapter 7 provides concluding remarks and discusses potential future research directions.

Chapter 2

Related Work

In this chapter, we introduce the terminology related to relational phrases and their paraphrases. We describe sources of these phrases, methods for discovering relationships between them and potential applications.

2.1 Terminology

Relational phrases and their paraphrases are the main focus of this thesis. Before going into details let us first define what relational phrases are, how they are constructed and what kinds of relationships between these phrases are of interest to us.

2.1.1 Relational Phrases

Relational phrases are natural language phrases that describe some relation. These phrases can describe some linguistic relation (e.g., *is a subclass of*, *is a synonym of*), a relation from a Knowledge Base (KB) (e.g., *bornIn*, *actedIn*) or some relation that is not defined in a KB yet. For example, the group of phrases: "composed a song to commemorate his home country," "misses his home," "would like to return to," "pined for" implicitly describes a relation between an emigrant musician and his home country.

In this thesis, we discuss mainly binary relations – relations that can occur between two common noun phrases or named entities (like relation *was born in* in Example 1). The sentence where the relation is extracted from is often referred to as *relation instance*.

Example 1. *Chopin was born in a village in Poland.*

2.1.2 Relational Phrases Representation

There are multiple choices when it comes to representing a relation by a relational phrase. A relational phrase can be formed as a subsequence of words, which occur in a sentence (Example 2). We will refer to this representation as *word sequence* representation.

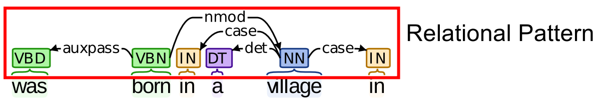
Example 2. *was born in*

A relational phrase can also be represented as a *lifted sequence of words.* In the lifted representation, words of a relational phrase can be replaced by names of classes they belong to (e.g., part of speech tags, wildcards, or ontological types (Nakashole et al., 2012)). With this representation, the relation from Example 1 can be transformed into the form in Example 3. Lifted sequence of words gives an interpretable representation. Moreover, it groups together relational phrases with small word variances.

Example 3. *was born [Preposition] a [LOCATION] in*

In the *dependency parse tree path* representation, a relational phrase is depicted by a path in a dependency parse tree. The extracted relation from Example 1 could be represented by a path in Example 4.

Example 4.



A different option is a *latent representation.* Relations could then be represented by a real-valued vector. Vector representations are less interpretable but can be used in downstream applications such as analogy discovery (Mikolov et al., 2013). In Chapters 3, 4, 5, we investigate relational phrases with the *lifted sequence of words* representation, whereas in Chapter 6 we consider the *word sequence* representation.

2.1.3 Semantic Types of Arguments

The representation of relational phrases can be enhanced with information about the kind of arguments the relation can take. This can be achieved by adding *semantic types* of relation arguments. Similarly as with relational phrases, this information can be represented in many ways.

In the first option, we could omit the information about semantic types of relation arguments. This approach brings a risk of merging multiple meanings under one textual representation. For example, if we ignore semantic type information, the textual pattern "played in" will express both acting in a movie and participating in a sport competition.

The second option is to provide types of arguments that usually occur with a relation. In this case, the relation extracted from the sentence in Example 1 could have a form of "*<person>* was born in *<country>*," where *<person>* and *<country>* are textual representations of classes of domain and range arguments of the relation. There are multiple design choices when it comes to the type system. Popular choices are fixed sets of manually selected types (e.g. *Person*, *Location*, *Organization*), the full set or subset of the WordNet noun hierarchy (Fellbaum, 1998), or the WordNet hierarchy combined with the Wikipedia category system (Suchanek et al., 2007; Navigli and Ponzetto, 2012).

A more expressive but less interpretable option of representing the argument types is the distributional representation (Moro and Navigli, 2012; Bovi et al., 2015). In this representation, the types of an argument are represented as a vector of classes with real-valued scores, as shown in Example 5. The scores represent probabilities of an argument of a class occurring with the relation.

Example 5. *<person:0.95, company:0.05> was born <country:0.75, city:0.25>*

The last option for arguments semantic type representation is the latent representation. Types are represented as real-valued vectors, which correspond to arguments in a vector space model. In all Chapters, we consider relational phrases with single types.

2.1.4 Synonymy

The core interest of this thesis is finding the relationships between relational phrases. One of them is the *synonymy* relationship that indicates semantic equality between two relational phrases. In this thesis, we refer to relational phrases of the same meaning as *synonyms* or *relational paraphrases* interchangeably. The relational paraphrases can be represented as a list of pairs or they can be represented as clusters of relational phrases of a similar meaning (with soft or hard membership). A sample pair of relational paraphrases is shown in Example 6 and a sample cluster of relational paraphrases is presented in Example 7.

Example 6. *<person>'s birthplace is <city>* $\approx$ *<person>'s born in <city>*

Example 7. *<person> was born in; 's birthplace was; came from; originated from; was from <country>*

In Chapter 5 and 6, we will investigate finding synonyms between relational phrases.

2.1.5 Hypernymy

The second interesting relationship between relational phrases is the *hypernymy* relationship. This relationship, in the verb sense domain, is also referred to as a troponymy relationship. Two relational phrases are in a hypernymy relationship if the first phrase (hyponym) is more specific than the second one (hypernym). Example 8 depicts this situation.

Example 8. *<person> is a spouse of <person>* (hyponym) → *<person> is in a relationship with <person>* (hypernym)

Moreover, the hypernymy relationship usually implies that arguments that co-occur with the hyponym phrase should also co-occur with the hypernym phrase. Figure 1.2 shows an example of this phenomenon. This relationship is strongly connected with *entailment* (one can infer the second relational phrase from the first) and *subsumption* (first relational phrase occurs with a subset of argument pairs of the second relational phrase). In Chapters 3 and 4, we will concentrate on the hypernymy relationship between relational paraphrases.

2.2 Organizing Relational Phrases

After introducing relational phrases and related concepts, we can now explain how the phrases can be extracted and organized. In Section 2.2.1 (Relation Extraction), we describe methods that extract relational phrases from a text and classify them into a fixed set of representable relations. In Section 2.2.2 (Open Information Extraction), we report systems for extracting relational phrases without the classification step. The set of representable relations is unbounded in this case. Next, we investigate the methods for organizing relational phrases by performing relation clustering (Section 2.2.3) and creating a subsumption graph of relations (Section 2.2.4).

2.2.1 Relation Extraction

In the first scenario, we consider extracting relational phrases for a fixed set of relations. We define Relation Extraction as a task of extracting and classifying relational phrases from a text. The classification step maps an extracted relation to a predefined set of relations. For example, a Relation Extraction algorithm could extract a phrase "'s birthplace was" from a sentence "Chopin's birthplace was Poland" and classify it as a *bornIn* relation in YAGO KB.

The source of representable relations can be divided into two categories. Relations can come from the manually defined sets or from the semi-automatically constructed KBs. In the first category, representable relations are limited to the manually defined sets, such as WordNet (Fellbaum, 1998) or the ACE RDR dataset[1]. In this case, relations usually represent linguistic relations (e.g., hypernymy, synonymy, meronymy, or causality) or handcrafted relations (e.g., agent, instrument, location, object, or possessor). One of the most notable extraction methods using a limited manually defined set of relations is the work by Hearst (1992), where the system, in a text, was looking for relational

[1] http://itl.nist.gov/iad/mig/tests/ace/2004/

phrases describing *IS-A* relation. In the second category, relations come from KBs (Suchanek et al., 2007; Auer et al., 2007; Bollacker et al., 2008). In this category, the relations often represent relations obtained semi-automatically from web sources such as Wikipedia (e.g., *wasBornIn*, *graduatedFrom*, *participatedIn*).

Methods for relational phrases extraction for a fixed set of relations can also be divided into two categories. The methods can be supervised or semi-supervised. First of all, the extraction algorithm can be fully supervised. The training data is then fully annotated like in (Zhao and Grishman, 2005), where the authors used the ACE RDR dataset to train a relation detection and extraction system. The training and test data consisted of 23 different kinds of relations and the training was performed using Support Vector Machines (SVM). More recent works for relation extraction and classification employed deep neural networks (Zeng et al., 2014; Xu et al., 2015). These works leverage datasets provided by SemEval-2010 Task 8 (Hendrickx et al., 2010). However, these two approaches concentrated on a much smaller set of possible relations. Systems by Weston et al. (2013) and Wang et al. (2014) created relation embeddings and, based on them, they mapped extracted textual patterns to Freebase (Bollacker et al., 2008) relations.

On the other hand, the semi-supervised methods start with a few high-precision seed training examples. Suchanek et al. (2006) used a list of famous birthdays and information included in WordNet to annotate a text and create a training data. The authors employed features extracted from dependency parse trees of potential relational phrases to classify whether a relational pattern expresses a given target semantic relation. The classification step was based on SVM and k-nearest-neighbors classifiers. The list of target relations included *birthdate*, *synonymy*, and *instanceOf*. Bunescu and Mooney (2007) used a small amount of positive and negative examples of pairs of named entities that occur with a particular relation. Based on this small initial dataset, the relation instances were extracted. Finally, the extracted relation instances were used to train the SVM classifier. One of the first approaches for semi-supervised relation extraction were presented in DIPRE (Brin, 1998) and Snowball (Agichtein and Gravano, 2000) systems. The authors used a bootstrapping approach with a few high-precision seed relational patterns. A small number of manually selected seed relation instances were used to extract, from a large corpus, new relational patterns, which were used to extract even more relation instances. KnowItAll (Etzioni et al., 2005), for learning how to extract relation instances, utilized two kinds of training data, namely (1) words that describe the class of the arguments, and (2) a small set of seed extraction patterns. Bootstrapping was also used to gather semantic types of relations arguments (Kozareva and Hovy, 2010). In self-supervised approaches, automatic heuristics generated labeled data for training the relation extractor. For example, NELL (Carlson et al., 2010; Mitchell et al., 2015) is a web-scale self-supervised learning system that runs continuously. A small handcrafted ontology containing argument types and relations is used to constrain and steer the relation instance extraction process.

Unlike previous learning algorithms that use either a small or manually created set of initial training data, distantly-supervised methods (a sub-category of semi-supervised methods) utilize a huge amount of heuristically created training examples. The initial training examples are often obtained from KBs like Freebase, DBpedia, or YAGO, which can be seen as a source of supervision. In (Mintz et al., 2009), the authors started with 102 relations from Freebase and 17,000 seed relation instances. The pairs of entities, which occur with a Freebase relation, were mapped to sentences in a large unlabeled corpus. From those sentences, textual features were extracted to train a relation classifier. The algorithm combined both supervised (training a relation classifier) and unsupervised (extracting relational phrases from an unlabeled corpus) approaches. PROSPERA (Nakashole et al., 2011), on top of bootstrapping and distant supervision, used logical consistency rules to improve precision of the relation instance extraction process. PROSPERA extended SOFIE (Suchanek et al., 2009) but is more scalable and provides higher recall due to a pattern generalization approach. The other system by Riedel et al. (2010), on top of distant supervision, incorporated probabilistic graphical models to improve the quality of the results. This idea was extended by Hoffmann et al. (2011) and Surdeanu et al. (2012) to a multi-instance multi-label learning framework (where an extracted relational phrase could represent multiple KB relations). Another improvement was proposed by Min et al. (2013) where the quality of distant supervision training data was improved by reducing false negative examples. A different kind of extension was presented by Yao et al. (2010). There, in addition to distant supervision, compatibility of relation types and arguments types was considered by jointly performing relation extraction and entity identification. More recently, the authors of (Zeng et al., 2015) followed recent trends and combined deep neural network model with distant supervision for relation extraction.

Finally, although most of the mentioned approaches concentrate on the English language, there are still some systems that extract multilingual relations. For instance, one of the few examples is BOA (Gerber and Ngomo, 2012) – a system that use a bootstrapping approach to find English and German relational phrases.

2.2.2 Open Information Extraction

Open Information Extraction (OpenIE) methods process natural language texts to produce triples of a surface form for arguments and relational phrases of binary relations. For instance, an OpenIE method could extract the tuple (*Chopin*, *was born in*, *Poland*) from the sentence in Example 1. The paradigm of OpenIE was introduced to overcome the constraint of a fixed set of representable relations. In this paradigm, any phrase can represent some relation, even if that relation does not exist in any schema of any KB. Unlike the methods presented in Section 2.2.1, the OpenIE systems are mostly unsupervised, that is there is no initial set of relations and no training data at all.

The OpenIE paradigm was introduced with the TextRunner system (Banko et al., 2007). TextRunner is a fast, highly scalable OpenIE system. The scalability and high speed of the system come from using only shallow linguistic processing and performing only a single-pass over the source corpus. After the extraction, relations and their arguments are assigned a score based on the redundancy in a text. A similar approach was presented in WOE^{pos} (Wu and Weld, 2010). Additionally, the authors utilized heuristic matches between Wikipedia infobox attribute values and their corresponding sentences to construct training data.

Another improvement in the OpenIE domain was presented in the ReVerb system (Fader et al., 2011). One of the problems in OpenIE is the extraction of incoherent and uninterpretable relational phrases. The solution proposed in ReVerb was to define syntactic constraints on top of possible extracted relational phrases. The relational phrase must be mediated by a verb, that is have a form "VERB X* PREP." Moreover, the authors used corpus count statistics to filter out non-informative extractions.

The next step in the advancement of OpenIE was presented in OLLIE (Mausam et al., 2012). That work expanded the syntactic scope of relation phrases by allowing not only verb mediated relational phrases like in ReVerb, but also noun or adjective mediated ones. Furthermore, the system analyzed the context around an extraction which allows the filtering out of non-factual extractions.

The aforementioned systems used the shallow linguistic processing for the purposes of OpenIE. WOE^{parse} (Wu and Weld, 2010) introduced dependency-based parsing in the OpenIE domain. In (Xu et al., 2013), dependency parse trees were used to (1) determine whether there is a relation between two named entities, and (2) determine if the extracted words of a relational phrases form an appropriate relation. The method was presented in two versions. The first version was supervised and used SVM with dependency tree kernels. The second was unsupervised and replaced SVM with handcrafted rules on dependency parse trees. Similarly, DepOE (Gamallo et al., 2012) and EXEMPLAR (de Sá Mesquita et al., 2013; Schmidek and Barbosa, 2014) used manually defined extraction rules on top of dependency parse trees. Moreover, DepOE (Gamallo et al., 2012) enables the extraction not only in the English, but also in Spanish, Portuguese, and Galician languages.

OntExt (Mohamed et al., 2011), PATTY (Nakashole et al., 2012), and WiseNet (Moro and Navigli, 2012; Moro and Navigli, 2013) used semantic types of relation arguments in the domain of OpenIE. A relational phrase is represented not only by a textual pattern but is also enriched with information about classes of relation arguments. This semantic type information can steer the extraction.

Linguistic theories about sentence structure were also employed to help OpenIE. ClausIE (Corro and Gemulla, 2013), similarly as in OLLIE, overcame the constraint of representing relations only with the "VERB X* PREP" pattern. The authors proposed a clause-based approach for OpenIE. Using linguistic knowledge about the grammar of the English language, they first detected one of the twelve possible clauses (sentence templates) and then based on that information the extraction was performed. The authors of CSD-IE (Bast and Haussmann, 2013; Bast and Haussmann, 2014) explored Contextual Sentence Decomposition (CSD) for the purposes of OpenIE. The goal of CSD is to compute all parts of a sentence that semantically belong together. Later, based on the discovered sentence sub-parts, the triples of two arguments and a relational phrase were generated. A similar idea of decomposing a sentence into smaller pieces, which simplifies the extraction of relations with arguments, was presented in (Angeli et al., 2015). Additionally, after the sentence simplification and relation/argument extraction, the authors used natural logic inference over the short sentence sub-parts to determine the maximally specific arguments for each argument-relation-argument triple.

An extension over simple binary relational phrases was introduced in NESTIE (Bhutani et al., 2016). The authors proposed a nested representation to extract higher-order relations. Extracting higher-order relations allows for the inclusion of context under which the assertions are correct and complete. It also helps to more accurately reflect the meaning of the original sentence.

So far, texts written in the English language are the main focus of OpenIE systems. Other languages received much less attention. The already mentioned DepOE (Gamallo et al., 2012) and ArgOE (Gamallo and García, 2015) allowed performing OpenIE for the English, Spanish, Portuguese and Galician languages based on the dependency parsing and language-independent set of rules. The recent work of (Faruqui and Kumar, 2015) extracted relational phrases from Wikipedia in 61 languages using cross-lingual projection. The extraction was performed by combining the OLLIE (Mausam et al., 2012) tool with a translation system and projecting multilingual sentences back to English. The aforementioned systems are designed to be language independent. One of the examples of a system that is tailored towards language different than English is ExtrHech (Zhila and Gelbukh, 2014), which is developed for the Spanish language. Kim et al. (2011) used Korean-English parallel corpora for cross-lingual projection for the purposes of OpenIE. Systems created for the English language can also be reused for other languages with limitations. For example, the authors of (Falke et al., 2016) did not start from scratch with a system for a foreign language but translated OpenIE extraction rules from English to German.

2.2.3 Relation Clustering

In Subsections 2.2.1 and 2.2.2, we described what kinds of relational phrases can be extracted and how extraction can be achieved. The next important question is how we can organize this knowledge of extracted relations. The first step in organizing relational phrases is performing the relation clustering. With relation clustering, we can group together relational phrases of the same or similar meaning or, in other words, representing the same relation. The grouping can be performed by clustering textual representations of relations (relational phrases) or by clustering contexts where the relations occur (relation instances). This step is strongly connected with OpenIE because putting relational phrases into clusters of synonyms can help with describing the meaning of these ambiguous phrases. As mentioned earlier in Section 2.1.4, the dataset of clusters of synonymous relational phrases could also be seen as a resource of relational paraphrases.

The first notable effort to build up a resource for relational paraphrases was DIRT (Lin and Pantel, 2001a). DIRT, for finding relational paraphrases, used the extended version of Harris' Distributional Hypothesis. This version states that paths in the dependency parse trees linking the same sets of words tend to have the similar meaning. The similarity metric used to detect relational paraphrases was based on the pointwise mutual information between arguments of the considered relations. In the end, the paths in the dependency parse trees represent relational phrases. VerbOcean (Chklovski and Pantel, 2004) extended the approach presented in DIRT with information about relationships between verbs, such as similarity, strength, antonymy, enablement, and happens-before.

The classical clustering approaches from the Data Mining domain were also used for the purposes of the relation clustering. The authors of (Hasegawa et al., 2004) applied both Harris' Distributional Hypothesis and Hierarchical Agglomerative Clustering (HAC). They looked for pairs of named entities that co-occur with each other. Then, they defined the similarity between pairs of named entities as the cosine similarity between their contexts in the tf*idf vector space model. The context was defined as words that occur between two named entities. Finally, the authors used HAC with the aforementioned similarity measure to obtain clusters of pairs of named entities. The most frequent words in the named entities contexts represented both the label of a cluster and a relational phrase. A similar improved idea was introduced in SONEX (de Sá Mesquita, 2012). The authors extended the tf*idf model with domain frequency (df). Furthermore, HAC was accelerated with a sampling technique. A more restricted version was presented in OntoExt (Mohamed et al., 2011). Here, first of all, clusters of relations were limited to relations whose argument types belong to a predefined set of semantic types. Secondly, OntoExt used the k-means clustering algorithm, which requires providing upfront the number of output clusters. As the last step, OntExt used a classifier to find the label of a cluster, namely the most representative and semantically valid relational phrase.

Relation clustering can also be performed with Probabilistic Graphical Models. A further extension of aforementioned ideas was presented in RESOLVER (Yates and Etzioni, 2009). Here, the synonymity of relational phrases and the synonymity of relation arguments was modeled jointly. The second improvement was a probabilistic relational model that incorporates both information about distributional similarity as well as textual similarity.

The usage of Latent Dirichlet Allocation (LDA) was an another improvement proposed in (Yao et al., 2011; Yao et al., 2012). The authors employed a latent topic model with signals like relation argument words, words in the context, document theme and sentence theme to group similar relational phrases together. Moreover, the model allowed for putting a single relational phrase to multiple clusters. As a result, that approach was able to model polysemy among relational phrases – a phenomenon where a single relational phrase can have multiple meanings.

A related approach to latent topic models is Matrix Factorization. The authors of (Takamatsu et al., 2011) introduced matrix factorization for the purposes of finding synonymous relational phrases. The matrix columns represented the extracted relations and the rows represented features describing these relations – unigrams, bigrams and entity tags in the context. The matrix factorization algorithm was based on Latent Semantic Indexing (LSI). This approach worked well for small corpora. The next step, Universal Schema (Riedel et al., 2013) enabled matrix factorization for the relation extraction on a bigger scale. In this setup, rows in the matrix represented disambiguated pairs of relation arguments (named entities) and columns represented either OpenIE relational phrases or Freebase (Bollacker et al., 2008) relations. The outcome of the factorization generated similarity scores between relational phrases and Freebase relations. The produced scores were asymmetric; the similarity between relation A and B can be different than between B and A. Later, the idea from Universal Schema was extended with additional contextual information (Petroni et al., 2015) like pairs of co-occurring entities, semantic types of relation arguments or documents topics.

The combination of the fine-grained typing and the relational phrase clustering was proposed in PATTY (Nakashole et al., 2012). Fine-grained typing, which comes from the YAGO KB (Suchanek et al., 2007), helps to disambiguate relational phrases and handle polysemy – a situation where the same textual pattern can have multiple meanings. For example, "*<person>* covered *<song>*" (recording a new version of a song) and "*<person>* covered *<event>*" (reporting an event in a newspaper) can be separated and put into two different clusters. The clustering in PATTY was based on the textual similarity and argument co-occurrence. In a similar work by Moro and Navigli (2012), called WiseNet, the clustering was performed before assigning the fine-grained semantic types. In their case, the clustering was based on Harris' Distributional Hypothesis, where the left-hand side and the right-hand side context was modeled separately. Unlike in PATTY, the types of arguments of relations were not modeled by a single semantic type, but by a distribution of semantic types (in WiseNet represented with Wikipedia categories). The extension of WiseNet (Moro and Navigli, 2013) performed relational phrase clustering using distributed soft kernel k-medoids algorithm, which allowed a single relational phrase to belong to different clusters with different strength. The metric used in the clustering algorithm was based on syntactic and semantic features. A more recent work for the clustering of semantically typed relational phrases was presented in DefIE (Bovi et al., 2015). DefIE restricted relational phrases to those that come from well-formed sentences from the definitions dictionary. Similarly as before, the arguments types were represented by a distribution of semantic types. However, the clustering here came from the Word Sense Disambiguation (WSD) of relational phrases and the alignment of these phrases with BabelNet (Navigli and Ponzetto, 2012).

Similarly as in Section 2.2.1 (Relation Extraction) and Section 2.2.2 (Open Information Extraction), there are far fewer examples of methods for relation clustering that work with multilingual data. One of such methods is a work by Lewis and Steedman (2013), where semantically equivalent English and French relational phrases were clustered together based on the relation arguments' co-occurrence in French and English monolingual corpora. Another work, which considered multilingual relational phrases, is an extension of the Universal schema project (Verga et al., 2016). In that extension, English and Spanish relational phrases were mapped to Freebase relations and the similarity score among relational phrases was also computed.

2.2.4 Hierarchies of Relations

Organizing relations into clusters of synonyms, presented in Section 2.2.3, has a potential in helping with tasks like paraphrase generation and paraphrase detection. However, clustering leads to a resource that provides only symmetric relationships between relational phrases. There exist application domains where also asymmetric information could be useful. Constructing a hierarchy of relational phrases could help with storing the asymmetric relationships between relational phrases (entailment, hypernymy/hyponymy, or subsumption).

The easiest approach to organizing relations into a hierarchy is to perform it manually. An example of such endeavor is WordNet (Fellbaum, 1998), where verb senses, which can also be interpreted as short relations, were manually organized into a hypernymy/hyponymy hierarchy. Other examples of manually curated hierarchical taxonomies of verbs or verb classes are included in VerbNet (Kipper et al., 2008) and FrameNet (Baker et al., 1998).

A semiautomatic method for the verb taxonomy construction was presented in VerbOcean (Chklovski and Pantel, 2004). Among semantic relationships like similarity, antonymy, enablement, the resource offers also the *happens-before* relationship. With this extra relationship, verbs can be organized into a hierarchical structure.

So far, we discussed hierarchies that were constructed either in a manual or semiautomatic way. A significant approach for automatic construction of relation hierarchy is global learning of *entailment graphs* (Berant et al., 2010; Berant et al., 2011; Berant et al., 2012b; Berant et al., 2012a; Berant et al., 2015). Organizing relations in terms of entailment is one way of constructing a relation hierarchy. The first paper in this line of work (Berant et al.,

2010) used a local classifier for the entailment detection between pairs of relations, as well as a global optimization criterion executed with Integer Linear Programming (ILP). With ILP, a transitivity constraint was imposed on the outcome hierarchy. An extension of this work (Berant et al., 2011) computed entailment relationships between semantically typed relations using again ILP, but with an additional graph decomposition. Later, even more computational optimizations were introduced (Berant et al., 2012a; Berant et al., 2015).

The previously discussed resources concentrated mostly on short or single-verb relational phrases. An extension of a hierarchy of relational phrases was provided by Levy et al. (2014), where Berant et al. (2011)'s approach was used to create an entailment graph of OpenIE style relational phrases. A more data-driven system, PATTY (Nakashole et al., 2012), used information about relations arguments co-occurrence to not only cluster relational phrases, but also organize them into a subsumption hierarchy of relational phrases. Bovi et al. (2015) employed WSD and hierarchical information from BabelNet to find hypernymy/hyponymy pairs among relational phrases.

Classical methods for words/concepts hierarchy construction focused on noun phrases. A famous approach by Hearst (1992) used manually defined lexico-syntactic patterns (e.g. *such NP as NP*) to find hypernyms among noun phrases. This method was extended by Snow et al. (2004). In that extension, the lexico-syntactic patterns were automatically discovered using a machine learning approach and semi-supervision. Navigli and Velardi (2010) extracted hypernyms of words and concepts by (1) detecting whether a sentence is a definition, and (2) extracting hypernyms from the definition template. Ponzetto and Strube (2011) detected whether Wikipedia category links express the hypernymy relationship based on syntax-based and graph-based methods. A more recent methods proposed using word embeddings for hypernymy discovery (Anke et al., 2016) and combining huge text corpus with Hearst-like patterns (Seitner et al., 2016).

The aforementioned systems focused only on detecting whether a pair of noun phrases are in the hypernymy/hyponymy relationship. More powerful techniques consider the structure of the whole final hypernymy hierarchy using e.g., probabilistic (Snow et al., 2006), graph-based (Navigli et al., 2011; Velardi et al., 2013), or factor graph formulation (Bansal et al., 2014).

2.3 Applications

The creation of lexicons of relational phrases can be related to multiple problems. Moreover, relational phrases, semantic types of relation arguments, and relational paraphrases have shown their usefulness in many applications. Here, we discuss related problems and potential applications for lexicons of relational phrases.

2.3.1 Knowledge Base Construction

In recent years, the creation of KBs enabled advancement in computer science by organizing human knowledge in a more structured way. Usually, these constructs form information in terms of *facts*. A fact is most often structured in a *(subject, predicate, object)* tuple where *predicate* represents a relation between *subject* and *object*.

The first steps toward organizing human knowledge is visible in manually curated projects such as Cyc (Lenat, 1995), WordNet (Fellbaum, 1998), and FrameNet (Baker et al., 1998). WordNet contains information about nouns, verbs, and adjectives, as well as relationships between them like synonymy, antonymy, or hypernymy. Offering a hierarchy of verb senses, which can also be interpreted as simple relations, WordNet is also one of a few resources with a hierarchy of relations. FrameNet organizes over 1,200 *semantic frames*, which can also be interpreted as relations.

More recent endeavors for Knowledge Base Construction (KBC) concentrate on semi-automatic and automatic approaches on a large scale. These KBs focus on the storage and organization of *Named Entities* and relationships between them. However, the number of represented relations is usually limited.

YAGO (Suchanek et al., 2007; Hoffart et al., 2013; Mahdisoltani et al., 2015) extracts facts from the Wikipedia category system and Wikipedia infoboxes. Additionally, it maps Wikipedia named entities and categories to the WordNet taxonomy. DBpedia (Auer et al., 2007; Lehmann et al., 2015), similarly as YAGO, is based on Wikipedia. These KBs differ with respect to the number of represented relations. Whereas YAGO contains around 100 relations, DBpedia offers thousands concentrating more on the recall rather than on the precision of included facts. Freebase (Bollacker et al., 2008), for the creation of KB, considers information obtained via crowdsourcing. Later, Freebase was migrated to Wikidata (Vrandečić and Krötzsch, 2014), which is another example of a collaboratively edited KB. KnowItAll (Etzioni et al., 2005), unlike previously mentioned KBs, extracts facts directly from an unstructured content, namely texts available on Web pages. NELL (Mitchell et al., 2015) learns iteratively new facts from Web pages and integrates them into previously extracted facts. KnowledgeVault (Dong et al., 2014) and DeepDive (Shin et al., 2015) are other examples of KBs. They represent large-scale KBs where automatic methods extract facts with probabilistic annotation (facts have a probability of being true). Finally, a multilingual KB, BabelNet (Navigli and Ponzetto, 2010; Navigli and Ponzetto, 2012), merges multiple KBs into one structure.

All aforementioned KBs can benefit from the existence of a large taxonomy of relational phrases. First of all, knowing the relationships between the predicates in a KB can help with the extraction of new facts. With the knowledge that "*<professor>* works for *<university>*" is a hypernym of "*<professor>* is a professor at *<university>*," we can generate a new fact *(Chris, works for,* *Stanford)* from *(Chris, is a professor at, Stanford)*. Moreover, the relations in a taxonomy of relational phrases could be incorporated in a KB.

2.3.2 Textual Entailment

Another relevant application for a taxonomy of relational phrases is the textual entailment task. The task is defined as detecting whether there is a directional relationship between two textual expressions – *T text* and *H hypothesis*. *T* entails *H* if the meaning of *H* can be inferred from *T* (Dagan et al., 2013).

Lexicons of relational phrases and relational paraphrases were analyzed for their usefulness in the textual entailment scenario. Marsi et al. (2007) used DIRT's (Lin and Pantel, 2001a) relational paraphrases for Textual Entailment Recognition. The relational paraphrases were treated as rewriting rules in the paraphrase substitution algorithm on dependency parse trees. A similar idea was presented in (Dinu and Wang, 2009a; Dinu and Wang, 2009b), where DIRT paraphrases were additionally extended and refined with WordNet synonyms.

2.3.3 Question Answering

Another natural place for applying relational paraphrases is the Question Answering (QA) domain. A taxonomy of relational phrases could improve the question transformation component. For example, a QA system would not know how to answer the question "What's the birthplace of Frédéric Chopin?" if the sentence "Chopin's birthplace is Poland" did not occur in the text. However, with the relational paraphrases "*<person>* 's birthplace is *<country>*" and "*<person>* 's born in *<country>*," the system could answer that question with the sentence "Chopin was born in Poland." The task of finding a concrete, short textual answer for a question formulated in natural language is called Open-Domain Question Answering (Harabagiu et al., 2003).

The Open-Domain Question Answering problem was used multiple times for the extrinsic evaluation of relational phrases (Lin and Pantel, 2001b) and entailment rules (Schoenmackers et al., 2010). Both DIRT (Lin and Pantel, 2001b) and SHERLOCK (Schoenmackers et al., 2010) were shown to have a positive impact on the QA results. Ravichandran and Hovy (2002) went beyond relational phrases and used bootstrapping approaches to extract regular expression patterns applied to query paraphrasing on the TREC-10 dataset.

The potential of relational paraphrases was shown also in the Semantic Parsing problem. The goal of Semantic Parsing is to transform a natural language question into a logical representation that can be executed by a database. The enhanced ReVerb triples (Lin et al., 2012), containing semantically typed relational phrases, were used by Berant et al. (2013) to connect natural language expressions with the predicates from Freebase. That work was extended by a version containing a paraphrasing model (Berant and Liang, 2014). Another example of using connections between natural language relational phrases with KB predicates was shown by Yahya et al. (2013).

The aforementioned systems operate over a curated KB and consider a limited set of possible relations between entities. PARALEX (Fader et al., 2013) and its extension (Fader et al., 2014) offer end-to-end systems for QA which do not transform questions into logical representations but consider only *(subject, predicate, object)* query representation. The systems made use of both curated KBs, as well as automatically extracted resources like ReVerb, to answer natural language questions. The automatically extracted resources provided information about more relations between named entities. Moreover, the systems included PPDB (Ganitkevitch et al., 2013) as one of the ingredients of the question paraphrasing component. A similar idea of using relational paraphrases for question reformulation was presented in (Xu et al., 2016).

Another extension was presented in (Bordes et al., 2014) where the system, as one of the components, learned vector embeddings of ReVerb relational phrases.

2.3.4 Other Applications

Knowledge Base Construction, Textual Entailment, and Question Answering are the main problems where relational phrases and relational paraphrases could be applied to. However, lexicons of relational phrases could be used in many other application domains.

Relational paraphrases can be employed for Information Extraction (IE). Preemptive IE (Shinyama and Sekine, 2006) grouped the same types of events. For example, it can group articles about hurricane emergencies and discover what relations represent this kind of event (e.g., "*<hurricane>* hit *<city>*"). The clustering was performed based on the overlapping relational phrases between the articles. The fact spotting projects (Tylenda et al., 2014b; Tylenda et al., 2014a) used relational paraphrases to find textual evidence of facts included in KBs.

Another domain where relational phrases can be applied is Information Retrieval. Cafarella et al. (2006) considered using DIRT paraphrases to navigate over OpenIE-like tuples via query reformulation. Information about synonyms and hypernyms of relational phrases can be also employed in the search over Web tables (Gupta et al., 2014).

Bingel and Søgaard (2016) utilized conditional random fields and paraphrases from PPDB (Ganitkevitch et al., 2013) for text simplification. The PPDB resource, among available paraphrases, also includes relational paraphrases. Apart from text simplification, relational phrases were also used for headline generation (Alfonseca et al., 2013) and text summarization (Pighin et al., 2014). A taxonomy of relational phrases could be of use in the paraphrase generation and paraphrase detection tasks (Androutsopoulos and Malakasiotis, 2010). Relational paraphrases can serve as rewriting rules that transform a sentence into another one (Bar-Haim et al., 2007; Bar-Haim et al., 2009).

Rahman and Ng (2011) used relational phrases between Named Entities and noun phrases to perform Coreference Resolution. RESOLVER (Yates and Etzioni, 2009), for an extrinsic evaluation of the relational paraphrases mining mechanism, considered the Cross-Document Entity Resolution task.

The semantically typed relational phrases from PATTY (Nakashole et al., 2012) were employed for Named Entity Typing (Nakashole et al., 2013). The presented method was suitable for typing emerging entities, which have not been included in any KB yet. In that situation, typed relational phrases, which occur in the neighborhood of the named entity, give a very strong signal about its potential semantic type.

Relational paraphrases could be used to improve Machine Translation. Using paraphrases to generate more training data was shown to have a positive influence on Machine Translation systems (Nakov, 2008a; Nakov, 2008b; Marton et al., 2009).

Chapter 3

HARPY: Alignment of Relational Phrases

This chapter describes the first step into building a taxonomy of relational paraphrases. A taxonomy of relational paraphrases could help with the reasoning problem in natural language in tasks such as paraphrasing and entailment detection. The first resource presented in this thesis, HARPY (Grycner and Weikum, 2014), offers high-quality alignments between the relational phrases from the PATTY taxonomy (Nakashole et al., 2012) and the verb senses from WordNet (Fellbaum, 1998). Alignment information can form a useful resource for analyzing relational phrases. In addition, it can be used as a component of methods for relation hierarchy construction (e.g. RELLY (Grycner et al., 2015)).

3.1 Introduction

Motivation: This work addresses the task of discovering and organizing paraphrases of relations between entities (Lin and Pantel, 2001a; Fader et al., 2011; Nakashole et al., 2012; Moro and Navigli, 2012; Alfonseca et al., 2013). This task involves understanding that the phrases "travels to," "visits" and "on her tour through" (relating a person and a country) are synonymous and that "leader of" and "works with" (relating a person and an organization) are in a hypernymy relation: the former is subsumed by the latter. This kind of lexical knowledge can be harnessed for advanced tasks like question answering (Fader et al., 2013), search over web tables (Gupta et al., 2014), or event mining over news (Alfonseca et al., 2013).

Work along these lines has developed large repositories of relational paraphrases, most notably, the collections ReVerb (Fader et al., 2011), PATTY (Nakashole et al., 2012), and WiSeNet (Moro and Navigli, 2012). The largest of these, PATTY, contains approximately 350,000 synsets of phrases, each annotated with ontological types of their two arguments (e.g., person $\times$ country, or politician $\times$ political_party). However, the subsumption hierarchy of PATTY is very sparse. It contains only 8,000 hypernymy links between phrases, and the entire taxonomy is kind of fragmented into a many-rooted DAG (directed acyclic graph). Moreover, the synsets are rather noisy in the long tail with low confidence. WiSeNet, an alternative resource, has approximately 40,000 synsets and no hypernymy links.

WordNet (Fellbaum, 1998), on the other hand, is a very rich resource on synonymy and hypernymy. However, its coverage of binary relations (as opposed to unary predicates, mostly nouns) is restricted to (mostly) single-word verbs. WordNet has approximately 13,767 verb synsets, organized into a hierarchy with 13,239 hypernymy links. Unlike PATTY, though, WordNet does not associate verb senses with a lexical type signature for the subject and object arguments of a verb, and it is sparse in multi-word phrases. Resources like VerbNet (Kipper et al., 2008) or FrameNet (Baker et al., 1998) aim to overcome these deficiencies, but are much smaller.

Goal and Approach: In this work, our goal is to overcome the limitations of resources like PATTY and WordNet. We want to reconcile the wealth of PATTY's multi-word paraphrases with lexical typing, on one hand, and the clean hypernymy organization of WordNet verbs, on the other hand. To this end, we compute an alignment between the phrase synsets that PATTY provides with the verb senses of WordNet. This has mutual benefits:

1. we enhance many PATTY phrases with the clean hypernyms of WordNet, this way augmenting the subsumption hierarchy.

2. we extend WordNet verb senses with the lexical type signatures derived from PATTY.

Our approach uses a variety of features from both of the two aligned resources, as well as further auxiliary sources. Algorithmically, we build on an advanced notion of random walks over graphs, known as SimRank (Jeh and Widom, 2002). The system architecture is presented in Figure 3.1.

Contributions: Our method is able to construct a high-quality taxonomy of relational paraphrases, coined *HARPY*, that combines the richness of PATTY with the clean hierarchy of WordNet. The algorithm for computing the alignment is efficient and robust. One can think of the alignment as a way of sense-disambiguating PATTY phrases by mapping them to WordNet. HARPY links 20,812 of the PATTY phrases to WordNet. Conversely, 4,789 out of 13,767 WordNet verb senses are enriched with information from PATTY. We evaluate the quality of HARPY by extensive sampling with human assessment. We also demonstrate its benefit by the extrinsic use-case of annotating WordNet verb senses with lexical type signatures. The experimental data and HARPY resource are publicly available at `www.mpi-inf.mpg.de/yago-naga/patty/`.

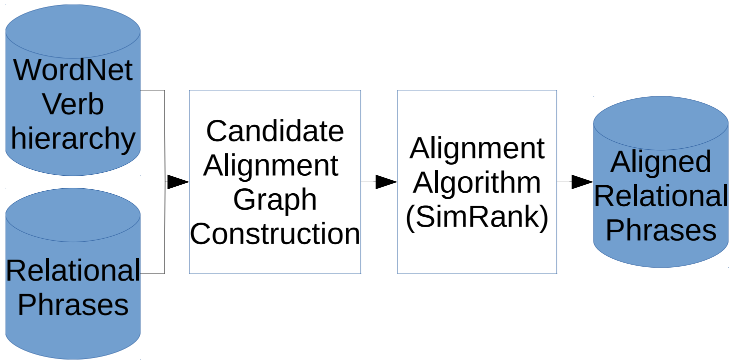


Figure 3.1: HARPY system architecture

3.2 Constructing a Candidate Alignment Graph

The general idea of the main algorithm is to align phrase synsets from the PATTY taxonomy with verb synsets in WordNet. To this end, we first construct a directed *Candidate Alignment Graph (CAG)*. Section 3.3 will then discuss the actual alignment algorithm.

Vertices of the CAG represent:

- synsets of *relational phrases* in PATTY, or phrases for short,
- *verb senses* from WordNet, verbs for short,
- *features* of either phrases or verbs.

Edges of the CAG correspond to relations between phrases, verbs, and features. We consider three types of relations here: similarity, hypernymy, and vertex-features. Edges are weighted (see Section 3.2.5).

3.2.1 Vertex Types

There are 6 kinds of vertices in the CAG. Since we aim to connect PATTY *phrases* with WordNet *verbs*, these two are the main kinds of vertices. Additionally, the graph contains feature vertices representing *noun senses* from WordNet (nouns for short), *surface verbs* as occurring in sample texts, *sentence frames* from WordNet, and specifically derived *phrase-verb vertices* connecting phrases and verbs. The latter are constructed by combining each phrase with its top-10 most similar verb senses. To this end, we retrieve all verb synsets from WordNet and rank the verb synsets by the cosine similarity between the support sentences that PATTY provides for its phrases (i.e., sentences from Wikipedia that contain instances of a phrase) and the usage examples in WordNet glosses. The resulting vertices are labeled by the combination of phrase id and verb-sense id. Having these combinations as vertices, rather than simply connecting phrases and verbs via edges, leads to a CAG structure that is better suited for our random walk algorithms (see Section 3.3). Table 3.1 gives examples of the 6 vertex types.

3.2.2 Edge Types

Edges in the graph represent 3 different types of relationships between vertices:

- For all relational phrases, all verb senses from WordNet and also all noun senses (as feature vertices), we capture their *hypernymy relations* as edges.
- We connect phrase-verb vertices with their constituents, phrase vertices and verb vertices, by *similarity edges*, with weights derived from the similarity computation.
- The remaining edges connect phrases or verbs with their respective feature vertices. There are 6 kinds of such *vertex-feature edges*, explained next.

3.2.3 Verb Features

The following features are associated with verb senses. A *lemma edge* connects a verb sense with one or more surface-verb vertices, as given in WordNet glosses. A *domain edge* edge connects a verb sense with noun senses that describe the usage domain of the verb (e.g. literature, politics). This information is retrieved from WordNet and the WordNet Domains project (Bentivogli et al., 2004). While the latter does not provide sense-disambiguated information, we need to add a mechanism which maps domain information to its WordNet noun sense counterpart. Therefore, we map domain surface nouns to their most frequent senses.

In addition, we harness the WordNet links of type *derivationally related form* to construct further edges between verb senses and noun-sense features in our CAG. The last types of edges for verb-sense features are *sentence frame edges*, between verb vertices and feature vertices of type sentence frame. WordNet for each verb sense provides information about its sentence frames. There are defined 35 possible sentence frames.

3.2.4 Phrase Features

Relational phrases are associated with the following features. A *verb-in-phrase edge* connects a phrase with a surface verb whenever the phrase contains the verb after lemmatization. Analogously to the domain edges for verb senses, we introduce *Wikipedia-category edges* between relational phrases and noun senses. PATTY provides us with Wikipedia articles where instances of a phrase occur. We consider all Wikipedia categories of such an article as a source for related noun senses. We use ontological types of the articles and the categories

Relational Phrase	*Verb Sense*	*Noun Sense*	*Surface Verb*	*Sentence Frame*	*Phrase-Verb Pair*
[person] succeeded [person]	succeed2#verb	king1#noun	succeed	Somebody `--s` somebody	(phrase_1, verb_sense_2)
[musician] played jazz with [musician]	play3#verb	music1#noun	play	Somebody `--s` something	(phrase_2, verb_sense_3)

Table 3.1: Vertex types with examples

Hypernymy	*Similarity*	*Lemma*	*Domain*	*Derivationally Related Form*	*Sentence Frame*
replace2#verb	(phrase_1, verb_sense_2)	"succeed", "come after"	politics1#noun	successor1#noun	Somebody `--s` somebody

Table 3.2: Vertices connected by different edges with the vertex "succeed2#verb" of type verb.

Hypernymy	*Similarity*	*Verbs in phrase*	*Wikipedia Category*	*Sentence Frame*
[person] replaced [person]	(phrase_1, verb_sense_2)	"succeed"	politician1#noun	Somebody `--s` somebody

Table 3.3: Vertices connected by different edges with the vertex "[person] succeeded [person]" of type phrase.

and their mappings to WordNet provided by the YAGO project (Suchanek et al., 2007). Finally, we also introduce *sentence-frame edges* between relational phrases and sentence-frame feature vertices. To avoid polluting the CAG with overly noisy connections, we apply specific tests. First, we check if the lexical argument types of a phrase and a frame are compatible (e.g., musician is compatible with person, but not with location). Second, we compare characteristic prepositions in the phrase and the frame. We create and edge only if these additional tests are affirmative.

Examples of vertices connected by the different edge types with verb vertices and phrase vertices are shown in Table 3.2 and 3.3, respectively.

3.2.5 Edge Weights

All edges in the graph are weighted. The weights are derived from frequency counts of features and/or similarity scores, or are simply set to 1 for binary cases (e.g., hypernymy edges). Lemma edges between verb senses and surface verbs vertices are weighted in proportion to the frequency count of a verb sense, as given by WordNet. Wikipedia-category edges have weights based on the number of occurrences of a relational phrase in Wikipedia articles and the frequencies of categories. Similarity edges have weights set according to the cosine similarity between examples of a verb sense and examples of a relational phrase.

Finally, we normalize all weights in the graph by requiring that the sum of weights of the incoming edges is equal to 1 for every vertex. For the verb and phrase vertices, we perform an additional normalization so that each kind of edge has the same impact in terms of the total edge weight per edge kind as shown in Figure 3.2.

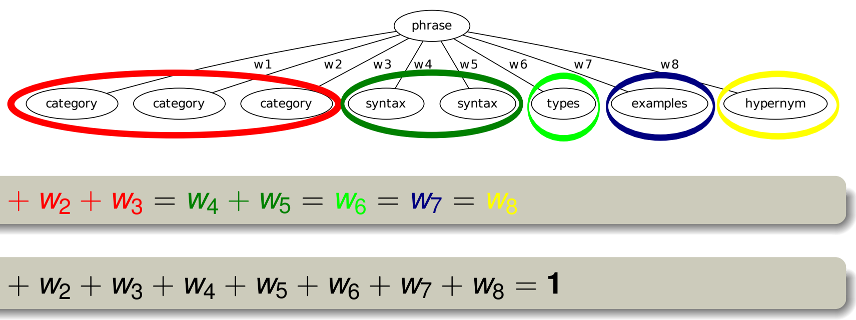


Figure 3.2: Example of the edge weights normalization procedure

The above procedure leads to a CAG with 238,437 vertices and 4,776,116 edges. Figure 3.3 shows an excerpt for illustration.

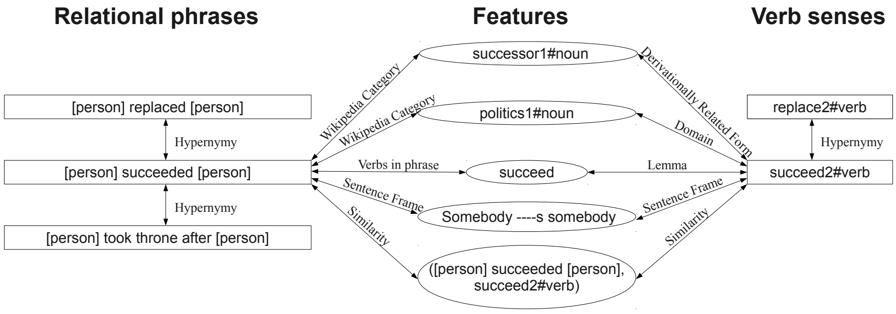


Figure 3.3: Excerpt from Candidate Alignment Graph

3.3 Alignment Algorithm

Our algorithm runs on the directed *CAG*. Intuitively, it aims to find "strong paths" between relational-phrase vertices and verb-sense vertices. We use random-walk methods to this end. For each relational phrase, we compute scores and a ranked list of verb senses to which the phrase likely corresponds. The top-ranked verb would ideally be the desired alignment.

3.3.1 SimRank

We employ the SimRank algorithm (Jeh and Widom, 2002), an advanced form of random walks. SimRank computes similarity scores between a pair of vertices in a weighted graph, based on the neighborhoods of the two vertices. The definition, formally given in Equation 3.1, is recursive: two vertices are similar if their neighborhoods are similar. In the standard SimRank equation, $I_i(a)$ represents the i^{th} (incoming) neighbor of vertex a, and C is a constant dampening factor.

$$s(a,b) = \frac{C}{|I(a)|\,|I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s\left(I_i(a), I_j(b)\right) \tag{3.1}$$

SimRank helps capturing long-distance dependencies between vertices in a graph. This would not be achieved by simpler similarity measures of context vectors. Note that SimRank is quite different from (Personalized) PageRank methods; SimRank can be seen as a random walk over pairs of nodes, not over individual nodes. During the CAG construction, we tried to keep the path lengths between phrase vertices and verb vertices uniform for all kinds of feature vertices, to avoid biasing the influence of specific features. Since the SimRank similarity is based on two random walks meeting, the method works best when all paths between source-target node pairs have even length. With this property SimRank produces better results; we introduced explicit phrase-verb vertices for this reason.

3.3.2 SimRank with Fingerprints

Unfortunately, SimRank has very high computational complexity: the runtime of a straightforward implementation is $\mathrm{O}(Kn^4)$, where n is the number of vertices in the graph and K is the number of iterations in an iterative fixpoint computation (in the style of the Jacobi method). However, there are much faster approximations of SimRank. We use a variant known as *SimRank with fingerprints* (Fogaras and Rácz, 2005) To approximate the SimRank score for two vertices, this method computes the *expected first meeting time* for two random walks originating from the two vertices (with randomized restarts). To this end, the method precomputes a fingerprint for each vertex a: a data structure holding the visiting probabilities of vertices for standard random walks originating in a. A fast implementation actually runs random walks a specified number of times, to estimate the visiting probabilities. For two vertices a and b, the expected number of hops until their random walks meet in a common vertex is then efficiently computed from the fingerprints of a and b. Moreover, this method allows computing the SimRank score for a pair of vertices on demand, only for vertex pairs of interest, rather than having to compute all $O(n^2)$ scores.

The original SimRank method works with unweighted graphs. In our setting, we modify transition probabilities according to edge weights. Our extended SimRank variant is equivalent to Equation 3.2, where $W(a, b)$ denotes the weight of the edge between a and b. This equation is similar to the weighted variant of (Antonellis et al., 2008).

$$s_w(a,b) = C * \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} W(a, I_i(a)) * W(b, I_j(b)) * s_w(I_i(a), I_j(b)) \qquad (3.2)$$

Unlike the original SimRank method, we also incorporate random jumps in the underlying random-walk model. Each vertex has a different random jump probability, explained next.

3.3.3 Random Jumps

The original SimRank definition favors vertices with smaller neighborhoods. To avoid this bias, we introduce a form of smoothing on the graph. Whenever a phrase vertex or verb vertex lacks some of the feature types that other vertices may have, we introduce an option for random jumps from the given vertex to any other vertex in the graph. For each missing kind of feature (e.g., domain feature or sentence-frame feature), we assign a probability mass of ϵ, a small constant, for a random jump. So if several features are missing, there is an accumulated probability for a jump. The target of a random jump is always chosen with uniform distribution. A final normalization of edge weights (with linear adjustment) ensures that the possible transitions from a vertex form a proper probability distribution. The method works also without smoothing (i.e., setting the constant to 0), but the results tend to be worse. The results are not very sensitive to the exact choice of the random-jump parameter.

3.3.4 Filtering and Candidate Pruning

The target of our alignment is the WordNet verb hierarchy, but not all relational phrases can be mapped into this target space. Therefore, we restrict ourselves to a subset of relational phrases that contain exactly one verb. This eliminates noun phrases (e.g. "father of") and phrases that contain multiple verbs (e.g. "succeed and died," "succeeded in persuading"). Noun phrases should be aligned to the WordNet noun hierarchy and it should be treated as a different task (using e.g. state-of-the-art work (Ponzetto and Navigli, 2010)). Multi-verb phrases often pose semantic difficulties. Note that the verbs in these phrases are always transitive verbs, as PATTY is derived from subject-phrase-object structures in large corpora. We also used the cardinalities of the support sentences in PATTY for pruning the noisy tail of phrases, by dropping all phrases that have only a single instance.

To avoid computing SimRank scores for every pair of vertices, we prune the search space as follows. We consider only pairs of relational phrases and verb senses which contain the same surface verb (with lemmatization).

3.3.5 Deriving Hypernymy Links

Once we have alignments between phrases and verbs, we derive hypernymy relations *among phrases* as follows. Whenever phrases p_1 and p_2 are aligned with verb senses v_1 and v_2, respectively, and v_1 is a direct or transitive hypernym of v_2, we infer that p_1 is a hypernym of p_2. We consider transitive hypernyms because not every WordNet verb sense has a phrase aligned with it; without transitivity we would obtain a very sparse hierarchy. By the acyclicity of the WordNet hypernymy structure, the process yields a proper DAG. However, the output contains redundant links (direct ones and transitive ones connecting the same pair of phrases); these are subsequently eliminated by a transitive reduction algorithm (Aho et al., 1972). The example of deriving hypernymy links is presented in Figure 3.4.

3.4 Evaluation

We evaluated the quality of the HARPY alignments by manual assessment of a large sample set, and compared it against several alternative methods.

Baselines: We compared our SimRank-based method against the following baselines, each given the same feature set:

- *Cosine Similarity:* for each relational phrase and verb sense, we create a contextual vector (in the spirit of distributional semantics) consisting of

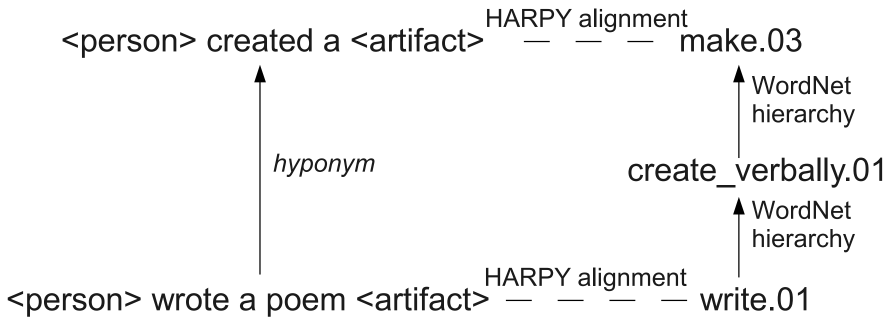


Figure 3.4: Deriving Hypernymy Links

the features described in Section 3.2, with tf-idf-based weights (Manning et al., 2008a). The alignment ranking is computed by the cosine similarity of tf-idf-weighted contextual vectors.

- *Modified Adsorption (MAD):* a label propagation algorithm (Talukdar and Crammer, 2009) run on the Candidate Alignment Graph. In our setting, each relational phrase is a label. Initially, only the respective phrase vertices have this label. The algorithm propagates labels to other vertices, based on the graph's edge weights. The top-k results for the alignment of a phrase are the verb senses with the highest probability for the phrase label. We use the Junto Label Propagation Toolkit [1].

- *Personalized PageRank (PPR):* a method for random walks with random jumps back to the start vertex (Haveliwala, 2002). For each phrase, a separate PPR is performed. The ranking of verb senses is produced by the visiting probabilities according to the PPR scores.

- *Most Frequent Sense (MSF):* For each phrase, we consider only verb senses that contain the same surface verb (with lemmatization), and rank them by the WordNet frequency information.

Assessment: We retrieved a random subset of 261 relational phrases considered for alignment, and showed the results of the different alignment methods to two human judges. For each relational phrase, we displayed its textual form, list of usage examples, and the top-5 ranked list of verb senses computed by each method under comparison. Each verb sense was enriched with information about its lemmas, its gloss, and examples. The evaluators were asked to identify the verb sense that is semantically equivalent to the given relational phrase (including the option of saying "none"). The example of the evaluated phrase is presented in Figure 3.5.

Quality Measures: As all methods compute a ranked list of verb senses for a given phrase where exactly one list item is correct, we use quality measures

[1] http://code.google.com/p/junto/

Subject	Phrase	Object
wordnet_organization_108008335	wrote [[det]] song about;	wordnet_person_100007846

- American metalcore band Unearth wrote a song about Giles Corey named Giles for their album III : In the Eyes of Fire .
- TributesGlasgow band Glasvegas wrote the song `` Flowers And Football Tops '' about Donald 's murder from the point of view of his father .
- Relient K wrote a song about Marty McFly called `` Hello McFly '' on their eponymous first album .
- Australian punk band Frenzal Rhomb wrote a song about Karl , on their 2011 album , Smoko At The Pet Food Factory , called ` Alvarez ' .
- Quando Benigni prese in braccio Berlinguer 1983 , Youtube Italian folk music band Modena City Ramblers wrote a song about Berlinguer 's funeral , `` I Funerali di Berlinguer '' , which was published on their first full length album , Riportando Tutto a Casa .

	Lemmas	Definition	Examples
☐	seek	try to get or reach	seek a position. seek an education. seek happiness
☑	compose, write	write music	Beethoven composed nine symphonies
☐	test	determine the presence or properties of (a substance)	
☐	go	be sounded, played, or expressed	How does this song go again?
☐	about face	turn, usually 180 degrees	
☑	write_on, write_of, write_about	write about a particular topic	Snow wrote about China
☐	write	communicate by letter	He wrote that he would be coming soon

Figure 3.5: Excerpt from the evaluation framework

geared for such rankings: Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain ($NDCG$). In addition, we report on the precision for top-k results, for small k (1, 3, or 5). Here, a top-k result is considered good if the correct verb senses appear among the top-k alignments, for a given phrase.

Results: The results are shown in Table 3.4. Our method outperforms all baselines. Among the competitors, MFS shows the best performance. This is not so surprising; MFS is rarely outperformed in word sense disambiguation (McCarthy et al., 2004; Navigli and Lapata, 2010). Our gains over MFS are remarkable. In total, HARPY aligned 20,812 phrases to 4,789 verb senses, and also obtained 616,792 hypernymy links between phrases.

The evaluation process led to high inter-judge agreement, with Cohen's Kappa around 0.678. The number of samples, 261, was large enough for statistical significance: we performed a paired t-test for MRR, $NDCG$ and $Precision@1$ of the SimRank results against each of the baselines, and obtained p-values below 0.05.

	SimRank	MFS	PPR	MAD	Cosine
MRR	**0.698**	0.664	0.553	0.463	0.252
NDCG	**0.733**	0.705	0.584	0.510	0.279
Precision@1	**0.571**	0.517	0.410	0.318	0.161
Precision@3	**0.793**	0.778	0.644	0.594	0.307
Precision@5	**0.874**	0.866	0.736	0.670	0.391

Table 3.4: Evaluation

Tables 3.5 and 3.6 shows example results that HARPY computed. Table 3.5 has correct outputs. We see that HARPY manages to distinguish between the sport, musical, and theatrical senses of the verb "play." As shown in Table 3.6, HARPY also produces some spurious results, with various factors contributing to these errors. For example, the phrase "covered on album" was aligned with the first sense of "cover" since there is no musical sense for "cover" in WordNet. Other errors arise from mistakes in the original PATTY repository of relational phrases. For example, the travel sense of the verb "head" was aligned with the phrase "head of" because "head of" and "head to" were in the same PATTY synset. Yet another cause of problems is the extremely fine granularity of WordNet: even for humans it is often hard to distinguish between love as a state of liking and love as being enamored.

3.5 Extrinsic Study: Lexical Types for WordNet Verbs

As an extrinsic use-case for the HARPY resource, we studied the task of inferring lexical types for the subject and object arguments of a WordNet verb sense. For a given verb sense, we propagate the type signature of the relational phrase with the highest alignment score.

For comparison, this procedure is performed with the HARPY alignments as well as the alignments by the baseline methods. We showed a uniformly sampled set of 261 results to human judges, who assessed as valid or invalid. Additionally, we had a set of the 100 most-confident results (those derived from the highest alignment scores) assessed in the same manner.

For the uniform samples, the type signature derived from HARPY had a precision of 0.46, whereas the best of the baselines (PPR and Cosine) achieved 0.39. For the top-100 samples, HARPY achieved a precision of 0.81. Table 3.7 shows some example results, demonstrating the added value beyond WordNet.

3.6 Related Work

With the proliferation of knowledge bases, like Freebase (Google Knowledge Graph), DBpedia, YAGO, or ConceptNet, there is a wealth of resources about

Relational phrase	Verb Sense	WordNet definition
[musician] played with [musician]	play3	play on an instrument
[actor] played role in [event]	act3	play a role or part
[person] played hockey for [club]	play1	participate in games or sport
[person] was shooting [person]	shoot2	kill by firing a missile
[movie] be shot in [city]	film1	make a film of something
[composition] written by [composer]	compose2	write music
[writer] writing at [organization]	write1	produce a literary work
[musician] performed in [city]	perform3	give a performance
[architect] also designed in [city]	design3	create the design
[sovereign] succeeded his father [person]	succeed2	be the successor of
[person] succeeds in [artifact]	succeed1	attain success
[person] wrote nine books [person]	publish3	issued for publication
[artist] illustrated works [write]	illustrate3	supply with illustrations
[aviator] flew for [organization]	fly3	operate an airplane

Table 3.5: Correct examples

Relational phrase	Verb Sense	WordNet definition
[person] covered on album [artifact]	cover1	provide with a covering or cause to be covered
[person] head of [artifact]	head1	to go or travel towards
[person] becomes sure that [person]	become1	enter or assume a certain state or condition
[person] is loved by [person]	love1	have a great affection or liking for
[wrestler] wrestled in [organization]	wrestle1	combat to overcome an opposing tendency or force

Table 3.6: Wrong alignment examples

entities and semantic classes (i.e., unary predicates and their instances). In contrast, the systematic compilation of paraphrases for relations (i.e., binary predicates) has received much less attention. Some of the knowledge-base projects, especially those that center on Open Information Extraction (OpenIE), make intensive use of surface patterns (e.g., verbal phrases) that indicate relations (e.g., (Carlson et al., 2010; Fader et al., 2011; Mausam et al., 2012; Speer and

Domain	Range	Verb Sense	WordNet definition
country	country	export1	sell or transfer abroad
person	country	head2	be in charge of
organization	organization	own1	have ownership or possession of
person	person	predate1	be earlier in time; go back further
saint	organization	reverence1	regard with feelings of respect and reverence
person	artifact	rush5	run with the ball, in football
organization	person	sustain4	supply with necessities and support
musician	musician	play3	play on an instrument
football_player	athlete	pass20	throw (a ball) to another player
singer	composer	inspire2	supply the inspiration for
ruler	country	suppress1	to put down by force or authority
architect	city	design2	plan something for a specific role or purpose or effect
priest	saint	canonize2	treat as a sacred person
country	country	ally_with1	unite formally; of interest groups or countries
company	organization	deal13	sell
artifact	computer_game	port8	modify (software) for use on a different machine or platform
artifact	actor	star1	feature as the star
director	event	direct4	be in charge of
person	song	sing1	deliver by singing
coach	organization	coach1	act as a trainer or coach to

Table 3.7: Type inference examples by HARPY

Havasi, 2012; Wu et al., 2012)); however, they do not organize these patterns into a WordNet-style taxonomy.

Prior work towards such taxonomies go back to the projects DIRT (Lin and Pantel, 2001a), VerbOcean (Chklovski and Pantel, 2004), and VerbNet (Kipper et al., 2008). However, the resulting resources were mostly restricted to single verbs. ReVerb (Fader et al., 2011) extended these approaches by automatically mining entire phrases from Web contents, but still with the focus on verbal structures. PATTY (Nakashole et al., 2012) used sequence mining algorithms for gathering a general class of relational phrases, organizing them into synsets, and inferring lexical type signatures. WiseNet (Moro and Navigli, 2012) harnessed phrases from Wikipedia articles and clustered them into synsets of relational phrases. All of these works are fairly limited in their coverage of subsumptions (hypernymy) between relational phrases.

There is ample work on computing alignments among different kinds of lexical thesauri, dictionaries, taxonomies, ontologies, and other forms of linguistic or semantic resources. Prominent cases along these lines include the alignments between FrameNet and WordNet (Ferrández et al., 2010), VerbNet and PropBank (Palmer, 2009), Wikionary and WordNet (Meyer and Gurevych, 2012), and across multilingual WordNets and/or Wikipedia editions (e.g., (de Melo and Weikum, 2009; Navigli and Ponzetto, 2012)). For aligning ontologies based on OWL and RDF logics, there is a series of annual benchmark competitions (Grau et al., 2013). Most approaches are based on relatedness measures and context similarities between words or concepts and their neighborhoods in the respective resources (e.g., (Banerjee and Pedersen, 2003; Budanitsky and Hirst, 2006; Gabrilovich and Markovitch, 2007)). Algorithmically, this translates into a nearest-neighbor (most-similar) assignment between entries of different resources. More sophisticated methods use similarities merely to assign weights to relatedness edges in a graph, and then employ random walks on such a graph (e.g., (Pilehvar et al., 2013)). The prevalent method of this kind uses Personalized Page Rank (Haveliwala, 2002)), computing stationary probabilities for reaching nodes in one resource when starting random walks on a given node of the other resources (with randomized restarts).

Computing alignments between resources can sometimes be viewed as a task of disambiguation words or concepts in one resource by mapping them to the other resource (e.g., mapping Wiktionary entries onto WordNet senses). Thus, the huge body of work on word sense disambiguation (WSD) is relevant, too. Methodologically, this research also relies, to a large extent, on relatedness/similarity measures and random walks on appropriately constructed graphs. See (Navigli, 2009) for an extensive survey.

There is remotely related work on several other tasks in computational linguistics and text mining. These include semantic relatedness between concepts or words (e.g., (Gabrilovich and Markovitch, 2007; Pilehvar et al., 2013)), type inference for the arguments of a phrase (e.g., (Kozareva and Hovy, 2010; Nakashole et al., 2013)), and entailment among verbs (e.g., (Hashimoto et al., 2009)). The SemEval-2010 task on classification of semantic relations (Hendrickx et al., 2010) addressed the problem of predicting the relation for a given sentence and pair of nominals, but was limited to a small prespecified set of relations.

3.7 Conclusion

HARPY is a new resource that aligns lexically typed multi-word phrases for binary relations with WordNet verb senses. By judiciously devising appropriate features and adapting and extending an advanced random-walk method, SimRank, we achieved high-quality alignments, as shown in our evaluation. This creates added value for both the resource of relational phrases, PATTY, and WordNet. Phrases are now organized into a hypernymy hierarchy with high coverage, an important aspect on which the PATTY work fell short. WordNet verb senses, on the other hand, are extended by a rich set of paraphrases and also by lexical type signatures inherited from the phrases. We believe that this new resource is a useful asset for computational linguistics. The future work could concentrate on aligning additional resources like WiseNet (Moro and Navigli, 2012), FrameNet (Baker et al., 1998) or VerbNet (Kipper et al., 2008). The HARPY resource is publicly available at `www.mpi-inf.mpg.de/yago-naga/patty/`.

Chapter 4

RELLY: Hierarchy of Relational Phrases

The work in Chapter 3 offers high-quality alignments between relational phrases and WordNet verb senses. Using alignment information, we could impose the WordNet verb sense hierarchy on the relational phrases. This simple approach leads to a hierarchy which has high recall (unlike the hierarchy in PATTY) but unfortunately suffers from low precision. This chapter describes RELLY (Grycner et al., 2015), which improves the HARPY output hierarchy by combining multiple signals within the Probabilistic Soft Logic (PSL) framework.

4.1 Introduction

One of the many challenges in natural language understanding is interpreting the multi-word phrases that denote relationships between entities. Semantically organizing the complex relationships between diverse phrases is crucial to applications including question answering, open information extraction, paraphrasing, and entailment detection (Yahya et al., 2012; Fader et al., 2011; Madnani et al., 2012; Dagan et al., 2005). For example, a corpus containing the phrase "George Burns was married to Gracie Allen" allows us to answer the query "Who was the spouse of George Burns?". However, "Jay Z is in a relationship with Beyoncé" provides insufficient information to determine whether the couple is married. To capture the knowledge found in a text, relational phrases need to be systematically organized with lexical links like synonymy ("married to" and "spouse of") and hypernymy ("in a relationship" generalizing "married to").

Many projects address the challenge of understanding relational phrases, but existing linguistic resources are often limited to synonymy, suffer from low precision, or have low coverage. Systems such as DIRT (Lin and Pantel, 2001a), RESOLVER (Yates and Etzioni, 2009), and WiseNet (Moro and Navigli, 2012) have used sophisticated clustering techniques to determine synonymous phrases, but do not provide subsumption information. The PATTY

(Nakashole et al., 2012) project goes beyond clustering and introduces a subsumption hierarchy, but suffers from sparsity and contains few hypernymy links. The HARPY (Grycner and Weikum, 2014) project extended PATTY, generating 600,000 hypernymy links, but with low precision. Berant et al. (2011) introduced entailment graphs that provided a high-quality subsumption hierarchy. This method required partitioning the graph and the largest component consisted of 120 relations. A number of manually-curated relational taxonomies such as WordNet (Fellbaum, 1998), VerbNet (Kipper et al., 2008), and FrameNet (Baker et al., 1998) also offer high-precision hierarchies with limited coverage.

In this work, we introduce RELLY, a method for producing a hypernymy graph that has both high coverage and precision. We build on previous work, integrating the high-precision knowledge in resources such as YAGO (Suchanek et al., 2007) and WordNet with noisy statistical information from OpenIE projects PATTY and HARPY. RELLY maintains a consistent graph by including collective global constraints such as transitivity, asymmetry, and acyclicity. Scalability is often a concern when employing collective reasoning over large corpora, but our system can produce graphs with over 100,000 edges on conventional hardware. As a result, we produce a large, complete, and high-precision hypernym graph that includes alignments and type information.

RELLY leverages Probabilistic Soft Logic (PSL) (Bach et al., 2015), a popular probabilistic modeling framework, to collectively infer hypernymy links at scale. PSL uses continuously-valued variables and evidence, allowing easy integration of uncertain statistical information while encoding dependencies between variables using a first-order logic syntax. We define a PSL model with rules that combine statistical features, semantic information, and structural constraints. Statistical features, such as argument overlap and alignments to WordNet verbs senses, allow RELLY to learn from large text collections. Semantic information, such as type information for relation arguments, improves precision of the resulting inferences. Structural constraints, such as transitivity and acyclicity, enforce a complete and consistent set of edges. Using this PSL model, we learn rule weights with a small amount of training data and then perform joint inference over all hypernymy links in the graph.

We highlight three major contributions of our work. First, we introduce RELLY, a scalable method for integrating statistical and semantic signals to produce a hypernymy graph. RELLY is extensible and can easily incorporate additional information sources and features. Second, we generate a complete and precise hypernymy graph over 20,000 relational phrases and 35,000 hypernymy links. We have publicly released this hypernymy graph as a resource for the NLP community. Third, we present a thorough empirical evaluation to measure the precision of the hypernymy graph as well as demonstrate its usefulness in a real-world document ranking task. Our results show a high precision (0.78) and superior performance in document ranking compared to state-of-the-art models such as word2vec (Mikolov et al., 2013).

4.2 Background

Before describing the details of RELLY, we begin with necessary background information on the task of semantically organizing relational phrases, as well as the probabilistic soft logic modeling language which we use to develop our hypernymy graph construction method.

4.2.1 Relational Phrases

Relational phrases are textual representations of relations which occur between named entities (e.g., "Terry Pratchett") or noun phrases (e.g., "the great writer"). Nakashole et al. (2012) identify relational phrases with the *semantic type signature* of the relation, i.e. the fine-grained lexical types of left- and right-hand side arguments. For example, "Terry Pratchett published his new novel The Colour of Magic" is an instance of the relational phrase "<*person*> published his * ADJ novel <*book*>." In this case, the left-hand argument (the domain of the relation) has the type <*person*> and the right-hand argument (the range of the relation) has the type <*book*>.

Several projects from the Open Information Extraction (OpenIE) community have addressed the task of finding synonyms of relational phrases using clustering algorithms. The biggest collection of relational phrases and their synonyms is currently the PATTY project (Nakashole et al., 2012), with around 350,000 semantically typed relational phrases. Prominent alternatives are WiseNet (Moro and Navigli, 2012), which offers 40,000 synsets of relational phrases, PPDB (Ganitkevitch et al., 2013), which contains over 220 million paraphrase pairs, as well as DIRT and VerbOcean (Lin and Pantel, 2001a; Chklovski and Pantel, 2004) which inspired the approach and results pursued here.

Relational phrases can be further organized into a hierarchical structure according to their hypernymy (subsumption) relationships. For example, "<*person*> moves to <*country*>" is a hypernym of the relational phrase "<*musician*> emigrates to <*country*>." Of the aforementioned collections, only PATTY attempts to automatically create a subsumption hierarchy for the extracted relational phrases. The authors of the HARPY system argue that the sparseness of PATTY's graph comes from the lack of general phrases in the source corpus. As a solution, they propose using the WordNet verb hierarchy (which contains general verb senses) to construct a similar hierarchy with PATTY's relational phrases. The graph obtained by HARPY consists of around 600,000 hypernymy links for around 20,000 relational phrases. However, the final graph was not evaluated for precision; rather, the evaluation was instead concentrated on the alignment between verb senses and relations.

In this work, we will make use of several concepts that are closely related to hypernymy, which we define below. Note that although the following definitions concern verbs, we also apply them to relational phrases:

- *hypernym*: the verb Y is a hypernym of the verb X if Y is more general than X. *To perceive* is a hypernym of *to listen* (Bai et al., 2010).
- *troponym*: the verb Y is a troponym of the verb X if doing Y is doing X, in some manner. *To lisp* is a troponym of *to talk* (Bai et al., 2010). Troponym is a verb counterpart for *hyponym*, which applies to nouns. In this work we use these two terms interchangeably.
- *entailment*: the verb Y is entailed by X if, by doing X, you must be doing Y. *To sleep* is entailed by *to snore* (Bai et al., 2010).

4.2.2 Probabilistic Soft Logic

Our approach is based on Probabilistic Soft Logic (PSL), a popular statistical relational learning system which we briefly describe here. PSL is a templating language for a class of graphical models known as hinge-loss Markov random fields. PSL models are specified using rules in first-order logic syntax, expressing dependencies between interrelated variables. For example, the PSL rule

$$\begin{aligned} w : \textsc{Hypernym}(P_1, P_2) \wedge \textsc{Hypernym}(P_2, P_3) \\ \Rightarrow \textsc{Hypernym}(P_1, P_3) \end{aligned}$$

expresses the transitivity of hypernyms: if phrase P_1 is a hypernym of phrase P_2 and P_2 is a hypernym of P_3, then P_1 is a hypernym of P_3. Rules are weighted (w) to indicate their importance in the model, and weight learning in PSL allows these weights to be learned from training data.

Each rule is grounded by substituting the variables in the rule with constants, e.g.“married to” and “relative of” for P_1 and P_2. However, unlike previous approaches such as Markov Logic Networks, the atoms in each logical rule take values in the [0,1] continuous domain. In addition to providing a natural way of incorporating uncertainty and similarity into models, continuous-valued variables allow the inference objective to be formulated as convex optimization making MAP inference extremely efficient, with empirical performance that scales linearly with the number of ground rules.

4.3 Hypernymy Graph Construction

In this section we detail RELLY, our system for constructing a hypernymy graph. RELLY incorporates semantic and statistical information from sources such as YAGO, WordNet, PATTY, and HARPY, and uses PSL to combine and reason over these sources. For each source, we introduce a PSL predicate (Table 4.1). The predicates are divided into three categories: *statistical* (continuous-valued features arising from statistical methods), *semantic* (binary predicates acquired from knowledge bases) and *output* (the target variables). We relate these predicates with a series of rules which combine alignment links, argument similarity, and hierarchical information. The collection of rules defines the PSL model, which we describe in Section 4.3.1 and Table 4.2.

In the resulting hypernymy graph, an edge from a relational phrase $R1$ to a relational phrase $R2$ denotes that $R1$ is more specific than $R2$, i.e. $R2$ is a hypernym of $R1$. For example, there is an edge from $R1$ = "<*musician*> emigrates to <*country*>" to $R2$ = "<*person*> moves to <*country*>." In the PSL model the strength of this edge is represented by the confidence score of the predicate $hyponym(R1, R2)$.

4.3.1 PSL Rules

The PSL rules that define the model are shown in Table 4.2. Each of the rules is additionally supplied with a weight which describes its importance in the model. The weights are learned from a small hand-crafted hierarchy of relational phrases. The full PSL model combines multiple statistical and semantic signals into the hypernymy graph.

Our model includes rules to encode signals that provide evidence for hypernymy, as well as rules to encode consistency in the graph. One statistical signal for phrase subsumption is *argument overlap*. If the arguments to a relational phrase $R1$ are also found as arguments to another relational phrase $R2$, $R1$ and $R2$ may be synonymous or $R2$ may be a hypernym of $R1$. We use two measures of argument overlap, $weedsInclusion$ and $pattySubsumption$, in rules 1 and 2, respectively, to capture the relationship between argument overlap and subsumption. Another signal, used in rule 3, is the *alignment* between relational phrases and WordNet verb senses. If relational phrases $R1$ and $R2$ are aligned to WordNet verb senses $Vb1$ and $Vb2$ which are in a hyponymy relationship, then this is the evidence that $R1$ is more specific than $R2$. An example of using HARPY alignment links and WordNet hierarchy is shown in Figure 4.1.

We encode local consistency requirements using Rules 4–6. Rule 4 (*types compatibility*) is a constraint to restrict hypernymy links to be between relations whose types are compatible, i.e they are identical or the types of the more specific relation are subtypes of the types of the more general relation. Rules 5 and 6 create a *transitive closure* of both WordNet and YAGO hierarchies. As a result of these rules, we can use indirect hyponyms (in rule 3) or indirect subtypes (in rule 4).

Finally, rules 7, 8 and 9 shape the structure of the output graph with collective global constraints. Rule 7 (*asymmetry*) removes bidirectional links, rule 8 (*transitivity*) creates a transitive closure of the graph and rule 9 (*acyclicity*) prevents the creation of small cycles in the graph.

PSL predicate	**Type**	**Description**
$weedsInclusion(R1, R2)$	statistical	degree of inclusion of sets of argument pairs of relations defined as $\frac{\lvert ArgsR1 \rvert}{\lvert ArgsR1 \cap ArgsR2 \rvert}$ (Weeds and Weir, 2003)
$pattySubsumption(R1, R2)$	statistical	PATTY subsumption (Nakashole et al., 2012)
$harpy(R1, Vb1)$	statistical	alignment links between relational phrases and WordNet verb senses (Grycner and Weikum, 2014)
$wordnetHyponym(Vb1, Vb2)$	semantic	hyponymy link between WordNet verb senses
$lType(R1, TL1)$	semantic	left (domain) type of arguments of a relational phrase
$rType(R1, TR1)$	semantic	right (range) type of arguments of a relational phrase
$yagoHyponym(T1, T2)$	semantic	$T1$ is a subtype of $T2$ in YAGO hierarchy
$candidateHyponym(R1, R2)$	output	relational phrase $R1$ is more specific than $R2$ (without enforcing consistent argument types)
$hyponym(R1, R2)$	output	relational phrase $R1$ is more specific than $R2$

Table 4.1: PSL predicates

Id	**Feature**	**PSL rule**
1	Weeds inclusion	$weedsInclusion(R1, R2) \Rightarrow candidateHyponym(R1, R2)$
2	Patty subsumption	$pattySubsumption(R1, R2) \Rightarrow candidateHyponym(R1, R2)$
3	Harpy alignment	$wordnetHyponym(Vb1, Vb2) \wedge harpy(R1, Vb1) \wedge harpy(R2, Vb2) \Rightarrow candidateHyponym(R1, R2)$
4	Types compatibility	$candidateHyponym(R1, R2) \wedge lType(R1, TL1) \wedge rType(R1, TR1) \wedge lType(R2, TL2) \wedge rType(R2, TR2)$ $\wedge yagoHyponym(TL1, TL2) \wedge yagoHyponym(TR1, TR2) \Rightarrow hyponym(R1, R2)$
5	WordNet hierarchy	$wordnetHyponym(Vb1, Vb2) \wedge wordnetHyponym(Vb2, Vb3) \Rightarrow wordnetHyponym(Vb1, Vb3)$
6	Yago hierarchy	$yagoHyponym(T1, T2) \wedge yagoHyponym(T2, T3) \Rightarrow yagoHyponym(T1, T3)$
7	Asymmetry	$hyponym(R1, R2) \Rightarrow \neg hyponym(R2, R1)$
8	Transitivity	$hyponym(R1, R2) \wedge hyponym(R2, R3) \Rightarrow hyponym(R1, R3)$
9	Acyclicity	$hyponym(R1, R2) \wedge hyponym(R2, R3) \Rightarrow \neg hyponym(R3, R1)$

Table 4.2: PSL rules

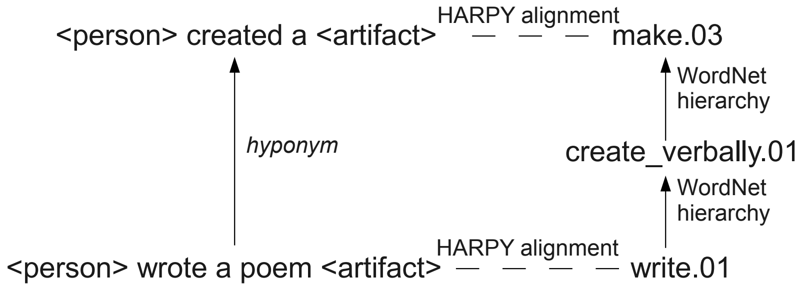


Figure 4.1: HARPY alignment usage

4.3.2 RELLY Overview

RELLY has four stages: data pre-processing, rule weight learning, inference, and thresholding.

First, in the data pre-processing stage, we assign confidence scores of 0 or 1 for the binary-valued semantic predicates in the PSL model. For example, the $wordnetHyponym(Vb1, Vb2)$ confidence score is set to 1 if there is a hyponymy link between verb senses $Vb1$ and $Vb2$ and 0 otherwise. In other cases, the confidence is set to a similarity score of a feature which is represented by a predicate. For example, the $weedsInclusion(R1, R2)$ confidence is equal to the Weeds inclusion score between relations $R1$ and $R2$.

In the next stage the weights of the PSL rules described in Table 4.2 are learned from a small handcrafted graph of relational phrases. The weight learning is performed using an EM algorithm. Later, the most-probable explanation (MPE) state of the output predicates is inferred.

Finally, we export the inferred confidence scores of the predicate *hyponym* and perform additional cleaning. Whenever two links contradict each other (e.g. we have both *hyponym*$(R1, R2)$ and *hyponym*$(R2, R1)$) we remove the link with the lower confidence score. If both predicates have the same confidence score we exclude them both from the final graph. Additionally, we only consider links with a confidence score above an empirically chosen threshold of 0.2.

4.4 Evaluation

In our experiments, we use a large corpus of relational phrases to construct a hypernymy graph using RELLY. We evaluate RELLY using both intrinsic and extrinsic evaluation. In the intrinsic evaluation, we asked human annotators to judge the relationship between two relational phrases and compared results from several hypernymy graphs. In the extrinsic evaluation, we used the hypernymy graph for a real-world document ranking task and measured the mean reciprocal rank (MRR) for a number of methods. In both evaluations, the hypernymy graph constructed by RELLY demonstrates significantly better performance than competing algorithms.

4.4.1 Dataset

We use RELLY to build a hypernymy graph with data from the PATTY and HARPY projects. The input to our system consists of 20,812 relational phrases and the associated argument types extracted from the English-language Wikipedia website using the PATTY system. For simplicity, we only include relational phrases that contain exactly one verb (e.g. "took the throne"), excluding noun phrases (e.g. "member of") and phrases containing multiple verbs (e.g. "hit and run"). The verb "to be" and modal verbs were not considered in the dataset. We also include HARPY alignments to the corresponding verb senses in WordNet for each phrase in the corpus. Additionally, we use a subset of the type-subsumption hierarchy from YAGO consisting of 144 types and 323 subsumption relationships.

During graph inference, RELLY evaluated 7.9M possible hypernymy links using 9.7M ground logical rules and constraints. Ultimately, RELLY produced 35,613 hypernymy links between relational phrases with confidence scores above 0.2. The hypernymy graph consisted of 3,730 roots. Running RELLY on a multi-core 2.27GHz server with 64GB of RAM required approximately 20 hours. For comparison, PATTY produced 8,162 subsumption links out of 350,569 phrases with approximately 2,300 roots.

4.4.2 Intrinsic Evaluation

In our intrinsic evaluation, we assess the precision of hypernymy links inferred by RELLY and compare with the precision of hypernymy graphs of PATTY and HARPY. In this evaluation, we measure precision for both the most confident hypernymy links in the system (precision@100) and the precision of a random sample of 100 hypernymy links. Each set of hypernymy links were presented to several human annotators for labeling.

To measure precision@100, we choose the top 100 hypernymy links using the confidence scores reported by PSL. We similarly choose the top 100 links from PATTY using the PATTY subsumption score. Since HARPY does not provide confidence scores, we were unable to compute precision@100 for HARPY.

For each of the three systems, we used the full set of hypernymy links they produce, which consisted of 8,000 links from PATTY, 600,000 links from HARPY and 35,000 links from RELLY. We randomly sampled 100 hypernymy links from each of these systems.

We presented the selected hypernymy links to several human annotators. The labeling task required the annotator to judge the relationship between two relational phrases in a hypernymy link. For each relational phrase, we provided annotators with type information about the phrase arguments (domain and

	Prec.	**Range**	**Cvg.**
precision@100			
RELLY	0.87	0.81 - 0.92	35K
PATTY	0.83	0.76 - 0.90	8K
random sample			
RELLY	0.78	0.71 - 0.84	35K
PATTY	0.75	0.68 - 0.82	8K
HARPY	0.43	0.35 - 0.52	600K

Table 4.3: Intrinsic evaluation

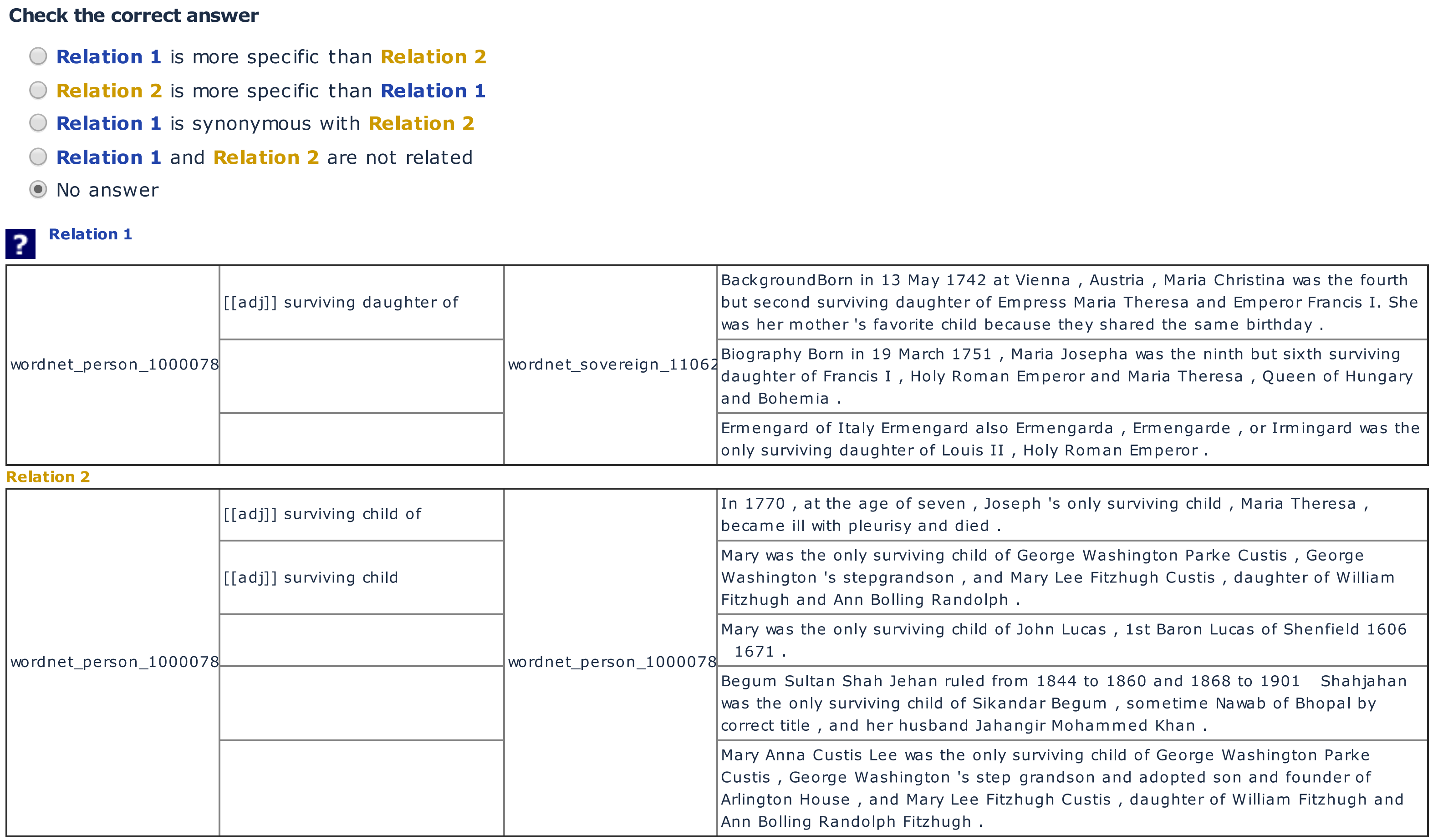
range) and examples of sentences that use the relational phrase. Based on this information, annotators could make one of four judgments:

1. the phrases are unrelated.
2. the phrases are synonymous.
3. the first phrase is more specific than the second phrase.
4. the second phrase is more specific than the first phrase.

The example of the evaluation survey is presented in Figure 4.2. This evaluation task had good inter-annotator agreement, with a Cohen's Kappa of 0.624. Separately, the precision@100 dataset had Cohen's Kappa of 0.708 and the randomly sampled dataset had Cohen's Kappa of 0.521.

We show the results of the intrinsic evaluation in Table 4.3 with 0.9-confidence Wilson score interval (Brown et al., 2001). In comparison to HARPY and PATTY, RELLY has higher precision for both precision@100 and random evaluations. Precision in RELLY is comparable to PATTY, but RELLY has more than four times as many hypernym links. HARPY has far more hypernymy links, but with a precision of 0.43, we find that many of these links are incorrect.

Table 4.4 includes example hypernymy links from RELLY. There are examples where PATTY's subsumption is a dominant signal ("<*person*> publicly accused <*person*>" ⇒ "<*person*> accused <*person*>"). We also observe YAGO type hierarchy influence ("<*athlete*> played for <*team*>" ⇒ "<*person*> played for <*organization*>"), as well as the influence of combined WordNet hierarchy with HARPY alignments ("<*person*> marry daughter <*person*>" ⇒ "<*person*> joins <*person*>"). The advantage of RELLY is that it computes the final graph jointly and incorporates transitivity, asymmetry and acyclicity rules. It leads to less semantic drift in longer hypernymy chains (e.g. Figure 4.3) compared with PATTY where "<*organization*> merged <*organization*>" can lead to "<*team*> beat <*team*>."



Check the correct answer

- ○ **Relation 1** is more specific than **Relation 2**
- ○ **Relation 2** is more specific than **Relation 1**
- ○ **Relation 1** is synonymous with **Relation 2**
- ○ **Relation 1** and **Relation 2** are not related
- ◉ No answer

Relation 1

wordnet_person_1000078	[[adj]] surviving daughter of	wordnet_sovereign_11062	BackgroundBorn in 13 May 1742 at Vienna , Austria , Maria Christina was the fourth but second surviving daughter of Empress Maria Theresa and Emperor Francis I. She was her mother 's favorite child because they shared the same birthday .
			Biography Born in 19 March 1751 , Maria Josepha was the ninth but sixth surviving daughter of Francis I , Holy Roman Emperor and Maria Theresa , Queen of Hungary and Bohemia .
			Ermengard of Italy Ermengard also Ermengarda , Ermengarde , or Irmingard was the only surviving daughter of Louis II , Holy Roman Emperor .

Relation 2

wordnet_person_1000078	[[adj]] surviving child of	wordnet_person_1000078	In 1770 , at the age of seven , Joseph 's only surviving child , Maria Theresa , became ill with pleurisy and died .
	[[adj]] surviving child		Mary was the only surviving child of George Washington Parke Custis , George Washington 's stepgrandson , and Mary Lee Fitzhugh Custis , daughter of William Fitzhugh and Ann Bolling Randolph .
			Mary was the only surviving child of John Lucas , 1st Baron Lucas of Shenfield 1606 1671 .
			Begum Sultan Shah Jehan ruled from 1844 to 1860 and 1868 to 1901 Shahjahan was the only surviving child of Sikandar Begum , sometime Nawab of Bhopal by correct title , and her husband Jahangir Mohammed Khan .
			Mary Anna Custis Lee was the only surviving child of George Washington Parke Custis , George Washington 's step grandson and adopted son and founder of Arlington House , and Mary Lee Fitzhugh Custis , daughter of William Fitzhugh and Ann Bolling Randolph Fitzhugh .

Figure 4.2: Excerpt from the evaluation survey

Hyponym relational phrase			Hypernym relational phrase		
Domain	Text pattern	Range	Domain	Text pattern	Range
<head of state>	abdicated in favor of	sovereign	<person>	resigns as	<person>
<person>	publicly accused	<person>	<person>	accused	<person>
<person>	marry daughter	<person>	<person>	joins	<person>
<person>	had paid	<person>	<person>	interacted with	<person>
<athlete>	played for	team	<person>	played for	<organization>
<album>	featured guest appearances from	<person>	<artifact>	featured	<person>
<person>	attended college in	<city>	<person>	attended in	<city>
<priest>	was appointed bishop by	<spiritual_leader>	<person>	when appointed by	<person>
<person>	defected to	<country>	<person>	fled to	<country>
<person>	intercepted pass from quarterback	<person>	<person>	intercepted pass	<person>
<person>	praised writing	<person>	<person>	praised	<person>
<actor>	was reunited with costar	<person>	<person>	was combined with	<person>

Table 4.4: Example RELLY hypernymy links

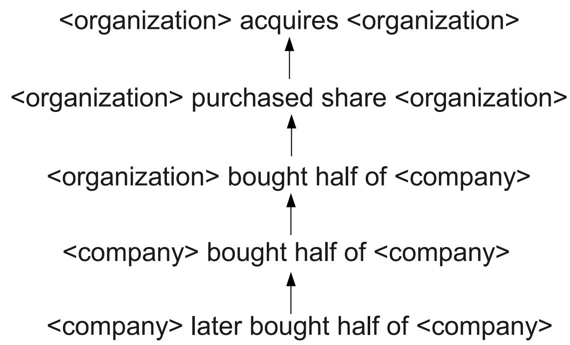


Figure 4.3: Chain of hypernymy

4.4.3 Ablation Study

Two advantages of RELLY that we have highlighted are easily incorporating new information sources and collectively enforcing global constraints. To analyze the influence of these system components, we performed an ablation study where we omitted PSL rules corresponding to specific model features. Using this approach, we quantify the importance of these features to RELLY's performance.

First, we demonstrate the value of type information in determining hypernymy. The YAGO type hierarchy allows RELLY to detect hypernymy links between relational phrases where types do not match exactly, but are compatible through type subsumption. When the YAGO type hierarchy rules are omitted from the model, coverage is reduced dramatically; the resulting hypernymy graph contains only 12,000 hypernymy links in contrast to the 35,000 links in the original model. Additionally, removing YAGO type information harms precision, with a precision of 0.75 ± 0.09 with 0.9-confidence Wilson score interval for a random sample of 100 examples.

Next, we show how global constraints on the hypernymy graph such as anti-symmetry and acyclicity improve the quality of the hypernymy graph. Since the relational phrases generated by PATTY are clustered to find synonymous relations, these global constraints prevent RELLY from merging clusters. When the anti-symmetry and acyclicity rules were removed from the model, the resulting hypernymy graph included approximately 500 additional hypernymy links, while 10 existing links were removed. We manually evaluated the newly introduced links, and found that the majority of links were false positives.

4.4.4 Entailment Graph Induction

We compared the performance of PSL against the Integer Linear Programming (ILP) formulation by Berant et al. (2011). The comparison was performed on the task of creating entailment graphs as described by Berant et al. (2011). this

	Prec.	**Rec.**	**F1**
Berant et al. (2011)	0.422	0.434	0.428
PSL	**0.461**	**0.435**	**0.447**

Table 4.5: Results for Entailment graphs induction

task is strongly related to finding hypernyms of relational phrases. The experiments were executed on the dataset of 10 manually annotated graphs. In total this dataset contains 3,427 positive and 35,585 negative examples. Our model uses the transitivity rule ($entails(A, B) \wedge entails(B, C) \Rightarrow entails(A, C)$). We also include the local entailment scores ($score(A, B) \Rightarrow entails(A, B)$) which were released by Berant et al. (2011). Table 4.5 presents micro-averaged precision, recall and F1 scores for this comparison.

PSL was much faster than the other exact methods used for this problem. To compare efficiency we measured the run-time of our method. Without any graph decomposition it took on average 232 seconds. The experiments were performed on a multi-core 2.67GHz server with 32GB of RAM. The methods reported in (Berant et al., 2012a), which did not utilize graph decomposition method, had run-time above 5000 seconds.

4.4.5 Extrinsic Evaluation

The ultimate goal of producing a high-quality hypernymy graph is to deepen our understanding of natural language and improve performance on the many NLP applications. One such application is document retrieval, where billions of queries are performed each day through search engines. In our extrinsic evaluation, we demonstrate how a hypernymy graph can improve performance on a document ranking and retrieval task.

We consider a task where an input query document is compared to a corpus of documents with the aim of finding the most relevant related documents. To isolate the evaluation to relational phrases, we *anonymize* the documents, by replacing all named entities and noun phrases with placeholders. For example, the sentence "`The villain has already fled to the Republica de Isthmus`" is anonymized to "`* has already fled to *`." Anonymized retrieval has potential applications in security and for sensitive documents.

We collected a dataset consisting of movie plot summaries from two different websites, Wikipedia and the Internet Movie Database (IMDB). We chose plot synopses from 25 James Bond movies and 23 movies based on the Marvel Comics characters. For each plot synopsis, we have two plot descriptions: one from Wikipedia and another from IMDB. Given a query in the form of an anonymized plot description from one website, the task is to rank the anonymized plot descriptions from the other dataset using relational phrase similarity. For example, given a query plot description of "Iron Man" from

Wikipedia, rank plot descriptions from IMDB with the goal of maximizing the ranking of the corresponding "Iron Man" plot summary. We evaluate the quality of these rankings using the mean reciprocal rank (*MRR*) score, $\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}$. Here, Q is the number of documents in the collection (i.e. 2*48 = 96) and rank_i is the position of the counterpart document in the ranking of document i.

As baseline algorithms, we use a unigram word2vec model and a bigram model. In the unigram word2vec model documents are represented by the average of the 300-dimensional word vectors trained on part of Google News dataset (about 100 billion words) (Mikolov et al., 2013). We could not use the bigram word2vec model because of the frequent occurrence of the placeholder symbol. In the bigram model, documents are represented by vectors in the bag-of-bigrams model with bigram frequency weights. The similarity measure in both cases is the cosine similarity measure.

As the first of our approaches we proposed a solution purely based on relational phrases. In the *relational phrases* model we extract relational phrases from a text and we map them to their synsets from PATTY (clusters of synonyms). A phrase is mapped to a synset if the Jaccard similarity between tokens of extracted relation and tokens of one of the phrases in the synset is above a threshold. Next we represent the document as a vector of the relational phrase synsets weighted by the frequency of the synset in the document (bag-of-relational_phrases). The similarity score between two documents is the cosine similarity between two vectors representing two documents. The ranking is created based on the similarity scores. In the *relational phrases + hypernyms* model we add hypernyms of the extracted relational phrases to the document vector (based on the hypernymy graph). Hypernyms are additionally weighted by the confidence score produced by the algorithm described in the Section 4.3. In the second approach, we combine relational phrases models with the best of the baselines. The similarity score is then equal to $\lambda sim_1 + (1-\lambda) sim_2$. The λ parameter is trained on a different dataset (2*8 plot descriptions of Harry Potter movies). Training was performed by maximization of the MRR score using grid search. We consider the combination of the *bigram* model with *relational phrases*, as well as the combination of the *bigram* model with *relational phrases + hypernyms*.

The results of the experiment are presented in Table 4.6. The best MRR score was obtained by *relational phrases + hypernyms + bigrams* model. The number of samples, 96, was large enough for statistical significance. We performed a paired t-test for MRR between each of these methods. The obtained p-values were below 0.05.

4.5 Related Work

The biggest sources of hypernyms, subsumptions, and hierarchical structure can be found in existing knowledge bases. Examples of these are Freebase

	MRR score
word2vec	0.26
bigram	0.55
relational phrases	0.28
+ hypernyms	0.25
+ bigrams	0.58
+ hypernyms + bigrams	**0.60**

Table 4.6: Extrinsic evaluation (Bond & Marvel)

(Bollacker et al., 2008), YAGO, DBPedia (Lehmann et al., 2015), and Google Knowledge Vault (Dong et al., 2014). However, these knowledge bases are mainly concentrated on named entities and noun phrases, and the variety of relations between entities is much smaller. Relations and information about them are underrepresented.

OpenIE systems try to solve this problem by extracting new relations from natural text. These new relations do not necessarily follow the standard schema of knowledge bases. Additionally, these systems often organize the newly extracted relations by clustering or hierarchy construction. A first attempt to extract and cluster similar relations was presented in DIRT. This work was followed by projects such as ReVerb, PATTY, WiseNet, NELL (Carlson et al., 2010), and RESOLVER (Yates and Etzioni, 2009). PATTY and WiseNet also introduced semantic types to their concept of relational phrases. All of these systems rely on the co-occurrence of arguments of clustered relations. A different approach was presented in PPDB, where the authors cluster phrases based on the similarity of translations to other languages.

Of these systems, only PATTY attempted to create a hierarchy of relations and the result was very sparse. HARPY aimed to overcome this problem by disambiguating and aligning relational phrases with WordNet, and performing a simple reconstruction of the WordNet hierarchy on top of relational phrases from PATTY. A very similar problem was addressed in the entailment graph project (Levy et al., 2014). The authors automatically created graphs of entailments between propositions, using ILP as one of the main components. Propositions can be encoded as triples of form *(subject, relation, object)*. Edges in the entailment graph occur between these triples, whereas edges connect typed relations in PATTY and HARPY. Moreover, the relations in the propositions were mainly limited to single verbs, whereas in our case we also consider longer relational phrases. Relations with semantic types were also used in typed entailment graphs (Berant et al., 2011). However, the type hierarchy was not considered there, which prevented from creating links between two relations with different semantic types. The input dataset was also smaller – the biggest graph consisted of 118 relations.

Although there is a scarcity of automatically created taxonomies of relations, there exist several manually curated taxonomies. Manually crafted verb or relation hierarchies are available in WordNet, VerbNet and FrameNet. WordNet has 13,767 verb synsets, which are organized into a hierarchy with 13,239 hypernymy links.

Automatic construction of taxonomies of named entities or noun phrases has received much more attention than organization of verbs or relations. In Snow et al. (2006), the WordNet taxonomy was extended by 10,000 novel noun synsets with hypernym-hyponym links. In Bansal et al. (2014), the authors reconstructed WordNet's noun hypernymy/hyponymy hierarchy from scratch using a probabilistic graphical model formulation. Another method of organizing noun phrases was proposed by Mehdad et al. (2013), where an entailment graph of noun phrases was constructed.

Building a hypernymy graph for relational phrases is strongly related with the textual entailment task (Dagan et al., 2010). This concept was introduced in the Recognizing Textual Entailment (RTE) shared task (Dagan et al., 2005). Instead of short typed relational phrases, the input data are two texts – the entailing text T and the hypothesis text H. According to Dagan et al. (2005)'s definition, "T entails H if, typically, a human reading T would infer that H is most probably true."

In RELLY, we use Probabilistic Soft Logic (PSL) as the main ingredient of our approach. PSL was successfully used for numerous other applications including knowledge graph construction (Pujara et al., 2013), trust in social networks (Huang et al., 2012b), ontology alignment (Broecheler and Getoor, 2009), and social group modeling (Huang et al., 2012a).

4.6 Conclusion

This work presents RELLY, a scalable method for integrating statistical and semantic signals to produce a hypernymy graph of relational phrases. We used RELLY to create a hypernymy graph that has both high coverage and precision, as shown in our evaluation. RELLY is extensible and can easily incorporate additional information sources and features. The hypernymy graph of relational phrases could potentially be useful for many problems of natural language processing and information retrieval. For example, we applied the hypernymy graph to a document-relevance task, which we used to evaluate RELLY extrinsically. In a future work, RELLY could incorporate more information sources and statistical signals and be expanded to infer multi-verb or noun relational phrases. The RELLY resource is publicly available at `www.mpi-inf.mpg.de/yago-naga/patty/`.

Chapter 5

Relational Clustering with PSL

In the previous chapter (Chapter 4), we discussed the hierarchy construction of relational phrases using Probabilistic Soft Logic (PSL). In this chapter, we explore the applicability of PSL framework to finding clusters of synonymous relational phrases (Grycner et al., 2014).

5.1 Introduction

To fully understand a written text, a machine must be able to understand the meaning of the relational phrases occurring within it. Relational phrases are textual representations of relations which occur between common noun phrases (e.g., "the movie star") or named entities (e.g., "George Clooney") with the same *semantic type signature*. For example, "Bob is married to Alice" connects two entities of the type <*person*>, and is an instance of the relational phrase "<*person*> is married to <*person*>." In this case, both the left side (domain) and the right side (range) arguments of the phrase have the type <*person*>. The problem of discovering and organizing relational phrases was addressed in many previous works (Fader et al., 2011; Mohamed et al., 2011; Nakashole et al., 2012; Moro and Navigli, 2012).

A key step toward detecting the meaning of these phrases is to find their synonyms, in a task known as semantic clustering. In this problem the goal is to group together phrases which have similar meanings. For example, "<*person*> is married to <*person*>" should be clustered together with a relation "<*person*> is a spouse of <*person*>." This kind of clustering has many applications, including paraphrase detection or generation, information extraction, semantic parsing, and question answering.

State of the art systems for clustering of relational phrases concentrate mostly on two aspects – syntactic similarity and argument co-occurrence statistics. The PATTY system (Nakashole et al., 2012) performs clustering using both argument overlap statistics and phrase syntactic similarity. WiseNet (Moro and Navigli, 2012) uses similar ideas but includes a soft clustering option.

Universal Schema (Riedel et al., 2013) applies collaborative filtering on the relation-arguments co-occurrence matrix to find similarity scores between relational phrases. Moreover, it learns vector representations for phrases, which allows to encode asymmetry between them. Universal Schema additionally combines closed IE (prespecified) with open IE (newly discovered) phrases. NELL (Carlson et al., 2010) also organizes extracted relations into groups of synonyms, but the number of clusters is limited to a fixed set of prespecified relations. In our case, the number of clusters is unknown. DIRT (Lin and Pantel, 2001a) was one of the first works which addressed finding synonymous relational phrases using the distributional hypothesis (Harris, 1954). RESOLVER (Yates and Etzioni, 2009) extended that idea with computing pairwise similarities between relations and applying Hierarchical Agglomerative Clustering (HAC). Unlike PATTY, WiseNet or our PSL models, both of the aforementioned systems work with untyped relational phrases. The PPDB paraphrase database (Ganitkevitch et al., 2013) uses a very different approach, employing bilingual texts. However, they concentrate on paraphrases of textual phrases rather than finding synonymous or similar relations.

The above approaches are capable of performing semantic clustering at large scale. However, they are limited in terms of the relational features used. OntExt (Mohamed et al., 2011) uses richer contextual information for clustering but can extract and cluster relations of only one pair of argument semantic types at one time. There is a need for more powerful methods which can incorporate many types of relational features, yet can solve large-scale semantic clustering problems.

To address this challenge, in this work, we present a method for clustering relational phrases using a statistical relational learning system called Probabilistic Soft Logic (PSL) (Kimmig et al., 2012). The proposed method has several advantages. First, the PSL modeling language allows us to easily build a rich model incorporating both similarity measures and relational features. Moreover, it is efficient enough to be applied to a dataset containing 200,000 relational phrases. More details about PSL are included in Section 4.2.2.

We perform a quantitative evaluation of the proposed PSL model on a small dataset. The performance of our approach is compared against a set of baselines, including textual similarity and argument overlap, demonstrating the efficacy of the technique. Additionally, we report the outcome of the PSL method on the large-scale dataset extracted by PATTY (Nakashole et al., 2012), illustrating that the proposed method is highly scalable.

5.2 Semantic Clustering using PSL

The proposed models begin with the ideas of (Nakashole et al., 2012) and (Moro and Navigli, 2012), and build upon them using relational learning techniques. This is accomplished using probabilistic soft logic, a declarative language for specifying templates for probabilistic graphical models. The resulting models, known as hinge-loss Markov random fields (HL-MRFs) (Bach et al., 2013), define probability densities over continuous random variables in the range [0,1]. Inference in this setting is a convex optimization task, which can be solved efficiently and at scale. PSL has been successfully applied to problems such as entity resolution (Pujara et al., 2013), trust in social networks (Huang et al., 2012b), ontology alignment (Broecheler and Getoor, 2009), and social group modeling (Huang et al., 2012a). A PSL model is specified by a set of weighted first order logical rules. The resulting Markov random field gives higher probability density to states where the rules are closer to being satisfied, as measured by a continuous relaxation of Boolean logic. We now describe our proposed approach.

Predicates: The PSL models introduced in this work use the following predicates:

- $args(R_1, X, Y)$ - is true when relational phrase R_1 occurs with domain argument X and range argument Y.
- $types(R_1, T_1, T_2)$ - is true when T_1 and T_2 are domain and range types of relation R_1.
- $phrase(R_1, STR_1)$ - is true when STR_1 is a textual representation of relational phrase R_1. It is an implementation detail which allows us to separate numerical identifier R_1 from string STR_1 in the logic of our models.
- $simPattern(STR_1, STR_2)$ - is true when Jaccard similarity of tokens of strings STR_1 and STR_2 is above a threshold.
- $similar(R_1, R_2)$ - an open predicate which is true when relational phrases R_1 and R_2 belong to the same cluster.

PSL rules: The rules used in the PSL models are shown in Table 5.1. We employed standard similarity functions for argument similarity and textual similarity (rules 1,2,3,4). Rule 5 ensures that the similarity relation between relational phrases is transitive. Additionally, we define a prior on the inference predicate *similar* (rule 6). It says that we should assume that two phrases are not *similar* with a small weight. This can be overridden with evidence as defined in the other rules.

PSL models: We have developed three sets of rules used by the PSL framework. The full list of rules used by our models is shown in Table 5.1. Each model runs in two stages – weight learning and inference. Weight learning is performed on a separate training data set. In the inference stage, for each model, the defined rules with learned weights are applied to the test data. We used the following models:

- *PSL_sim*: In this model we use argument and textual similarity without types (rules 1,2,6)
- *PSL_types*: In this model we add to *PSL_sim* information about the semantic types of the domain and the range (rules 3,4,6)

Id	Name	Rule
1	Argument similarity (without types)	$args(R_1, X, Y) \wedge args(R_2, X, Y) \Rightarrow similar(R_1, R_2)$
2	Textual similarity (without types)	$phrase(R_1, STR_1) \wedge phrase(R_2, STR_2) \wedge simPattern(STR_1, STR_2) \Rightarrow similar(R_1, R_2)$
3	Argument similarity and type compatibility	$args(R_1, X, Y) \wedge args(R_2, X, Y) \wedge types(R_1, T_1, T_2) \wedge types(R_2, T_1, T_2) \Rightarrow similar(R_1, R_2)$
4	Textual similarity and type compatibility	$phrase(R_1, STR_1) \wedge phrase(R_2, STR_2) \wedge simPattern(STR_1, STR_2) \wedge types(R_1, T_1, T_2) \wedge types(R_2, T_1, T_2) \Rightarrow similar(R_1, R_2)$
5	Transitivity rule	$similar(R_1, R_2) \wedge similar(R_2, R_3) \Rightarrow similar(R_1, R_3)$
6	Negative prior	$\neg similar(R_1, R_2)$

Table 5.1: Rules used in PSL models

- *PSL_trans*: This model extends *PSL_types* with transitivity rules (rules 3,4,5,6)

5.3 Evaluation

To assess the quality of PSL models we evaluated the proposed models and compared the results against baseline algorithms.

Baselines: To be able to compare PSL models we have implemented a set of baselines. All of these methods predict similarity between relational phrases. We consider four baselines:

- *Random:* for every pair of relations we randomly decide whether they are similar or not.
- *Arguments overlap:* two relations are similar if they occur with the same arguments. The higher the percentage of common arguments is, the more similar the relations are. This follows the distributional hypothesis (Harris, 1954). For example, if the relations "is married to" and "is husband of" occurred only with arguments "George Clooney" and "Amal Alamuddin" then they would be clustered together because of 100% arguments overlap. The threshold above which two relations are clustered together is set using training data with the grid search algorithm.
- *String similarity:* to determine the similarity of two relations we use the Jaccard similarity of the sets of tokens of their textual representations. Again, two relations are said to be similar if their textual similarity is above a threshold, chosen using a grid search.
- *String & types similarity:* we use string similarity and add an additional feature, types compatibility. Type compatibility introduces a constraint which requires that two relations can be similar only if the semantic types of the domains of each relation are equal and the semantic types of the ranges are also equal.

Dataset: We prepared the ground truth semi-automatically based on the clusters of relational phrases produced by the PATTY system (Nakashole et al., 2012). For the first experiment we used all together 526 relational phrases which were divided into training and test sets. The training set contains 399 phrases and 3,454 argument-relation-argument triples. These phrases were divided into 229 clusters. The test set contains 127 phrases and 1,494 argument-relation-argument triples. These phrases were divided into 58 clusters.

Evaluation setup: We treat our problem in two ways – first as a link prediction task and second as a clustering problem. In the first setting the goal is to predict which relational phrases are similar. For every pair of relational phrases the algorithms determine whether there should be a similarity link between them. The similarity links form a similarity graph. Next we compare the produced similarity links against the similarity links in the ground truth.

As the evaluation metrics for the link prediction task we use F1 score and area under the receiver operating characteristic curve (AUC).

In the second setting the goal is to form clusters of synonymous relational phrases. In this case we take the output of the previous setting and organize phrases into clusters by applying a connected component detection algorithm (Hopcroft and Tarjan, 1973). Relational phrases which are in the same component are put to the same cluster. For the purpose of the evaluation we assume that phrases inside a cluster are all connected with similarity links. Again, we compare similarity links against the ground truth and compute F1 score and AUC. Additionally, we compute Normalized Mutual Information (NMI) (Manning et al., 2008b) – a metric used specifically for the clustering problem.

In order to compute statistical significance the experiment was repeated 20 times on random subsets of the patterns in the test data. We performed a paired t-test for all metrics (F1, AUC for link prediction and clustering; NMI for clustering) of the PSL models results against baselines and obtained p-values below 0.05.

Results: The results of the evaluation are shown in Table 5.2. The *String & types similarity* method performs the best out of all baselines in terms of all metrics and settings. The *PSL_types* and *PSL_trans* models have higher scores than baselines in both link prediction and clustering tasks. Moreover, in the link prediction task, we can see the difference between *PSL_types* and *PSL_trans* models. This shows us the influence of the *transitivity rule*.

After the application of the connected components detection algorithm there is no difference between *PSL_types* and *PSL_trans* models. This means that the transitivity rule in PSL models tends to play the same function as the connected components detection algorithm. However, PSL allows us to incorporate it in a single model rather than in a few separate algorithms. The cause of the equal scores of *PSL_types* and *PSL_trans* could be the size of the dataset. We do not observe big clusters in the data, therefore the transitivity cannot be often applied.

	Link prediction		Clustering		
	F1	AUC	F1	AUC	NMI
Random	0.0174	0.5000	0.0221	0.5000	0.8020
Arguments overlap	0.1204	0.5337	0.1167	0.5349	0.8740
String similarity	0.4017	0.6281	0.5878	0.7312	0.9236
String & types similarity	0.4085	0.6283	0.6350	0.7326	0.9321
PSL_sim	0.3675	0.6146	0.5684	0.7205	0.9193
PSL_type	0.6690	0.7538	**0.7151**	**0.8011**	**0.9448**
PSL_trans	**0.7290**	**0.8017**	**0.7151**	**0.8011**	**0.9448**

Table 5.2: Evaluation

Studies on the PATTY dataset: We also investigated the performance of the best performing PSL model on a larger amount of data. To increase the speed of the algorithm we used precomputed similarities of textual representations of relational phrases. We used a subset of the patterns taken using PATTY (Nakashole et al., 2012). The subset consisted of 200,000 patterns, contained information about 1,158,417 argument-relation-argument triples and was originally organized into 162,289 clusters. As a result of running the algorithm we obtained 144,634 clusters. Some example clusters created during this process are shown in Table 5.3.

Cluster	Domain	Phrases	Range
1	sovereign	became emperor as; ascended the throne as; succeeded as; took the throne as	head of state
2	person	be consecrated by; enthroned as; also consecrated;	priest
3	actor	had starred in; best known for playing on; again starred in;	event
4	person	and defeated the; successfully defended against;	team

Table 5.3: Example clusters of phrases

5.4 Conclusion

In this work, we demonstrate an approach for semantic clustering of relational phrases. This approach uses the PSL framework for modeling similarities between phrases. In the experiments, we showed that our method outperforms several baselines and is capable of reconstructing a clustering performed by PATTY (Nakashole et al., 2012). Moreover, we applied PSL to a dataset whose size is comparable with datasets used by state of the art systems for extraction and clustering of relational phrases. Since we used basic features and basic similarity measures there is still the potential for further improvement. The future work could include checking the expressiveness of PSL to incorporate other similarity measures into our models. Furthermore, other sources of relational phrases can be used (e.g. WiseNet (Moro and Navigli, 2012)).

Chapter 6

POLY: Mining Relational Paraphrases

The projects presented in previous chapters used textual and distributional similarity as the main cues for reasoning about relationships between relational phrases. Additional similarity signals could improve both precision and recall of relational paraphrases resources. In this chapter, we present POLY, a resource and a method for relation argument typing and finding paraphrases of semantically typed relational phrases using translation information (Grycner and Weikum, 2016).

6.1 Introduction

Motivation. Information extraction from text typically yields relational triples: a binary relation along with its two arguments. Often the relation is expressed by a verb phrase, and the two arguments are named entities. We refer to the surface form of the relation in a triple as a *relational phrase.* Repositories of relational phrases are an asset for a variety of tasks, including information extraction, textual entailment, and question answering.

This work presents a new method for systematically organizing a large set of such phrases. Specifically, we aim to construct equivalence classes of synonymous phrases, analogously to how WordNet organizes unary predicates as noun-centric synsets (aka. semantic classes). For example, the following relational phrases should be in the same equivalence class: “sings in,” “is vocalist in,” “voice in” denoting a relation between a musician and a song.

State of the Art and its Limitations. Starting with the seminal work on DIRT (Lin and Pantel, 2001a), there have been various attempts on building comprehensive resources for relational phrases. Recent endeavors of this kind include PATTY (Nakashole et al., 2012), WiseNet (Moro and Navigli, 2012) and DEFIE (Bovi et al., 2015). Out of these DEFIE is the cleanest resource. However, the equivalence classes tend to be small, prioritizing precision over recall. On the other hand, PPDB (Ganitkevitch et al., 2013; Pavlick et al., 2015) offers the largest repository of paraphrases. However, the paraphrases are not relation-centric and they are not semantically typed. So it misses out on the opportunity of using types to distinguish identical phrases with different semantics, for example, *performance in* with argument types *musician* and *song* versus *performance in* with argument types *athlete* and *competition*.

Our Approach. We start with a large collection of relational triples, obtained by shallow information extraction. Specifically, we use the collection of Faruqui and Kumar (2015), obtained by combining the OLLIE tool with Google Translate and projecting multilingual sentences back to English. Note that the task addressed in that work is relational triple extraction, which is orthogonal to our problem of organizing the relational phrases in these triples into synonymy sets.

We canonicalize the subject and object arguments of triples by applying named entity disambiguation and word sense disambiguation wherever possible. Using a knowledge base of entity types, we can then infer prevalent type signatures for relational phrases. Finally, based on a suite of judiciously devised probabilistic distance measures, we cluster phrases in a type-compatible way using a graph-cut technique. The resulting repository contains about 1 Million relational phrases, organized into around 160,000 clusters.

Contribution. This work makes the following contributions:

- a novel method for constructing a large repository of relational phrases, based on judicious clustering and type filtering;
- a new linguistic resource, coined POLY, of relational phrases with semantic typing, organized in equivalence classes;
- an intrinsic evaluation of the POLY resource, demonstrating its high quality in comparison to PATTY and DEFIE;
- an extrinsic evaluation of POLY, demonstrating its benefits for question answering.

The POLY resource is publicly available [1].

6.2 Method Overview

Our approach consists of two stages: *relational phrase typing* and *relational phrase clustering*. In Section 6.3, we explain how we infer semantic types of the arguments of a relational phrase. In Section 6.4, we present the model for computing synonyms of relational phrases (i.e., paraphrases) and organizing them into clusters.

[1] www.mpi-inf.mpg.de/yago-naga/poly/

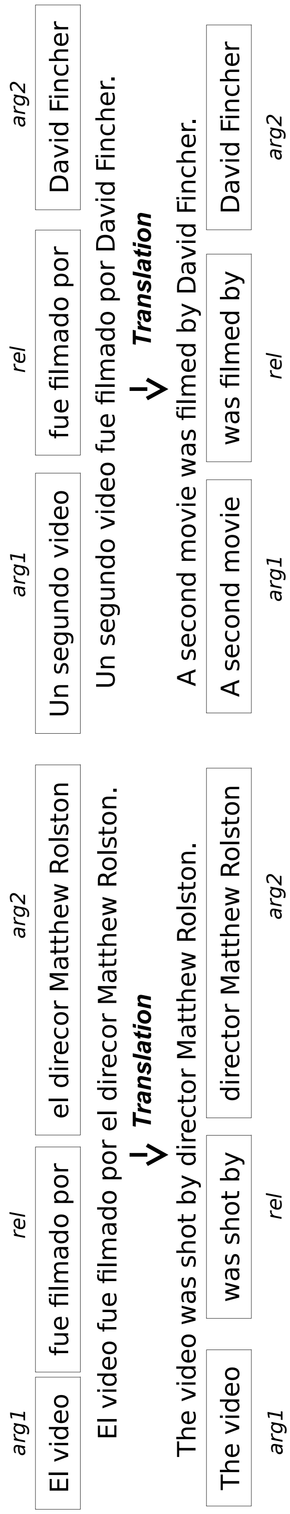


Figure 6.1: Multilingual input sentences and triples

A major asset for our approach is a large corpus of multilingual sentences from the work of Faruqui and Kumar (2015). That dataset contains sentences from Wikipedia articles in many languages. Each sentence has been processed by an Open Information Extraction (OpenIE) method (Banko et al., 2007), specifically the OLLIE tool (Mausam et al., 2012), which produces a triple of surface phrases that correspond to a relational phrase candidate and its two arguments (subject and object). Each non-English sentence has been translated into English using Google Translate, thus leveraging the rich statistics that Google has obtained from all kinds of parallel multilingual texts. Altogether, the data from Faruqui and Kumar (2015) provides 135 million triples in 61 languages and in English (from the translations of the corresponding sentences). This is the noisy input to our method. Figure 6.1 shows two Spanish sentences, the extracted triples of Spanish phrases, the sentences' translations to English, and the extracted triples of English phrases.

The figure shows that identical phrases in the foreign language - "fue filmado por" - may be translated into different English phrases: "was shot by" vs. "was filmed by," depending on the context in the respective sentences. This is the main insight that our method builds on. The two resulting English phrases have a certain likelihood of being paraphrases of the same relation. However, this is an uncertain hypotheses only, given the ambiguity of language, the noise induced by machine translation and the potential errors of the triple extraction. Therefore, our method needs to de-noise these input phrases and quantify to what extent the the relational phrases are indeed synonymous. We discuss this in Sections 6.3 and 6.4.

6.3 Relation Typing

This section explains how we assign semantic types to relational phrases. For example, the relational phrase "wrote" could be typed as "*<author>* wrote *<paper>*," as one candidate. The typing helps us to disambiguate the meaning of the relational phrase and later find correct synonyms. The relational phrase "shot" could have synonyms "directed" or "killed with a gun." However, they represent different senses of the phrase *shot*. With semantic typing, we can separate these two meanings and determine that "*<person>* shot *<person>*" is a synonym of "*<person>* killed with a gun *<person>*," whereas "*<director>* shot *<movie>*" is a synonym of "*<director>* directed *<movie>*."

Relation typing has the following steps: argument extraction, argument disambiguation, argument typing, and type filtering. The overview of the whole process is shown in Figure 6.2. The output is a set of candidate types for the left and right arguments of each English relational phrase.

6.3.1 Argument Extraction

For the typing of a relational phrase, we have to determine words in the left and right arguments that give cues for semantic types. To this end, we identify

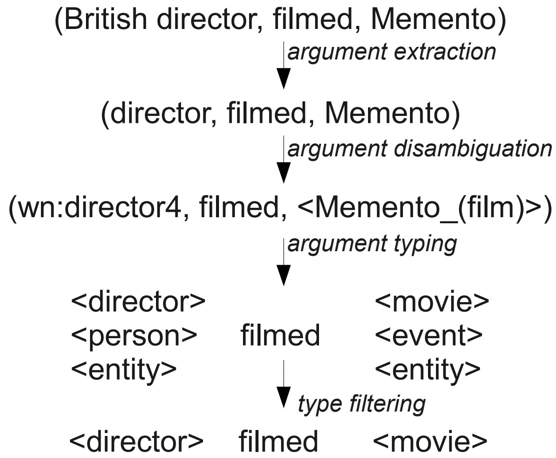


Figure 6.2: Relation typing pipeline

named entities, whose types can be looked up in a knowledge base, and the head words of common noun phrases. As output, we produce a ranked list of entity mentions and common nouns.

To create this ranking, we perform POS tagging and noun phrase chunking using Stanford CoreNLP (Manning et al., 2014) and Apache OpenNLP[2]. For head noun extraction, we use the YAGO Javatools[3] and a small set of manually crafted regular expressions. Since the input sentences were the results of a machine translation system we could not use dependency parsing, because sentences were often grammatically incorrect.

Finally, we extract all noun phrases which contain the same head noun. These noun phrases are then sorted according to their lengths.

For example, for input phrase *contemporary British director who also created "Inception"*, our method would yield *contemporary British director*, *British director*, *director* in decreasing order.

6.3.2 Argument Disambiguation

The second step is responsible for the disambiguation of the noun phrase and named entity candidates. We use the YAGO3 knowledge base (Mahdisoltani et al., 2015) for named entities, and WordNet (Fellbaum, 1998) for noun phrases. We proceed in the ranking order of the phrases from the first step.

We employ several heuristics to find the best sense. Candidate senses are looked up in YAGO3 and WordNet, respectively, and each candidate is scored. The scores are based on the following:

[2] `https://opennlp.apache.org/`
[3] `https://www.mpi-inf.mpg.de/yago-naga/javatools/`

- Frequency count prior: This is the number of Wikipedia incoming links for named entities in YAGO3, or the frequency count of noun phrase senses in WordNet.
- Wikipedia prior: We increase scores of YAGO3 entities whose URL strings (i.e., Wikipedia article names) occur in the Wikipedia page from which the triple was extracted.
- Translation prior: We boost the scores of senses whose translations occur in the original input sentence. For example, the word “stage” is disambiguated as *opera stage* rather than *phase*, because the original German sentence contains the word “Bühne” (German word for a concert stage) and not “Phase.” The translations of word senses are obtained from Universal WordNet (UWN) (de Melo and Weikum, 2009).

Generally, we prefer WordNet noun phrases over YAGO3 named entities since noun phrases have lower type ambiguity (fewer possible types). The final score of a sense s is:

$$score(\mathrm{s}) = \alpha freq(\mathrm{s}) + \beta wiki(\mathrm{s}) + \gamma trans(\mathrm{s}) \tag{6.1}$$

where $freq(\mathrm{s})$ is the frequency count of s, and $wiki(\mathrm{s})$ and $trans(\mathrm{s})$ equal maximal frequency count if the Wikipedia prior and Translation prior conditions hold (and otherwise set to 0). α, β, γ are tunable hyper-parameters (set using withheld data, in experiments).

Finally, from the list of candidate noun phrases and named entities, we generate a disambiguated argument: either a WordNet synset or a YAGO3 entity identifier.

6.3.3 Argument Typing

In the third step of relation typing, we assign candidate types to the disambiguated arguments. To this end, we query YAGO3 for semantic types (including transitive hypernyms) for a given YAGO3 or WordNet identifier.

The type system used in POLY consists of a subset of the WordNet noun hierarchy. We restrict ourselves to 734 types, chosen semi-automatically as follows. We selected the 1000 most frequent WordNet types in YAGO3 (including transitive hypernyms). Redundant and non-informative types were filtered out by the following technique: all types were organized into a directed acyclic graph, and we removed a type when the frequency count of some of its children was higher than 80% of the parent’s count. For example, we removed type *trainer* since more than 80% of trainers in YAGO3 are also *coaches*. In addition, we manually removed a few non-informative types (e.g. *expressive style*).

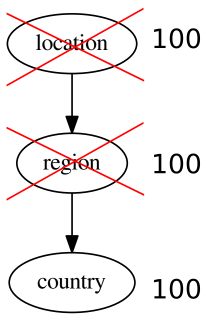
As output, we obtain lists of semantic types for every argument of every relational phrase.

6.3.4 Type Filtering

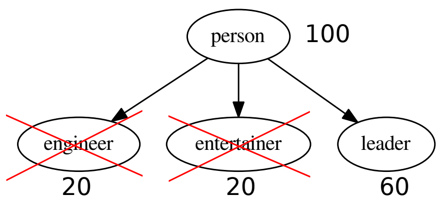
In the last step, we filter types one more time. This time we filter candidate types separately for each distinct relational phrase, in order to choose the most suitable specific type signature for each phrase. This choice is made by type tree pruning.

For each relational phrase, we aggregate all types of the left arguments and all types of the right arguments, summing up their their frequency counts. This information is organized into a directed acyclic graph, based on type hypernymy. Then we prune types as follows (similarly to Section 6.3.3):

- We remove a parent type when the relative frequency count of one of the children types is larger than 80% of the parent's count (Example in Figure 6.3(a)).
- We remove a child type when its relative frequency count is smaller than 20% of the parent's count (Example in Figure 6.3(b)).



((a)) Parent removal



((b)) Children removal

Figure 6.3: Example of type filtering. Numbers represent aggregated frequency counts.

For each of the two arguments of the relational phrase we allow only those types which were left after the pruning. The final output is a set of relational phrases where each has a set of likely type signatures (i.e., pairs of types for the relation's two arguments).

6.4 Relation Clustering

The second stage of the POLY method addresses the relation clustering. The algorithm takes semantically typed relational phrases as input, quantifies the semantic similarity between relational phrases, and organizes them into clusters of synonyms. The key insight that our approach hinges on is that that synonymous phrases have similar translations in a different language. In our setting, two English phrases are semantically similar if they were translated from the same relational phrases in a foreign language. For example, the two phrases “was shot by” and “was filmed by” in Figure 6.1 are both obtained from the same Spanish phrase “fue filmado por.” Moreover, the triples with that Spanish phrase had the same types for the left arguments and right arguments, respectively.

Similarities between English phrases are cast into edge weights of a graph with phrases as nodes. This graph is then partitioned to obtain clusters.

6.4.1 Probabilistic Similarity Measures

The phrase similarities in POLY are based on probabilistic measures. Let us first introduce the notation:

- F: a set of relational phrases from a foreign language F
- E: a set of translations of relational phrases from language F to English
- $c(f, e)$: number of times of translating relational phrase $f \in F$ into relational phrase $e \in E$
- $c(f)$, $c(e)$: frequency counts for relational phrase $f \in F$ and translated relational phrase $e \in E$
- $p(e|f) = \frac{c(f,e)}{c(f)}$: (estimator for the) probability of translating $f \in F$ into $e \in E$
- $p(f|e) = \frac{c(f,e)}{c(e)}$: (estimator for the) probability of $e \in E$ being a translation of $f \in F$

We define

$$p(e_1|e_2) = \sum_f p(e_1|f) * p(f|e_2) \tag{6.2}$$

as the probability of generating relational phrase $e_1 \in E$ from relational phrase $e_2 \in E$.

Finally we define

$$support(e_1, e_2) = \sum_{f \in F} c(f, e_1) * c(f, e_2) \tag{6.3}$$

and

$$confidence(e_1, e_2) = \frac{2}{\frac{1}{p(e_1|e_2)} + \frac{1}{p(e_2|e_1)}} \tag{6.4}$$

Confidence is the final similarity measure used in POLY. We use the harmonic mean in Equation 6.4 to dampen similarity scores that have big differences in their probabilities in Equation 6.2. Typically, pairs e_1, e_2 with such wide gaps in their probabilities come from subsumptions, not synonymous paraphrases.

Cluster of relational phrases
<location> is the heart of <location>
<location> is situated in <location>
<location> is enclosed by <location>
<location> is located amidst <location>
<location> is surrounded by <location>
<location> lies in <location>
<location> is located in <location>
<location> is in <location>
<location> is spread in <location>
<location> is bounded by <location>

Table 6.1: Example of a cluster of relational phrases

Finally, we compute the support and confidence for every pair of English relational phrases which have a common source phrase of translation. We prune phrase pairs with low support (below a threshold), and rank the remaining pairs by confidence.

6.4.2 Graph Clustering

To compute clusters of relational phrases, we use modularity-based graph partitioning. Specifically, we use the partitioning algorithm of Blondel et al. (2008). The resulting clusters (i.e., subgraphs) are then ranked by their weighted graph density multiplied by the graph size (Equation 6.5). The example of a cluster of relational phrases is shown in Figure 6.1.

$$\frac{\sum_{(e_i,e_j)\in E} sim(e_i,e_j)}{|V| * |V-1|} * |V| \tag{6.5}$$

6.5 Evaluation

For the experimental evaluation, we primarily chose triples from the German language (and their English translations). With about 23 million triples, German is the language with the largest number of extractions in the dataset from Faruqui and Kumar (2015), and there are about 2.5 million distinct relational phrases from the German-to-English translation. The POLY method is implemented using Apache Spark, so it scales out to handle such large inputs.

After applying the relation typing algorithm, we obtain around 10 million typed relational phrases. If we ignored the semantic types, we would have about 950,000 distinct phrases. On this input data, POLY detected 1,401,599 pairs of synonyms. The synonymous phrases were organized into 158,725 clusters.

In the following, we present an intrinsic evaluation, an ablation study, and an extrinsic use case. For the intrinsic evaluation, we asked human annotators to judge whether two typed relational phrases are synonymous or not. We

	Precision	**Range**
Top 250	0.91	0.87 – 0.94
Random	0.83	0.78 – 0.87

Table 6.2: Precision of synonym pairs in POLY

also studied source languages other than German. In addition, we compared POLY against PATTY (Nakashole et al., 2012) and DEFIE (Bovi et al., 2015) on the relation paraphrasing task. For the extrinsic evaluation, we considered a simple question answering system and studied to what extent similarities between typed relational phrases can contribute to answering more questions.

6.5.1 Precision of Synonyms

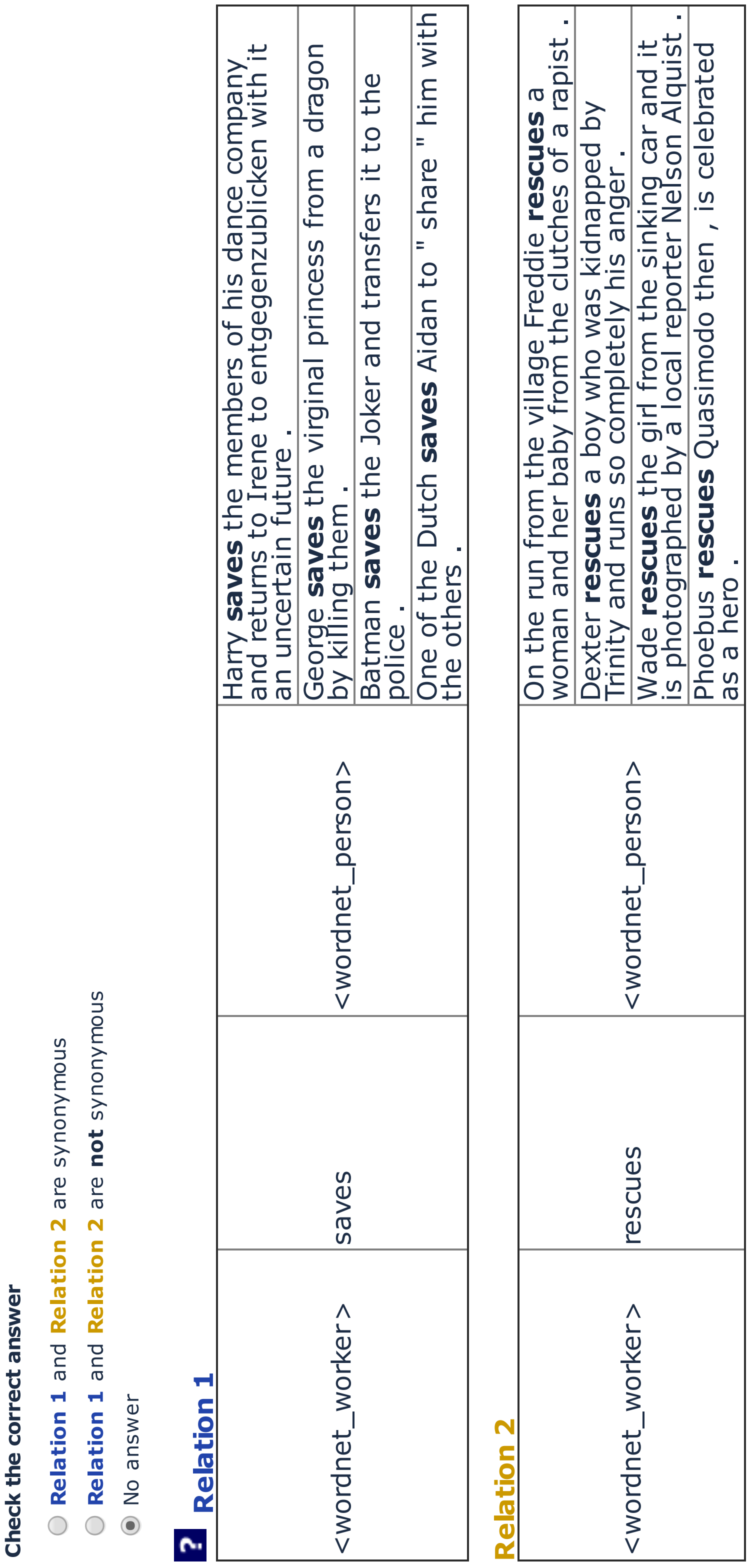
To assess the precision of the discovered synonymy among relational phrases (i.e., clusters of paraphrases), we sampled POLY's output. We assessed the 250 pairs of synonyms with the highest similarity scores. Additionally, we assessed a sample of 250 pairs of synonyms, randomly drawn from POLY's output.

These pairs of synonyms were shown to several human annotators to check their correctness. Relational phrases were presented by showing the semantic types, the textual representation of the relational phrase and sample sentences where the phrase was found. The annotators were asked whether two relational phrases have the same meaning or not. They could also abstain. The example from the evaluation framework is shown in Figure 6.4.

The results of this evaluation are shown in Table 6.2 with (lower bounds and upper bounds of) the 0.95-confidence Wilson score intervals (Brown et al., 2001). This evaluation task had good inter-annotator agreement, with Fleiss' Kappa around 0.6. Table 6.3 shows anecdotal examples of synonymous pairs of relational phrases.

These results show that POLY's quality is comparable with state-of-the-art baselines resources. WiseNet (Moro and Navigli, 2012) is reported to have precision of 0.85 for 30,000 clusters of relational phrases. This is also the only prior work where the precision of synonymy of semantically typed relational phrases was evaluated. The other systems did not report that measure. However, they performed the evaluation of subsumption, entailment or hypernymy relationships which are related to synonymy. Subsumptions in PATTY have precision of 0.83 for top 100 and 0.75 for a random sample. Hypernyms in RELLY are reported to have precision of 0.87 for top 100 and 0.78 for a random sample. DEFIE performed separate evaluations for hypernyms generated directly from WordNet (precision 0.87) and hypernyms obtained through a substring generalization algorithm (precision 0.9).

Typical errors in the paraphrase discovery of POLY come from incorrect translations or extraction errors. For example, "heard" and "belongs to" were clus-



Check the correct answer

- ○ **Relation 1** and **Relation 2** are synonymous
- ○ **Relation 1** and **Relation 2** are **not** synonymous
- ◉ No answer

? Relation 1

<wordnet_worker>	saves	<wordnet_person>	Harry **saves** the members of his dance company and returns to Irene to entgegenzublicken with it an uncertain future .
			George **saves** the virginal princess from a dragon by killing them .
			Batman **saves** the Joker and transfers it to the police .
			One of the Dutch **saves** Aidan to " share " him with the others .

Relation 2

<wordnet_worker>	rescues	<wordnet_person>	On the run from the village Freddie **rescues** a woman and her baby from the clutches of a rapist .
			Dexter **rescues** a boy who was kidnapped by Trinity and runs so completely his anger .
			Wade **rescues** the girl from the sinking car and it is photographed by a local reporter Nelson Alquist .
			Phoebus **rescues** Quasimodo then , is celebrated as a hero .

Figure 6.4: Excerpt from the evaluation survey

Id	Relation phrase	Synonymous Relational Phrase
1	<location> is surrounded by <region>	<location> is the heart of <region>
2	<artifact> is reminiscent of <time_period>	<artifact> recalls <time_period>
3	<painter> was a participant in <show>	<painter> has participated in <show>
4	<group> maintains a partnership with <district>	<group> has partnered with <district>
5	<movie> was shot at <location>	<movie> was filmed in <location>
6	<person> was shot by <group>	<person> was shot dead by <group>
7	<movie> was shot by <film_director>	<movie> was directed by <film_director>
8	<performance> was first performed under <activity>	<performance> was premiered under <activity>
9	<abstraction> were carried out with <machine>	<abstraction> were performed with <machine>
10	<event> was conducted in <town>	<event> was performed in <town>
11	<movie> was awarded at <festival>	<movie> won an award at <festival>
12	<scholar> graduated in <science>	<scholar> holds a degree in <science>
13	<actor> also played in <show>	<actor> also starred in <show>
14	<act> took place at <stadium>	<act> was played in <stadium>
15	<hockey_player> competed at <time_period>	<hockey_player> played in <time_period>
16	<artifact> was hit several times by <weaponry>	<artifact> was repeatedly hit by <weaponry>
17	<location> was exposed to <attack>	<location> was subjected to <attack>
18	<town> is a sports club based in <district>	<town> is a sports club from <district>
19	<creation> was carried out by <botanist>	<creation> was made by <botanist>
20	<district> was conquered by <organization>	<district> was occupied by <organization>

Table 6.3: Examples of synonyms of semantically typed relational phrases

	Precision	Recall	F1
PATTY	0.63	0.32	0.42
DEFIE	0.66	0.32	0.44
POLY	**0.79**	**0.46**	**0.58**

Table 6.4: Comparison to the competitors

tered together because they were translated from the same semantically ambiguous German word "gehört." An example for extraction errors is that "took" and "participated in" were clustered together because "took" was incorrectly extracted from a sentence with the phrase "took part in." Other errors are caused by swapped order of arguments in a triple (i.e., mistakes in detecting passive form) and incorrect argument disambiguation.

6.5.2 Comparison to Competitors

To compare POLY with the closest competitors PATTY and DEFIE, we designed an experiment along the lines of the evaluation of Information Retrieval systems (e.g. TREC benchmarks). First, we randomly chose 100 semantically typed relational phrases with at least three words (to focus on the more interesting multi-word case, rather than single verbs). These relational phrases had to occur in all three resources. For every relational phrase we retrieved synonyms from all of the systems, forming a pool of candidates. Next, to remove minor syntactic variations of the same phrase, the relational phrases were lemmatized. In addition, we removed all leading prepositions, modal verbs, and adverbs.

We manually evaluated the correctness of the remaining paraphrase candidates for each of the 100 phrases. Precision was computed as the ratio of the correct synonyms by one system to the number of all synonyms provided by that system. Recall was computed as the ratio of the number of correct synonyms by one system to the number of all correct synonyms in the candidate pool from all three systems.

The results are presented in Table 6.4. All results are macro-averaged over the 100 sampled phrases. We performed a paired t-test for precision and recall of POLY against each of the systems and obtained p-values below 0.05. POLY and DEFIE offer much higher diversity of synonyms than PATTY. However, DEFIE's synonyms often do not fit the semantic type signature of the given relational phrase and are thus incorrect. For example, *was assumed by* was found to be a synonym of "<*group*> was acquired by <*group*>." PATTY, on the other hand, has higher recall due to its variety of prepositions attached to relational phrases; however, these also include spurious phrases, leading to lower precision. For example, "succeeded in" was found to be a synonym of "<*person*> was succeeded by <*leader*>." Overall, POLY achieves much higher precision and recall than both of these baselines.

	Precision	Coverage
POLY	0.83	1,401,599
− disambiguation	0.66 ± 0.06	1,279,941
Type system 100	0.76 ± 0.05	858,053
Type system 5	0.62 ± 0.06	236,804
Type filtering 0.7	0.81 ± 0.05	192,117
Type filtering 0.9	0.73 ± 0.05	2,061,257

Table 6.5: Ablation Study

6.5.3 Ablation Study

To evaluate the influence of different components, we performed an ablation study. We consider versions of POLY where Wikipedia prior and Translation prior (Section 6.3.2) are disregarded (− *disambiguation*), where the type system (Section 6.3.3) was limited to the 100 most frequent YAGO types (*Type system 100*) or to the 5 top-level types from the YAGO hierarchy (*Type system 5*), or where the type filtering parameter (Section 6.3.4) was set to 70% or 90% (*Type filtering 0.7/0.9*). The evaluation was done on random samples of 250 pairs of synonyms.

Table 6.5 shows the results with the 0.95-confidence Wilson score intervals. Without our argument disambiguation techniques, the precision drops heavily. When weakening the type system, our techniques for argument typing and type filtering are penalized, resulting in lower precision. So we see that all components of the POLY architecture are essential for achieving high-quality output. Lowering the type-filtering threshold yields results with comparable precision. However, increasing the threshold adversely affects the noise filtering.

6.5.4 Evaluation with Other Languages

In addition to evaluating the paraphrases derived from German, we also evaluated the relational phrase synonymy derived from a few other languages with lower numbers of extractions. We chose French, Hindi, and Russian, which were also used in the evaluation of the precision of relational phrases presented by Faruqui and Kumar (2015). The results are presented in Table 6.6, again with the 0.95-confidence Wilson score intervals.

Synonyms derived from French have similar quality as those from German. This is plausible as one would assume that French and German have similar quality in translation to English. Synonyms derived from Russian and Hindi have lower precision due to the lower translation quality. The precision for Hindi is lower, as the Hindi input corpus has much fewer sentences than for the other languages.

	Top 250	Random 250
French	0.93 ± 0.03	0.85 ± 0.04
Hindi	0.86 ± 0.05	0.71 ± 0.05
Russian	0.85 ± 0.05	0.77 ± 0.05

Table 6.6: Precision of synonym pairs for other languages

6.5.5 Extrinsic Evaluation: Question Answering

As an extrinsic use case for the POLY resource, we constructed a simple Question Answering (QA) system over knowledge graphs such as Freebase, and determined the number of questions for which the system can find a correct answer. We followed the approach presented by Fader et al. (2014). The system consists of question parsing, query rewriting and database look-up stages. We disregard the stage of ranking answer candidates, and merely test whether the system could return the right answer at all (i.e., would return it if the ranking were perfect).

In the question parsing stage, we use 10 high-precision parsing operators by Fader et al. (2014), which map questions (e.g., "Who invented papyrus?") to knowledge graph queries (e.g., *(?x, invented, papyrus)*). Additionally, we map question words to semantic types. For example, the word *who* is mapped to *person*, *where* to *location*, *when* to *abstract entity* and the rest of the question words are mapped to type *entity*.

We harness synonyms and hyponyms of relational phrases to paraphrase the predicate of the query. The paraphrases must be compatible with the semantic type of the question word. In the end, we use the original query, as well as found paraphrases, to query a database of subject, predicate, object triples. As the knowledge graph for this experiment we used the union of several collections: a triples database from OpenIE (Banko et al., 2007; Fader et al., 2011), Freebase (Bollacker et al., 2008), Probase (Wu et al., 2012), and NELL (Carlson et al., 2010). In total, this knowledge graph contained more than 900 Million triples.

We compared six systems for paraphrasing semantically typed relational phrases:

- **Basic**: no paraphrasing at all, merely using the originally generated query.
- **DEFIE**: using the taxonomy of relational phrases by Bovi et al. (2015).
- **PATTY**: using the taxonomy of relational phrases by Nakashole et al. (2012).
- **RELLY**: using the subset of the PATTY taxonomy with additional entailment relationships between relational phrases (Grycner et al., 2015).
- **POLY_DE**: using synonyms of relational phrases derived from the German language.

- **POLY_ALL**: using synonyms of relational phrases derived from the 61 available languages.

Since DEFIE's relational phrases are represented by BabelNet (Navigli and Ponzetto, 2012) word sense identifiers, we generated all possible lemmas for each identifier.

We ran the paraphrase-enhanced QA system for three benchmark sets of questions:

- **TREC**: the set of questions used for the evaluation of information retrieval QA systems (Voorhees and Tice, 2000)
- **WikiAnswers**: a randomly sampled set of questions from the WikiAnswers portal (Fader et al., 2013).
- **WebQuestions**: the set of questions about Freebase entities (Berant et al., 2013).

From these question sets, we kept only those questions which can be parsed by one of the 10 question parsing templates and have a correct answer in the gold-standard ground truth. In total, we executed 451 questions for TREC, 516 for WikiAnswers and 1979 for WebQuestions.

For every question, each paraphrasing system generates a set of answers. We measured for how many questions we could obtain at least one correct answer. Table 6.7 shows the results.

The best results were obtained by **POLY_ALL**. We performed a paired t-test for the results of POLY_DE and POLY_ALL against all other systems. The differences between POLY_ALL and the other systems are statistically significant with p-value below 0.05. Additionally, we evaluated paraphrasing systems which consist of combination of all of the described datasets and all of the described datasets without POLY. The difference between these two versions suggest that POLY contains many paraphrases which are available in none of the competing resources.

6.6 Related Work

Knowledge bases (KBs) contribute to many NLP tasks, including Word Sense Disambiguation (Moro et al., 2014), Named Entity Disambiguation (Hoffart et al., 2011), Question Answering (Fader et al., 2014), Coreference Resolution (Rahman and Ng, 2011), and Textual Entailment (Sha et al., 2015). Widely used KBs are DBpedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2008), YAGO (Mahdisoltani et al., 2015), Wikidata (Vrandečić and Krötzsch, 2014), and the Google Knowledge Vault (Dong et al., 2014). KBs have rich information about named entities, but are pretty sparse on relations. In the

	TREC	WikiAnswers	WebQuestions
Basic	193	144	365
DEFIE	197	147	394
RELLY	208	150	424
PATTY	213	155	475
POLY_DE	232	163	477
POLY_ALL	**238**	**173**	**530**
All	246	176	562
All / POLY	218	157	494
Questions	451	516	1979

Table 6.7: Number of questions with correct answer by each system for three question sets.

latter regard, manually created resources such as WordNet (Fellbaum, 1998), VerbNet (Kipper et al., 2008) or FrameNet (Baker et al., 1998) are much richer, but still face the limitation of labor-intensive input and human curation.

The paradigm of OpenIE was developed to overcome the weak coverage of relations in automatically constructed KBs. OpenIE methods process natural language texts to produce triples of surface forms for the arguments and relational phrase of binary relations. The first large-scale approach along these lines, TextRunner (Banko et al., 2007), was later improved by ReVerb (Fader et al., 2011) and OLLIE (Mausam et al., 2012). The focus of these methods has been on verbal phrases as relations, and there is little effort to determine lexical synonymy among these phrases.

The first notable effort to build up a resource for relational paraphrases is DIRT (Lin and Pantel, 2001a), based on Harris' Distributional Hypothesis to cluster syntactic patterns. RESOLVER (Yates and Etzioni, 2009) introduced a probabilistic relational model for predicting synonymy. Yao et al. (2012) incorporated latent topic models to resolve the ambiguity of relational phrases. Other probabilistic approaches employed matrix factorization for finding entailments between relations (Riedel et al., 2013; Petroni et al., 2015) or used probabilistic graphical models to find clusters of relations (Grycner et al., 2014). All of these approaches rely on the co-occurrence of the arguments of the relation.

Recent endeavors to construct large repositories of relational paraphrases are PATTY, WiseNet and DEFIE. PATTY (Nakashole et al., 2012) devised a sequence mining algorithm to extract relational phrases with semantic type signatures, and organized them into synonymy sets and hypernymy hierarchies. WiseNet (Moro and Navigli, 2012) tapped Wikipedia categories for a similar way of organizing relational paraphrases. DEFIE (Bovi et al., 2015) went even further and used word sense disambiguation, anchored in WordNet, to group phrases with the same meanings.

Translation models have previously been used for paraphrase detection. Barzilay and McKeown (2001) utilized multiple English translations of the same source text for paraphrase extraction. Bannard and Callison-Burch (2005) used the bilingual pivoting method on parallel corpora for the same task. Similar methods were performed at a much bigger scale by the Paraphrase Database (PPDB) project (Ganitkevitch et al., 2013; Pavlick et al., 2015). Unlike POLY, the focus of these projects was not on paraphrases of binary relations. Moreover, POLY considers the semantic type signatures of relations, which is missing in PPDB.

Research on OpenIE for languages other than English has received little attention. Kim et al. (2011) uses Korean-English parallel corpora for cross-lingual projection. Gamallo et al. (2012) developed an OpenIE system for Spanish and Portuguese using rules over shallow dependency parsing. The recent work of Faruqui and Kumar (2015) extracted relational phrases from Wikipedia in 61 languages using cross-lingual projection. Lewis and Steedman (2013) computed clusters of semantically equivalent English and French phrases, based on the arguments of relations.

6.7 Conclusions

We presented POLY, a method for clustering semantically typed English relational phrases using a multilingual corpus. We used POLY to create a repository with both high coverage and precision, as shown in our evaluation The synonyms of relational phrases could potentially be useful for many problems of natural language processing. As a use case, we showed that POLY can enhance question answering. The future work could include jointly harnessing all 61 languages in the corpus, rather than considering them pairwise, and aiming for a paraphrase resource in all languages. The POLY resource is publicly available at `www.mpi-inf.mpg.de/yago-naga/poly/`.

Chapter 7

Conclusion

7.1 Thesis Contribution

This dissertation addresses the problem of automatic construction of lexicons of relational phrases. Throughout this thesis, we analyzed automatic ways of finding counterparts of relational phrases in other taxonomies, automatic construction of a graph of relational phrases, methods for finding synonyms among them, and an approach for detecting fine-grained types of arguments of textual representations of relations. Information included in the lexicons, such as synonymy and hypernymy, could help in Question Answering (QA), Textual Entailment, Named Entity Disambiguation and extending Knowledge Bases (KBs).

The first contribution of this dissertation is HARPY (Grycner and Weikum, 2014). In HARPY, we investigated a graph-based method for aligning relational phrases from PATTY (Nakashole et al., 2012) with WordNet verb senses. The algorithm is based on SimRank applied to a graph consisting of phrases, verbs, and their shared features. The evaluation shows the high quality of the alignment links compared to the set of baselines and presents the potential in the extrinsic task, that is detecting fine-grained lexical types for the arguments of verb senses in WordNet. Obtaining the alignment links is the first step in constructing a hierarchy of relational phrases through WordNet hierarchy transfer.

The second contribution of this dissertation is RELLY (Grycner et al., 2015). The RELLY algorithm is a highly-scalable method for the construction of a large hypernymy graph of relational phrases. The collective probabilistic programming framework – Probabilistic Soft Logic (PSL) – integrates resources like PATTY, YAGO, and HARPY. Additionally, it allows us to impose the local and global constraints on the final structure of the graph. The evaluation shows the high precision of the hypernymy links in the output graph and a positive influence in the downstream application of comparing anonymized documents. Moreover, we analyzed the potential of PSL in the relation clustering task.

The third contribution of this dissertation is POLY (Grycner and Weikum, 2016). Our method, using a parallel corpus of sentences parsed into triples of phrases, clusters relational phrases using probabilistic measures. Moreover, multilingual information included in the corpus is used to detect fine-grained types of arguments of relational phrases. We compared POLY against other resources of relational paraphrases in the intrinsic evaluations and in the downstream application task - a simplified QA problem. The evaluation showed improvements over the competing resources.

7.2 Outlook and Future Work

Here, we discuss the limitations of the presented methods and various research opportunities for future work.

In Chapter 3 and Chapter 4, for the alignment and graph construction, we considered only relational phrases available in the PATTY resource. The alignment and graph construction for other resources (e.g., WiseNet, DefIE, POLY, or PPDB) could produce new interesting resources. Moreover, considering all resources jointly could improve the quality of the final graph of relational phrases.

Most of the relational phrases used in the projects in this thesis are built around verb phrases. The natural step for an extension would include considering verb phrases and noun phrases jointly within one graph of relations. Furthermore, an improvement of all methods could come from using relational phrases of higher-arity or higher-order. Throughout the thesis, we used and produced only relational phrases which represent first-order binary relations. However, the ultimate graph of relational phrases should also contain information about higher-arity and higher-order of relational phrases.

The focus of the thesis was on finding the synonymy (in the paraphrase clustering) and hypernymy relationships (in the graph construction) between relational phrases. A comprehensive KB of relations should also contain other relationships. Interesting examples of such relationships are antonymy, enablement, or causality.

Finally, the graph and the clusters of relational paraphrases contain relations expressed only in the English language. Finding relational paraphrases for languages other than English would bring added value. Similarly, a graph of foreign relational phrases would help to solve Natural Language Understanding (NLU) problems in under-resourced languages. Moreover, considering jointly multiple languages within one lexicon of relational paraphrases enables new application opportunities.

Bibliography

[Agichtein and Gravano2000] Eugene Agichtein and Luis Gravano. 2000. *Snowball*: extracting relations from large plain-text collections. In *ACM DL*, pages 85–94.

[Aho et al.1972] Alfred V. Aho, M. R. Garey, and Jeffrey D. Ullman. 1972. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1(2):131–137.

[Alfonseca et al.2013] Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. HEADY: news headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1243–1253. The Association for Computer Linguistics.

[Androutsopoulos and Malakasiotis2010] Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *J. Artif. Intell. Res. (JAIR)*, 38:135–187.

[Angeli et al.2015] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics.

[Anke et al.2016] Luis Espinosa Anke, José Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 424–435. The Association for Computational Linguistics.

[Antonellis et al.2008] Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2008. Simrank++: query rewriting through link analysis of the click graph. *PVLDB*, 1(1):408–421.

[Auer et al.2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

[Bach et al.2013] Stephen H. Bach, Bert Huang, Ben London, and Lise Getoor. 2013. Hinge-loss Markov random fields: Convex inference for structured prediction. In *Uncertainty in Artificial Intelligence (UAI).*

[Bach et al.2015] Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2015. Hinge-loss Markov Random Fields and probabilistic soft logic. *CoRR*, abs/1505.04406.

[Bai et al.2010] Rujiang Bai, Xiaoyue Wang, and Junhua Liao. 2010. Extract semantic information from wordnet to improve text classification performance. In Tai-Hoon Kim and Hojjat Adeli, editors, *Advances in Computer Science and Information Technology, AST/UCMA/ISA/ACN 2010 Conferences, Miyazaki, Japan, June 23-25, 2010. Joint Proceedings*, volume 6059 of *Lecture Notes in Computer Science*, pages 409–420. Springer.

[Baker et al.1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 86–90. Morgan Kaufmann Publishers / ACL.

[Banerjee and Pedersen2003] Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 805–810. Morgan Kaufmann.

[Banko et al.2007] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.

[Bannard and Callison-Burch2005] Colin J. Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. The Association for Computer Linguistics.

[Bansal et al.2014] Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1041–1051. The Association for Computer Linguistics.

[Bar-Haim et al.2007] Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 871–876. AAAI Press.

[Bar-Haim et al.2009] Roy Bar-Haim, Jonathan Berant, and Ido Dagan. 2009. A compact forest for scalable inference over entailment and paraphrase rules. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1056–1065. ACL.

[Barzilay and McKeown2001] Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France.*, pages 50–57. Morgan Kaufmann Publishers.

[Bast and Haussmann2013] Hannah Bast and Elmar Haussmann. 2013. Open information extraction via contextual sentence decomposition. In *2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, September 16-18, 2013*, pages 154–159. IEEE Computer Society.

[Bast and Haussmann2014] Hannah Bast and Elmar Haussmann. 2014. More informative open information extraction via simple inference. In Maarten de Rijke, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann, editors, *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, volume 8416 of *Lecture Notes in Computer Science*, pages 585–590. Springer.

[Bentivogli et al.2004] Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the wordnet domains hierarchy: Semantics, coverage and balancing. In *Proceedings of the Workshop on Mul-*

tilingual Linguistic Ressources, MLR '04, pages 101–108, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Berant and Liang2014] Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1415–1425. The Association for Computer Linguistics.

[Berant et al.2010] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1220–1229. The Association for Computer Linguistics.

[Berant et al.2011] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 610–619. The Association for Computer Linguistics.

[Berant et al.2012a] Jonathan Berant, Ido Dagan, Meni Adler, and Jacob Goldberger. 2012a. Efficient tree-based approximation for entailment graph learning. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 117–125. The Association for Computer Linguistics.

[Berant et al.2012b] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012b. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

[Berant et al.2013] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. The Association for Computational Linguistics.

[Berant et al.2015] Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):249–291.

[Bhutani et al.2016] Nikita Bhutani, H. V. Jagadish, and Dragomir R. Radev. 2016. Nested propositions in open information extraction. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 55–64. The Association for Computational Linguistics.

[Bingel and Søgaard2016] Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

[Blondel et al.2008] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

[Bollacker et al.2008] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM.

[Bordes et al.2014] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I*, volume 8724 of *Lecture Notes in Computer Science*, pages 165–180. Springer.

[Bovi et al.2015] Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL*, 3:529–543.

[Brin1998] Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto O. Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases, International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer.

[Broecheler and Getoor2009] Matthias Broecheler and Lise Getoor. 2009. Probabilistic similarity logic. In *International Workshop on Statistical Relational Learning*.

[Brown et al.2001] Lawrence D. Brown, T. Tony Cai, and Anirban Dasgupta. 2001. Interval estimation for a binomial proportion. *Statistical Science*, 16:101–133.

[Budanitsky and Hirst2006] Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

[Bunescu and Mooney2007] Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

[Cafarella et al.2006] Michael J. Cafarella, Oren Etzioni, and Dan Suciu. 2006. Structured queries over web text. *IEEE Data Eng. Bull.*, 29(4):45–51.

[Carlson et al.2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press.

[Chklovski and Pantel2004] Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 33–40. The Association for Computational Linguistics.

[Corro and Gemulla2013] Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: clause-based open information extraction. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo A. Baeza-Yates, and Sue B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366. International World Wide Web Conferences Steering Committee / ACM.

[Dagan et al.2005] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc, editors, *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

[Dagan et al.2010] Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches - erratum. *Natural Language Engineering*, 16(1):105.

[Dagan et al.2013] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

[de Melo and Weikum2009] Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 513–522. ACM.

[de Sá Mesquita et al.2013] Filipe de Sá Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 447–457. The Association for Computational Linguistics.

[de Sá Mesquita2012] Filipe de Sá Mesquita. 2012. Clustering techniques for open relation extraction. In Xin Luna Dong and M. Tamer Özsu, editors, *Proceedings of the ACM SIGMOD/PODS PhD Symposium 2012, Scottsdale, AZ, USA, May 20, 2012*, pages 27–32. ACM.

[Dinu and Wang2009a] Georgiana Dinu and Rui Wang. 2009a. Inference rules and their application to recognizing textual entailment. In Alex Lascarides, Claire Gardent, and Joakim Nivre, editors, *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 211–219. The Association for Computer Linguistics.

[Dinu and Wang2009b] Georgiana Dinu and Rui Wang. 2009b. Inference rules for recognizing textual entailment. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 90–103, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Dong et al.2014] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610. ACM.

[Etzioni et al.2005] Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134.

[Fader et al.2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1535–1545. The Association for Computational Linguistics.

[Fader et al.2013] Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1608–1618. The Association for Computer Linguistics.

[Fader et al.2014] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 1156–1165. ACM.

[Falke et al.2016] Tobias Falke, Gabriel Stanovsky, Iryna Gurevych, and Ido Dagan. 2016. Porting an open information extraction system from english to german. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 892–898. The Association for Computational Linguistics.

[Faruqui and Kumar2015] Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1351–1356. The Association for Computational Linguistics.

[Fellbaum1998] Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

[Ferrández et al.2010] Óscar Ferrández, Michael Ellsworth, Rafael Muñoz, and Collin F. Baker. 2010. Aligning framenet and wordnet based on semantic neighborhoods. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

[Fogaras and Rácz2005] Dániel Fogaras and Balázs Rácz. 2005. Scaling link-based similarity search. In Allan Ellis and Tatsuya Hagino, editors, *Proceed-*

ings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, pages 641–650. ACM.

[Gabrilovich and Markovitch2007] Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1606–1611.

[Gamallo and García2015] Pablo Gamallo and Marcos García. 2015. Multilingual open information extraction. In Francisco C. Pereira, Penousal Machado, Ernesto Costa, and Amílcar Cardoso, editors, *Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings*, volume 9273 of *Lecture Notes in Computer Science*, pages 711–722. Springer.

[Gamallo et al.2012] Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. 2012. Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, ROBUS-UNSUP '12, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ganitkevitch et al.2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: the paraphrase database. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 758–764. The Association for Computational Linguistics.

[Gerber and Ngomo2012] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. 2012. Extracting multilingual natural-language patterns for RDF predicates. In Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Aquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, volume 7603 of *Lecture Notes in Computer Science*, pages 87–96. Springer.

[Grau et al.2013] Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn dos Santos, and Ondrej Zamazal. 2013. Results of the ontology alignment evaluation initiative 2013. In Pavel Shvaiko, Jérôme Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jiménez-Ruiz, editors, *Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, Oc-*

tober 21, 2013., volume 1111 of *CEUR Workshop Proceedings*, pages 61–100. CEUR-WS.org.

[Grycner and Weikum2014] Adam Grycner and Gerhard Weikum. 2014. HARPY: hypernyms and alignment of relational paraphrases. In Jan Hajic and Junichi Tsujii, editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2195–2204. The Association for Computational Linguistics.

[Grycner and Weikum2016] Adam Grycner and Gerhard Weikum. 2016. POLY: mining relational paraphrases from multilingual sentences. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2183–2192. The Association for Computational Linguistics.

[Grycner et al.2014] Adam Grycner, Gerhard Weikum, Jay Pujara, James Foulds, and Lise Getoor. 2014. A unified probabilistic approach for semantic clustering of relational phrases. In *AKBC '14: Proceedings of the 2014 Workshop on Automated Knowledge Base Construction*.

[Grycner et al.2015] Adam Grycner, Gerhard Weikum, Jay Pujara, James R. Foulds, and Lise Getoor. 2015. RELLY: inferring hypernym relationships between relational phrases. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 971–981. The Association for Computational Linguistics.

[Gupta et al.2014] Rahul Gupta, Alon Y. Halevy, Xuezhi Wang, Steven Euijong Whang, and Fei Wu. 2014. Biperpedia: An ontology for search applications. *PVLDB*, 7(7):505–516.

[Harabagiu et al.2003] Sanda M. Harabagiu, Steven J. Maiorano, and Marius Pasca. 2003. Open-domain textual question answering techniques. *Natural Language Engineering*, 9(3):231–267.

[Harris1954] Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

[Hasegawa et al.2004] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In Donia Scott, Walter Daelemans, and Marilyn A. Walker, editors, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*, pages 415–422. The Association for Computational Linguistics.

[Hashimoto et al.2009] Chikara Hashimoto, Kentaro Torisawa, Kow Kuroda, Stijn De Saeger, Masaki Murata, and Jun'ichi Kazama. 2009. Large-scale verb entailment acquisition from the web. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1172–1181. ACL.

[Haveliwala2002] Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In David Lassner, Dave De Roure, and Arun Iyengar, editors, *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii*, pages 517–526. ACM.

[Hearst1992] Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992*, pages 539–545.

[Hendrickx et al.2010] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Katrin Erk and Carlo Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 33–38. The Association for Computer Linguistics.

[Hoffart et al.2011] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792. The Association for Computational Linguistics.

[Hoffart et al.2013] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.

[Hoffmann et al.2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 541–550. The Association for Computer Linguistics.

[Hopcroft and Tarjan1973] John E. Hopcroft and Robert Endre Tarjan. 1973. Efficient algorithms for graph manipulation [H] (algorithm 447). *Commun. ACM*, 16(6):372–378.

[Huang et al.2012a] Bert Huang, Stephen H. Bach, Eric Norris, Jay Pujara, and Lise Getoor. 2012a. Social group modeling with probabilistic soft logic. In *NIPS Workshop on Social Network and Social Media Analysis: Methods, Models, and Applications*.

[Huang et al.2012b] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. 2012b. Probabilistic soft logic for trust analysis in social networks. In *International Workshop on Statistical Relational Artificial Intelligence (StaRAI 2012)*.

[Jeh and Widom2002] Glen Jeh and Jennifer Widom. 2002. Simrank: a measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 538–543. ACM.

[Kim et al.2011] Seokhwan Kim, Minwoo Jeong, Jonghoon Lee, and Gary Geunbae Lee. 2011. A cross-lingual annotation projection-based self-supervision approach for open information extraction. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 741–748. The Association for Computer Linguistics.

[Kimmig et al.2012] Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*.

[Kipper et al.2008] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

[Kozareva and Hovy2010] Zornitsa Kozareva and Eduard H. Hovy. 2010. Learning arguments and supertypes of semantic relations using recursive patterns. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1482–1491. The Association for Computer Linguistics.

[Lehmann et al.2015] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[Lenat1995] Douglas B. Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):32–38.

[Levy et al.2014] Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for open IE propositions. In Roser Morante and Wen-tau

Yih, editors, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 87–97. The Association for Computational Linguistics.

[Lewis and Steedman2013] Mike Lewis and Mark Steedman. 2013. Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 681–692. The Association for Computational Linguistics.

[Lin and Pantel2001a] Dekang Lin and Patrick Pantel. 2001a. DIRT @sbt@discovery of inference rules from text. In Doheon Lee, Mario Schkolnick, Foster J. Provost, and Ramakrishnan Srikant, editors, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 323–328. ACM.

[Lin and Pantel2001b] Dekang Lin and Patrick Pantel. 2001b. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

[Lin et al.2012] Thomas Lin, Mausam, and Oren Etzioni. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 84–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Madnani et al.2012] Nitin Madnani, Joel R. Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 182–190. The Association for Computational Linguistics.

[Mahdisoltani et al.2015] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org.

[Manning et al.2008a] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, 2008a. *Introduction to information retrieval*, chapter 6. Scoring, Term Weighting, and the Vector Space Model. Cambridge University Press.

[Manning et al.2008b] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, 2008b. *Introduction to information retrieval*, chapter 16.3 Evaluation of clustering. Cambridge University Press.

[Manning et al.2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

[Marsi et al.2007] Erwin Marsi, Emiel Krahmer, and Wauter Bosma. 2007. Dependency-based paraphrasing for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 83–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Marton et al.2009] Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 381–390. ACL.

[Mausam et al.2012] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 523–534. The Association for Computational Linguistics.

[McCarthy et al.2004] Diana McCarthy, Rob Koeling, Julie Weeds, and John A. Carroll. 2004. Finding predominant word senses in untagged text. In Donia Scott, Walter Daelemans, and Marilyn A. Walker, editors, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*, pages 279–286. The Association for Computational Linguistics.

[Mehdad et al.2013] Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Shafiq R. Joty. 2013. Towards topic labeling with phrase entailment and aggregation. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 179–189. The Association for Computational Linguistics.

[Meyer and Gurevych2012] Christian M. Meyer and Iryna Gurevych. 2012. To exhibit is not to loiter: A multilingual, sense-disambiguated wiktionary for measuring verb similarity. In Martin Kay and Christian Boitet, editors, *COLING 2012, 24th International Conference on Computational Linguis-*

tics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India, pages 1763–1780. Indian Institute of Technology Bombay.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Min et al.2013] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 777–782. The Association for Computational Linguistics.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, and Janyce Wiebe, editors, *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011. The Association for Computer Linguistics.

[Mitchell et al.2015] Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2302–2310. AAAI Press.

[Mohamed et al.2011] Thahir Mohamed, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2011. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1447–1455. The Association for Computational Linguistics.

[Moro and Navigli2012] Andrea Moro and Roberto Navigli. 2012. WiSeNet: building a wikipedia-based semantic network with ontologized relations. In Xue-wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1672–1676. ACM.

[Moro and Navigli2013] Andrea Moro and Roberto Navigli. 2013. Integrating syntactic and semantic analysis into the open information extraction paradigm. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2148–2154. IJCAI/AAAI.

[Moro et al.2014] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *TACL*, 2:231–244.

[Nakashole et al.2011] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. 2011. Scalable knowledge harvesting with high precision and high recall. In Irwin King, Wolfgang Nejdl, and Hang Li, editors, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 227–236. ACM.

[Nakashole et al.2012] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1135–1145. The Association for Computational Linguistics.

[Nakashole et al.2013] Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1488–1497. The Association for Computer Linguistics.

[Nakov2008a] Preslav Nakov. 2008a. Improved statistical machine translation using monolingual paraphrases. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, editors, *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 338–342. IOS Press.

[Nakov2008b] Preslav Nakov. 2008b. Improving english-spanish statistical machine translation: Experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 147–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Navigli and Lapata2010] Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):678–692.

[Navigli and Ponzetto2010] Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 216–225. The Association for Computer Linguistics.

[Navigli and Ponzetto2012] Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

[Navigli and Velardi2010] Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1318–1327. The Association for Computer Linguistics.

[Navigli et al.2011] Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1872–1877. IJCAI/AAAI.

[Navigli2009] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

[Palmer2009] Martha. Palmer. 2009. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon ConferenceGenLex-09*, Pisa, Italy, Sept.

[Pavlick et al.2015] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 425–430. The Association for Computer Linguistics.

[Petroni et al.2015] Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. 2015. CORE: context-aware open relation extraction with factorization machines. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1763–1773. The Association for Computational Linguistics.

[Pighin et al.2014] Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 892–901. The Association for Computer Linguistics.

[Pilehvar et al.2013] Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1341–1351. The Association for Computer Linguistics.

[Ponzetto and Navigli2010] Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1522–1531. The Association for Computer Linguistics.

[Ponzetto and Strube2011] Simone Paolo Ponzetto and Michael Strube. 2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artif. Intell.*, 175(9-10):1737–1756.

[Pujara et al.2013] Jay Pujara, Hui Miao, Lise Getoor, and William W. Cohen. 2013. Knowledge graph identification. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors, *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 542–557. Springer.

[Rahman and Ng2011] Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 814–824. The Association for Computer Linguistics.

[Ravichandran and Hovy2002] Deepak Ravichandran and Eduard H. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 41–47. The Association for Computational Linguistics.

[Riedel et al.2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In José L.

Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 74–84. The Association for Computational Linguistics.

[Schmidek and Barbosa2014] Jordan Schmidek and Denilson Barbosa. 2014. Improving open relation extraction via sentence re-structuring. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 3720–3723. European Language Resources Association (ELRA).

[Schoenmackers et al.2010] Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel S. Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1088–1098. The Association for Computational Linguistics.

[Seitner et al.2016] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A large database of hypernymy relations extracted from the web. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* European Language Resources Association (ELRA).

[Sha et al.2015] Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui, and Tingsong Jiang. 2015. Recognizing textual entailment using probabilistic inference. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1620–1625. The Association for Computational Linguistics.

[Shin et al.2015] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. 2015. Incremental knowledge base construction using DeepDive. *PVLDB*, 8(11):1310–1321.

[Shinyama and Sekine2006] Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

[Snow et al.2004] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1297–1304.

[Snow et al.2006] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

[Speer and Havasi2012] Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3679–3686. European Language Resources Association (ELRA).

[Suchanek et al.2006] Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 712–717. ACM.

[Suchanek et al.2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM.

[Suchanek et al.2009] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. 2009. SOFIE: a self-organizing framework for information extraction. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 631–640. ACM.

[Surdeanu et al.2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 455–465. The Association for Computational Linguistics.

[Takamatsu et al.2011] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2011. Probabilistic matrix factorization leveraging contexts for unsupervised relation extraction. In Joshua Zhexue Huang, Longbing Cao, and Jaideep Srivastava, editors, *Advances in Knowledge Discovery and Data Mining - 15th Pacific-Asia Conference, PAKDD 2011, Shenzhen, China, May 24-27, 2011, Proceedings, Part I*, volume 6634 of *Lecture Notes in Computer Science*, pages 87–99. Springer.

[Talukdar and Crammer2009] Partha Pratim Talukdar and Koby Crammer. 2009. New regularized algorithms for transductive learning. In Wray L. Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II*, volume 5782 of *Lecture Notes in Computer Science*, pages 442–457. Springer.

[Tylenda et al.2014a] Tomasz Tylenda, Sarath Kumar Kondreddi, and Gerhard Weikum. 2014a. Spotting knowledge base facts in web texts. In *AKBC 2014, 4th Workshop on Automated Knowledge Base Construction*, Montreal, Canada. AKBC Board.

[Tylenda et al.2014b] Tomasz Tylenda, Yafang Wang, and Gerhard Weikum. 2014b. Spotting facts in the wild. In *Workshop on Automatic Creation and Curation of Knowledge Bases (WACCK)*, Snowbird, UT, USA.

[Velardi et al.2013] Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

[Verga et al.2016] Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 886–896. The Association for Computational Linguistics.

[Voorhees and Tice2000] Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR*, pages 200–207.

[Vrandečić and Krötzsch2014] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

[Wang et al.2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1591–1601. The Association for Computational Linguistics.

[Weeds and Weir2003] Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 81–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Weston et al.2013] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1366–1371. The Association for Computational Linguistics.

[Wu and Weld2010] Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 118–127. The Association for Computer Linguistics.

[Wu et al.2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 481–492. ACM.

[Xu et al.2013] Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of* *the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 868–877. The Association for Computational Linguistics.

[Xu et al.2015] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1785–1794. The Association for Computational Linguistics.

[Xu et al.2016] Ying Xu, Pascual Martínez-Gómez, Yusuke Miyao, and Randy Goebel. 2016. Paraphrase for open question answering: New dataset and methods. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 53–61, San Diego, California, June. Association for Computational Linguistics.

[Yahya et al.2012] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. 2012. Natural language questions for the web of data. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 379–390. The Association for Computational Linguistics.

[Yahya et al.2013] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. 2013. Robust question answering over the web of linked data. In Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi, editors, *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 1107–1116. ACM.

[Yao et al.2010] Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1013–1023. The Association for Computational Linguistics.

[Yao et al.2011] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1456–1466. ACL.

[Yao et al.2012] Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 712–720. The Association for Computer Linguistics.

[Yates and Etzioni2009] Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Intell. Res. (JAIR)*, 34:255–296.

[Zeng et al.2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In Jan Hajic and Junichi Tsujii, editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344. The Association for Computational Linguistics.

[Zeng et al.2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1753–1762. The Association for Computational Linguistics.

[Zhao and Grishman2005] Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 419–426. The Association for Computer Linguistics.

[Zhila and Gelbukh2014] Alisa Zhila and Alexander Gelbukh. 2014. Comparison of open information extraction for english and spanish. In *Proceedings of the 19th International Computational Linguistics Conference Dialogue*, Dialogue.

List of Figures

List of Tables

Appendix A

Acronyms

CAG	Candidate Alignment Graph
DAG	Directed Acyclic Graph
HAC	Hierarchical Agglomerative Clustering
IE	Information Extraction
ILP	Integer Linear Programming
KB	Knowledge Base
KBC	Knowledge Base Construction
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
NLU	Natural Language Understanding
OpenIE	Open Information Extraction
PSL	Probabilistic Soft Logic
QA	Question Answering
SVM	Support Vector Machines
WSD	Word Sense Disambiguation