RELLY: Inferring Hypernym Relationships Between Relational Phrases

Adam Grycner, Gerhard Weikum
Max-Planck Institute for Informatics
Campus E1.4, 66123
Saarbrücken, Germany
{agrycner, weikum}
@mpi-inf.mpg.de

Jay Pujara, James Foulds, Lise Getoor
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064
{jpujara,jfoulds,getoor}
@soe.ucsc.edu

Abstract

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 20K relational phrases and 35K hypernymy links. Our evaluation indicates a hypernymy link precision of 78%, and demonstrates the value of this resource for a document-relevance ranking task.

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 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, 2001), 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 600K 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 paper, 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.,

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 971–981,
Lisbon, Portugal, 17-21 September 2015.

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 100K 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 20K relational phrases and 35K 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).

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.

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 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, 2001; 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 paper 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).

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

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

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 ground 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.

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 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 3.1 and Table 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 $hyponym(R1, R2)$.

3.1 PSL Rules

The PSL rules that define the model are shown in Table 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

Table 1: PSL predicates;
$R1, R2$ are relational phrases, $Vb1, Vb2$ WordNet verb senses and $TL1, TR1, T1, T2$ YAGO types

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 $R2$

$R2$ 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 evidence that $R1$ is more specific than $R2$. An example of using HARPY alignment links and WordNet hierarchy is shown in Figure 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.

3.2 RELLY Overview

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

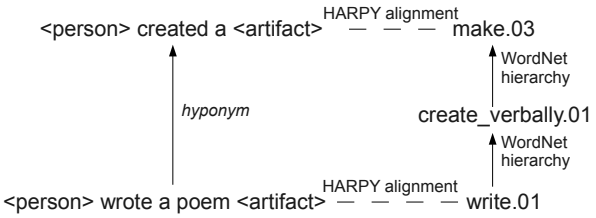

Figure 1: HARPY alignment usage

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 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

Table 2: PSL rules

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	$hyponym(R1, R2) \wedge hyponym(R2, R3) \Rightarrow \neg hyponym(R3, R1)$

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 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.

4.1 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.

4.2 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 8K links from PATTY, 600K links from HARPY and 35K 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 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. 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 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 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

Table 3: Intrinsic evaluation

	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

(e.g. Figure 2) compared with PATTY where "<*organization*> merged <*organization*>" can lead to "<*team*> beat <*team*>".

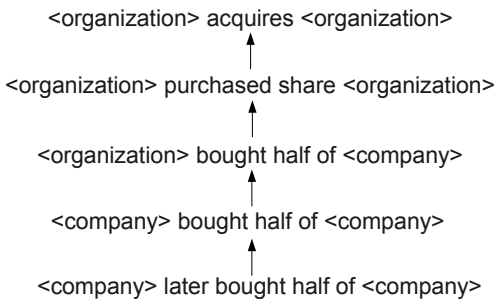

Figure 2: Chain of hypernymy

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

Table 4: Example RELLY hypernymy links

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

Table 5: Results for Entailment graphs induction

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

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 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 in (Berant et al., 2011). This 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 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., 2012), which did not utilize graph decomposition method, had run-time above 5000 seconds.

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, $\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i}$. 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 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*.

Table 6: Extrinsic evaluation (Bond & Marvel)

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

The results of the experiment are presented in Table 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.

5 Related Work

The biggest sources of hypernyms, subsumptions, and hierarchical structure can be found in existing knowledge bases. Examples of these are Freebase (Bollacker et al., 2008), YAGO, DBPedia (Lehmann et al., 2014), 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.

Open Information Extraction 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 Integer Linear Programing 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 in (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).

6 Conclusion

This paper 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. As a future work, RELLY can 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/`.

Acknowledgments: This work was partially supported by National Science Foundation (NSF) grant IIS1218488 and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC00337. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, IARPA, DoI/NBC, or the U.S. Government.

References

Stephen H. Bach, Matthias Broecheler, Ben. Huang, and Lise Getoor. 2015. Hinge-loss Markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG].

Rujiang Bai, Xiaoyue Wang, and Junhua Liao. 2010. Extract semantic information from wordnet to improve text classification performance. In *Advances in Computer Science and Information Technology*, pages 409–420.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of Association for Computational Linguistics and Conference on Computational Linguistics (COLING/ACL)*, pages 86–90.

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of Association for Computational Linguistics (ACL)*, pages 1041–1051.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of Association for Computational Linguistics (ACL)*, pages 610–619.

Jonathan Berant, Ido Dagan, Meni Adler, and Jacob Goldberger. 2012. Efficient tree-based approximation for entailment graph learning. In *Proceedings of Association for Computational Linguistics (ACL)*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

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

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

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 33–40.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches erratum. *Natural Language Engineering*, 16:105–105, 1.

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 *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT)*, pages 758–764.

Adam Grycner and Gerhard Weikum. 2014. HARPY: Hypernyms and alignment of relational paraphrases. In *Proceedings of Conference on Computational Linguistics (COLING)*, pages 2195–2204.

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*.

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)*.

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.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. 2014. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for Open IE propositions. In *Proceedings of Conference on Computational Natural Language Learning (CoNLL)*, pages 87–97.

Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@discovery of inference rules from text. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT)*, pages 182–190.

Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Shafiq Joty. 2013. Towards topic labeling with phrase entailment and aggregation. In *Proceedings of North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT)*, pages 179–189.

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

Andrea Moro and Roberto Navigli. 2012. WiseNet: building a wikipedia-based semantic network with ontologized relations. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1672–1676.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning EMNLP-CoNLL*, pages 1135–1145.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *International Semantic Web Conference (ISWC)*.

Rion Snow, Daniel Jurafsky, and Y. Andrew Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of Association for Computational Linguistics and International Conference on Computational Linguistics (COLING/ACL)*, pages 801–808.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of International Conference on World Wide Web (WWW)*, pages 697–706.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 81–88.

Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. 2012. Natural language questions for the web of data. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning EMNLP-CoNLL*, pages 379–390.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255–296.