POLY: Mining Relational Paraphrases from Multilingual Sentences

Adam Grycner
Max-Planck Institute for Informatics
Saarland Informatics Campus
Building E1.4, 66123
Saarbrücken, Germany
agrycner@mpi-inf.mpg.de

Gerhard Weikum
Max-Planck Institute for Informatics
Saarland Informatics Campus
Building E1.4, 66123
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

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 paper 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 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 paper presents a new method for systematically organizing a large set of such phrases. We aim to construct equivalence classes of synonymous phrases, analogously to how WordNet organizes unary predicates as noun-centric synsets (aka. semantic types). 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, 2001), there have been various attempts on building comprehensive resources for relational phrases. Recent works 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) 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 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

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2183–2192,
Austin, Texas, November 1-5, 2016.

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 ca. 1 Million relational phrases, organized into ca. 160,000 clusters.

Contribution. Our salient contributions are: i) a novel method for constructing a large repository of relational phrases, based on judicious clustering and type filtering; ii) a new linguistic resource, coined POLY, of relational phrases with semantic typing, organized in equivalence classes; iii) an intrinsic evaluation of POLY, demonstrating its high quality in comparison to PATTY and DEFIE; iv) an extrinsic evaluation of POLY, demonstrating its benefits for question answering. The POLY resource is publicly available [1].

2 Method Overview

Our approach consists of two stages: *relational phrase typing* and *relational phrase clustering*. In Section 3, we explain how we infer semantic types of the arguments of a relational phrase. In Section 4, we present the model for computing synonyms of relational phrases (i.e., paraphrases) and organizing them into clusters.

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 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 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 3 and 4.

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 output is a set of candidate types for the left and right arguments of each English relational phrase.

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 named entities, whose types can be looked up in a knowledge base, and the head words of common

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

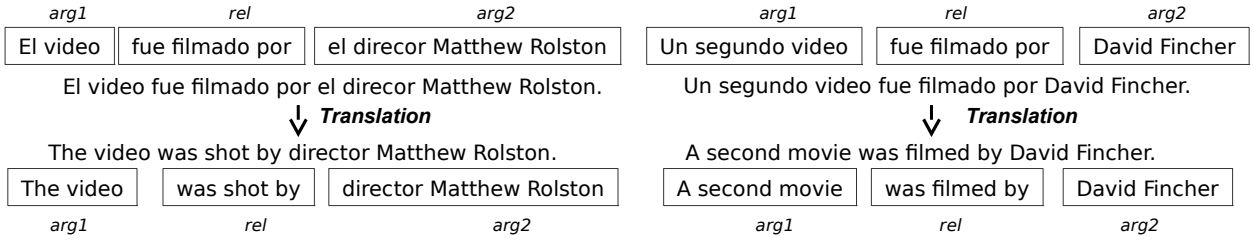

Figure 1: Multilingual input sentences and triples

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 set of manually crafted regular expressions. Since the input sentences result from machine translation, we could not use dependency parsing, because sentences are often ungrammatical.

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.

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.

Candidate senses are looked up in YAGO3 and WordNet, respectively, and each candidate is scored. The scores are based on:

- 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 titles) 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 (de Melo and Weikum, 2009).

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:

$$score(\mathrm{s}) = \alpha freq(\mathrm{s}) + \beta wiki(\mathrm{s}) + \gamma trans(\mathrm{s}) \quad (1)$$

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).

Finally, from the list of candidates, we generate a disambiguated argument: either a WordNet synset or a YAGO3 entity identifier.

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 (incl. 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 (incl. transitive hypernyms). Redundant and non-informative types were filtered out by the following technique: all types were organized into a directed acyclic graph (DAG), and

[2] opennlp.apache.org/

[3] mpi-inf.mpg.de/yago-naga/javatools/

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 the two arguments of each relational phrase.

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 DAG, based on type hypernymy. Then we prune types as follows (similarly to Section 3.3): i) remove a parent type when the relative frequency count of one of the children types is larger than 80% of the parent's count; ii) remove a child type when its relative frequency count is smaller than 20% of the parent's count.

For each of the two arguments of the relational phrase we allow only those types which are 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 arguments).

4 Relation Clustering

The second stage of POLY 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 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 and their argument types agree (see Figure 1 for an example). Similarities between English phrases are cast into edge weights of a graph with phrases as nodes. This graph is then partitioned to obtain clusters.

4.1 Probabilistic Similarity Measures

The phrase similarities in POLY are based on probabilistic measures. We use 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
- $c(f, e)$: no. 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 its translation $e \in E$
- $p(e|f) = \frac{c(f,e)}{c(f)}$: (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:

$$p(e_1|e_2) = \sum_f p(e_1|f) * p(f|e_2) \qquad (2)$$

as the probability of generating relational phrase $e_1 \in E$ from phrase $e_2 \in E$. Finally we define:

$$support(e_1, e_2) = \sum_{f \in F} c(f, e_1) * c(f, e_2) \qquad (3)$$

$$confidence(e_1, e_2) = \frac{2}{\frac{1}{p(e_1|e_2)} + \frac{1}{p(e_2|e_1)}} \qquad (4)$$

Confidence is the final similarity measure used in POLY. We use the harmonic mean in Equation 4 to dampen similarity scores that have big differences in their probabilities in Equation 2. Typically, pairs e_1, e_2 with such wide gaps in their probabilities come from subsumptions, not synonymous 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.

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

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>

Table 1: Example of a cluster of relational phrases

then ranked by their weighted graph density multiplied by the graph size (Equation 5). The example of a cluster is shown in Table 1.

$$\frac{\sum_{(e_i,e_j)\in E} sim(e_i,e_j)}{|V| * |V-1|} * |V| \quad (5)$$

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, 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 synonyms were organized into 158,725 clusters.

In the following, we present both an intrinsic evaluation 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 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.

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. We also assessed a sample of 250 pairs of synonyms, randomly drawn from POLY's output.

	Precision	Range
Top 250	0.91	$0.87 - 0.94$
Random	0.83	$0.78 - 0.87$

Table 2: Precision of synonym pairs in POLY

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 results of this evaluation are shown in Table 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 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. 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 clustered together because they were translated from the

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>

Table 3: Examples of synonyms of semantically typed relational phrases

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.

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

Table 4: Comparison to the competitors

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 3.2) are disregarded (*− disambiguation*), where the type system (Section 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 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 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 tech-

	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 5: Ablation Study

	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: Precision of synonyms (other languages)

niques 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 results in a worse noise filtering procedure.

5.4 Evaluation with Other Languages

In addition to paraphrases derived from German, we evaluated the relational phrase synonymy derived from a few other languages with lower numbers of extractions. We chose French, Hindi, and Russian (cf. (Faruqui and Kumar, 2015)). The results are presented in Table 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.

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 (i.e., would return with the perfect ranking).

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 collections: a triples database from OpenIE (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 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 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 random subset of questions from WikiAnswers (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 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.

	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 7: Number of questions with correct answer.

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), 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 (Vrandecic 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 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 Open Information Extraction (OIE) was developed to overcome the weak coverage of relations in automatically constructed KBs. OIE 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 them.

The first notable effort to build up a resource for relational paraphrases is DIRT (Lin and Pantel, 2001), 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 (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 OIE 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 OIE 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) clustered semantically equivalent English and French phrases, based on the arguments of relations.

7 Conclusions

We presented POLY, a method for clustering semantically typed English relational phrases using a multilingual corpus, resulting in a repository of semantically typed paraphrases with high coverage and precision. Future work includes jointly processing all 61 languages in the corpus, rather than considering them pairwise, to build a resource for all languages. The POLY resource is publicly available at `www.mpi-inf.mpg.de/yago-naga/poly/`.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *ACL*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

Colin J. Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL*.

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *ACL*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*.

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.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

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.

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 *AAAI*.

Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *CIKM*.

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 *KDD*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *ACL*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *KDD*.

Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In *NAACL*.

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

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.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *NAACL*.

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*.

Adam Grycner, Gerhard Weikum, Jay Pujara, James R. Foulds, and Lise Getoor. 2015. RELLY: Inferring hypernym relationships between relational phrases. In *EMNLP*.

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 *EMNLP*.

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 *IJCNLP*.

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

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*, 6:167–195.

Mike Lewis and Mark Steedman. 2013. Unsupervised induction of cross-lingual semantic relations. In *EMNLP*.

Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@discovery of inference rules from text. In *KDD*.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP*.

Andrea Moro and Roberto Navigli. 2012. WiseNet: building a wikipedia-based semantic network with ontologized relations. In *CIKM*.

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

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *EMNLP*.

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

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 *ACL*.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. 2015. CORE: Context-aware open relation extraction with factorization machines. In *EMNLP*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL*.

Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui, and Tingsong Jiang. 2015. Recognizing textual entailment using probabilistic inference. In *EMNLP*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR*.

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

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *ACL*.

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