Annotating Relation Inference in Context via Question Answering

Omer Levy **Ido Dagan**
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{omerlevy,dagan}@cs.biu.ac.il

Abstract

We present a new annotation method for collecting data on relation inference in context. We convert the inference task to one of simple factoid question answering, allowing us to easily scale up to 16,000 high-quality examples. Our method corrects a major bias in previous evaluations, making our dataset much more realistic.

1 Introduction

Recognizing entailment between natural-language relations (predicates) is a key challenge in many semantic tasks. For instance, in question answering (QA), it is often necessary to "bridge the lexical chasm" between the asker's choice of words and those that appear in the answer text. Relation inference can be notoriously difficult to automatically recognize because of semantic phenomena such as polysemy and metaphor:

Q: Which drug treats headaches?

A: Aspirin eliminates headaches.

In this context, "eliminates" implies "treats" and the answer is indeed "aspirin". However, this rule does not always hold for other cases – "eliminates patients" has a very different meaning from "treats patients". Hence, *context-sensitive* methods are required to solve relation inference.

Many methods have tried to address relation inference, from DIRT (Lin and Pantel, 2001) through Sherlock (Schoenmackers et al., 2010) to the more recent work on PPDB (Pavlick et al., 2015b) and RELLY (Grycner et al., 2015). However, the way these methods are evaluated remains largely inconsistent. Some papers that deal with phrasal inference in general (Beltagy et al., 2013; Pavlick et al., 2015a; Kruszewski et al., 2015) use an extrinsic task, such as a recent recognizing textual entailment (RTE) benchmark (Marelli et al., 2014). By nature, extrinsic tasks incorporate a variety of linguistic phenomena, making it harder to analyze the specific issues of relation inference.

The vast majority of papers that do focus on relation inference perform some form of post-hoc evaluation (Lin and Pantel, 2001; Szpektor et al., 2007; Schoenmackers et al., 2010; Weisman et al., 2012; Lewis and Steedman, 2013; Riedel et al., 2013; Rocktäschel et al., 2015; Grycner and Weikum, 2014; Grycner et al., 2015; Pavlick et al., 2015b). Typically, the proposed algorithm generates several inference rules between two relation templates, which are then evaluated manually. Some studies evaluate the rules out of context (is the rule "X *eliminates* Y"→"X *treats* Y" true?), while others apply them to textual data and evaluate the validity of the rule in context (given "aspirin *eliminates* headaches", is "aspirin *treats* headaches" true?). Not only are these post-hoc evaluations oblivious to recall, their "human in the loop" approach makes them expensive and virtually impossible to accurately replicate.

Hence, there is a real need for *pre-annotated* datasets for *intrinsic* evaluation of relation inference *in context*. Zeichner et al. (2012) constructed such a dataset by applying DIRT-trained inference rules to sampled texts, and then crowd-annotating whether each original text (premise) entails the text generated from applying the inference rule (hypothesis). However, this process is biased; by using DIRT to generate examples, the dataset is inherently blind to the many cases where relation inference exists, but is not captured by DIRT.

We present a new dataset for evaluating relation inference in context, which is *unbiased* towards one method or another, and *natural* to annotate. To create this dataset, we design a QA setting where annotators are presented with a single ques-

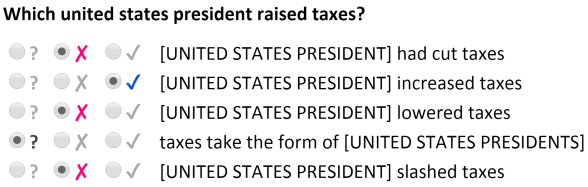

Figure 1: A screenshot from our annotation task.

tion and several automatically-retrieved text fragments. The annotators' goal is to mark which of the text fragments provide a potential answer to the question (see Figure 1). Since the entities in the text fragments are aligned with those in the question, this process implicitly annotates which *relations* entail the one in the question. For example, in Figure 1, if "[US PRESIDENT] increased taxes" provides an answer to "Which US president raised taxes?", then "increased" implies "raised" in that context. Because this task is so easy to annotate, we were able to scale up to 16,371 annotated examples (3,147 positive) with 91.3% precision for only $375 via crowdsourcing.

Finally, we evaluate a collection of existing methods and common practices on our dataset, and observe that even the best combination of methods cannot recall more than 25% of the positive examples without dipping below 80% precision. This places into perspective the huge amount of relevant cases of relation inference inherently ignored by the bias in (Zeichner et al., 2012). Moreover, this result shows that while our annotation task is easy for humans, it is difficult for existing algorithms, making it an appealing challenge for future research on relation inference. Our code[1] and data[2] are publicly available.

2 Relation Inference Datasets

To the best of our knowledge, there are only three pre-annotated datasets for evaluating relation inference in context.[3] Each example in these datasets consists of two binary relations, premise and hypothesis, and a label indicating whether the hypothesis is inferred from the premise. These relations are essentially Open IE (Banko et al., 2007) assertions, and can be represented as $(subject, relation, object)$ tuples.

Berant et al. (2011) annotated inference between *typed* relations ("[DRUG] *eliminates* [SYMPTOM]"→"[DRUG] *treats* [SYMPTOM]"), restricting the definition of "context". They also used the non-standard type-system from (Schoenmackers et al., 2010), which limits the dataset's applicability to other corpora. Levy et al. (2014) annotated inference between *instantiated* relations sharing at least one argument ("aspirin *eliminates* headaches"→"drugs *treat* headaches"). While this format captures a more natural notion of context, it also conflates the task of relation inference with that of entity inference ("aspirin"→"drug"). Both datasets were annotated by experts.

Zeichner et al. (2012) annotated inference between *instantiated* relations sharing *both* arguments:

aspirin *eliminates* headaches → aspirin *treats* headaches

aspirin *eliminates* headaches ↛ aspirin *murders* headaches

This format provides a broad definition of context on one hand, while isolating the task of relation inference. In addition, methods that can be evaluated on this type of data, can also be directly embedded into downstream applications, motivating subsequent work to use it as a benchmark (Melamud et al., 2013; Abend et al., 2014; Lewis, 2014). We therefore create our own dataset in this format.

The main drawback of Zeichner et al.'s process is that it is biased towards a specific relation inference method, DIRT (Lin and Pantel, 2001). Essentially, Zeichner et al. conducted a post-hoc evaluation of DIRT and recorded the results. While their approach does not suffer from the major disadvantages of post-hoc evaluation – cost and irreplicability – it ignores instances that do not behave according to DIRT's assumptions. These invisible examples amount to an enormous chunk of the inference performed when answering questions, which are covered by our approach (see §4).

3 Collection & Annotation Process

Our data collection and annotation process is designed to achieve two goals: (1) to efficiently sample premise-hypothesis pairs in an *unbiased* man-

[1] http://bitbucket.org/omerlevy/relation_inference_via_qa

[2] http://u.cs.biu.ac.il/~nlp/resources/downloads/relation_inference_via_qa

[3] It is worth noting the lexical substitution datasets (McCarthy and Navigli, 2007; Biemann, 2013; Kremer et al., 2014) also capture instances of relation inference. However, they do not focus on relations and are limited to single-word substitutions. Furthermore, the annotators are tasked with generating substitutions, whereas we are interested in judging (classifying) an existing substitution.

ner; (2) to allow for cheap, consistent, and scalable annotations based on an intuitive QA setting.

3.1 Methodology Overview

We start by collecting factoid questions. Each question is captured as a tuple $q = (q_{type}, q_{rel}, q_{arg})$, for example:

$$\text{Which } \overset{q_{type}}{\text{food}} \ \overset{q_{rel}}{\text{is included in}} \ \overset{q_{arg}}{\text{chocolate}}\,?$$

In addition to "Which?" questions, this template captures other WH-questions such as "Who?" (q_{type} = person).

We then collect a set of candidate answers for each question q. A candidate answer is also represented as a tuple $(a_{answer}, a_{rel}, a_{arg})$ or $(a_{arg}, a_{rel}, a_{answer})$, for example:

$$\overset{a_{arg}}{\text{chocolate}} \ \overset{a_{rel}}{\text{is made from}} \ \overset{a_{answer}}{\text{the cocoa bean}}$$

We collect answer candidates according to the following criteria:

1. $a_{arg} = q_{arg}$
2. a_{answer} is a type of q_{type}
3. $a_{rel} \neq q_{rel}$

These criteria isolate the task of relation inference from additional inference tasks, because they ensure that a's arguments are entailing q's. In addition, the first two criteria ensure that enough candidate answers actually answer the question, while the third discards trivial cases. In contrast to (Zeichner et al., 2012) and post-hoc evaluations, these criteria do not impose any bias on the relation pair a_{rel}, q_{rel}. Furthermore, we show in §3.2 that both a and q are both independent naturally-occurring texts, and are not machine-generated by applying a specific set of inference rules.

For each (a, q) pair, Mechanical Turk annotators are asked whether a provides an answer to q. This natural approach also enables batch annotation; for each question, several candidate answers can be presented at once without shifting the annotator's focus. To make sure that the annotators do not use their world knowledge about a_{answer}, we mask it during the annotation phase and replace it with q_{type} (see Figure 1 and §3.3).

Finally, we instantiate q_{type} with a_{answer}, so that each (a, q) pair fits Zeichner's format: instantiated predicates sharing both arguments.

3.2 Data Collection

We automatically collected 30,703 pairs of questions and candidate answers for annotation. Our process is largely inspired by (Fader et al., 2014).

Questions We collected 573 questions by manually converting questions from TREC (Voorhees and Tice, 2000), WikiAnswers (Fader et al., 2013), WebQuestions (Berant et al., 2013), to our "Which q_{type} q_{rel} q_{arg}?" format. Though many questions did fit our format, a large portion of them were about sports and celebrities, which were not applicable to our choice of corpus (Google books) and taxonomy (WordNet).[4]

Corpus QA requires some body of knowledge from which to retrieve candidate answers. We follow Fader et al. (2013; 2014), and use a collection of Open IE-style assertions (Banko et al., 2007) as our knowledge base. Specifically, we used hand-crafted syntactic rules[5] to extract over 63 million unique subject-relation-object triplets from Google's Syntactic N-grams (Goldberg and Orwant, 2013). The assertions may include multiword phrases as relations or arguments, as illustrated earlier. This process yields some ungrammatical or out-of-context assertions, which are later filtered during annotation (see §3.3).

Answer Candidates In §3.1 we defined three criteria for matching an answer candidate to a question, which we now translate into a retrieval process. We begin by retrieving all assertions where one of the arguments (subject or object) is equal to q_{arg}, ignoring stopwords and inflections. The matching argument is named a_{arg}, while the other (non-matching) argument becomes a_{answer}.

To implement the second criterion (a_{answer} is a type of q_{type}) we require a taxonomy T, as well as a word-sense disambiguation (WSD) algorithm to match natural-language terms to entities in T. In this work, we employ WordNet's hypernymy graph (Fellbaum, 1998) as T and Lesk (Lesk, 1986) for WSD (both via NLTK (Bird et al., 2009)). While automatic WSD is prone to some errors, these cases are usually annotated as nonsensical in the final phase.

Lastly, we remove instances where $a_{rel} = q_{rel}$.[6]

[4] This is the only part in our process that might introduce some bias. However, this bias is independent of existing relation inference methods such as DIRT.

[5] See supplementary material for a detailed description.

[6] Several additional filters were applied to prune non-grammatical assertions (see supplementary material).

3.3 Crowdsourced Annotation

Masking Answers We noticed that exposing a_{answer} to the annotator may skew the annotation; rather than annotating whether a_{rel} implies q_{rel} in the given context, the annotator might annotate whether a_{answer} answers q according to her general knowledge. For example:

Q: Which country borders Ethiopia?

A: Eritrea invaded Ethiopia.

An annotator might be misled by knowing in advance that Eritrea borders Ethiopia. Although an invasion typically requires land access, it does not imply a shared border, even in this context; "Italy invaded Ethiopia" also appears in our corpus, but it is not true that "Italy borders Ethiopia".

Effectively, what the annotator might be doing in this case is substituting q_{type} ("country") with a_{answer} ("Eritrea") and asking herself if the assertion $(a_{answer}, q_{rel}, q_{arg})$ is true ("Does Eritrea border Ethiopia?"). As demonstrated, this question may have a different answer from the inference question in which we are interested ("If a country invaded Ethiopia, does that country border Ethiopia?"). We therefore mask a_{answer} during annotation by replacing it with q_{type} as a placeholder:

A: [COUNTRY] invaded Ethiopia.

This forces the annotator to ask herself whether a_{rel} implies q_{rel} in this context, i.e. does invading Ethiopia imply sharing a border with it?

Labels Each annotator was given a single question with several matching candidate answers (20 on average), and asked to mark each candidate answer with one of three labels:

- ✓ The sentence answers the question.
- ✗ The sentence does not answer the question.
- ? The sentence does not make sense, or is severely non-grammatical.

Figure 1 shows several annotated examples. The third annotation (?) was useful in weeding out noisy assertions (23% of candidate answers).

Aggregation Overall, we created 1,500 questionnaires,[7] spanning a total of 30,703 (a, q) pairs. Each questionnaire was annotated by 5 different people, and aggregated using the unanimous-up-to-one (at least 4/5) rule. Examples that did not exhibit this kind of inter-annotator agreement were discarded, and so were examples which were determined as nonsensical/ungrammatical (annotated with ?). After aggregating and filtering, we were left with 3,147 positive (✓) and 13,224 negative (✗) examples.[8]

To evaluate this aggregation rule, we took a random subset of 32 questionnaires (594 (a, q) pairs) and annotated them ourselves (expert annotation). We then compared the aggregated crowdsourced annotation on the same (a, q) pairs to our own. The crowdsourced annotation yielded 91.3% precision on our expert annotations (i.e. only 8.7% of the crowd-annotated positives were expert-annotated as negative), while recalling 86.2% of expert-annotated positives.

4 Performance of Existing Methods

To provide a baseline for future work, we test the performance of two inference-rule resources and two methods of distributional inference on our dataset, as well as a lemma-similarity baseline.[9]

4.1 Baselines

Lemma Baseline We implemented a baseline that takes into account four features from the premise relation (a_{rel}) and the hypothesis relation (q_{rel}) after they have been lemmatized: (1) Does a_{rel} contain all of q_{rel}'s content words? (2) Do the relations share a verb? (3) Does the relations' active/passive voice match their arguments' alignments? (4) Do the relations agree on negation? The baseline will classify the example as positive if all features are true.

PPDB 2.0 We used the largest collection of paraphrases (XXXL) from PPDB (Pavlick et al., 2015b). These paraphrases include argument slots for cases where word order changes (e.g. passive/active).

Entailment Graph We used the publicly-available inference rules derived from Berant et al.'s (2011) entailment graph. These rules contain typed relations and can also be applied in a context-sensitive manner. However, ignoring the

[7] Each of our 573 questions had many candidate answers. These were split into smaller chunks (questionnaires) of less than 25 candidate answers each.

[8] This harsh filtering process is mainly a result of poor annotator quality. See supplementary material for a detailed description of the steps we took to improve annotator quality.

[9] To recreate the embeddings, see supplementary material.

types and applying the inference rules out of context worked better on our dataset, perhaps because Berant et al.'s taxonomy was learned from a different corpus.

Relation Embeddings Similar to DIRT (Lin and Pantel, 2001), we create vector representations for relations, which are then used to measure relation similarity. From the set of assertions extracted in §3.2, we create a dataset of relation-argument pairs, and use word2vecf (Levy and Goldberg, 2014) to train the embeddings. We also tried to use the arguments' embeddings to induce a context-sensitive measure of similarity, as suggested by Melamud et al. (2015); however, this method did not improve performance on our dataset.

Word Embeddings Using Google's Syntactic N-grams (Goldberg and Orwant, 2013), from which candidate answers were extracted, we trained dependency-based word embeddings with word2vecf (Levy and Goldberg, 2014). We used the average word vector to represent multi-word relations, and cosine to measure their similarity.

4.2 Results

Under the assumption that collections of inference rules are more precision-oriented, we also try different combinations of rule-based and embedding-based methods by first applying the rules and then calculating the embedding-based similarity only on instances that were not identified as positive by the rules. Since the embeddings produce a similarity score, not a classification, we plot all methods' performance on a single precision-recall curve (Figure 2).

All methods used the lemma baseline as a first step to identify positive examples; without it, performance drops dramatically. This is probably more of a dataset artifact than an observation about the baselines; just like we filtered examples where $a_{rel} \neq q_{rel}$, we could have used a more aggressive policy and removed all pairs that share lemmas.

It seems that most methods provide little value beyond the lemma baseline – the exception being Berant et al.'s (2011) entailment graph. Unifying the entailment graph with PPDB (and, implicitly, the lemma baseline) slightly improves performance, and provides a significantly better starting point for the method based on word embeddings. Even so, performance is still quite poor in absolute terms, with less than 25% recall at 80% precision.

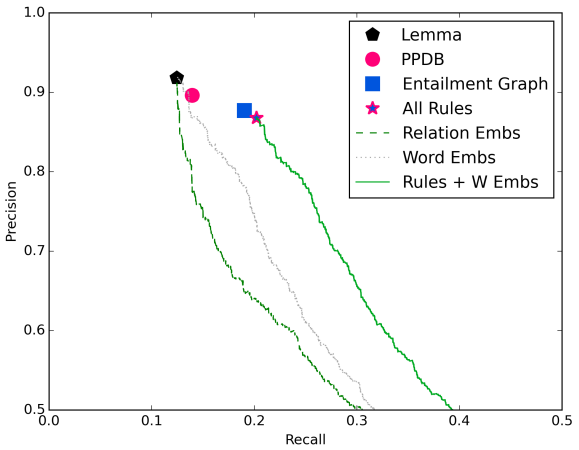

Figure 2: The performance of existing methods on our dataset. All methods are run on top of the lemma baseline. *All Rules* is the union of PPDB and the entailment graph. *Rules + W Embs* is a combination of *All Rules* and our word embeddings.

4.3 The Ramifications of Low Recall

These results emphasize the huge false-negative rate of existing methods. This suggests that a massive amount of inference examples, which are necessary for answering questions, are inherently ignored in (Zeichner et al., 2012) and post-hoc evaluations. Our dataset remedies this bias, and poses a new challenge for future research on relation inference.

Acknowledgements

This work was supported by the German Research Foundation via the German-Israeli Project Cooperation (grant DA 1600/1-1), the Israel Science Foundation grant 880/12, and by grants from the MAGNET program of the Israeli Office of the Chief Scientist (OCS).

References

[Abend et al.2014] Omri Abend, Shay B. Cohen, and Mark Steedman. 2014. Lexical inference over multi-word predicates: A distributional approach. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 644–654, Baltimore, Maryland, June. Association for Computational Linguistics.

[Banko et al.2007] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial*

Intelligence, Hyderabad, India, January 6-12, 2007, pages 2670–2676.

[Beltagy et al.2013] Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets markov: Deep semantics with probabilistic logical form. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 11–21, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

[Berant et al.2011] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA, June. Association for Computational Linguistics.

[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*, pages 1533–1544, Seattle, Washington, USA, October. Association for Computational Linguistics.

[Biemann2013] Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

[Bird et al.2009] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

[Fader et al.2013] Anthony Fader, Luke 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 (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Fader et al.2014] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1156–1165. ACM.

[Fellbaum1998] Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

[Goldberg and Orwant2013] Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

[Grycner and Weikum2014] Adam Grycner and Gerhard Weikum. 2014. Harpy: Hypernyms and alignment of relational paraphrases. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2195–2204, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

[Grycner et al.2015] Adam Grycner, Gerhard Weikum, Jay Pujara, James Foulds, and Lise Getoor. 2015. Relly: Inferring hypernym relationships between relational phrases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 971–981, Lisbon, Portugal, September. Association for Computational Linguistics.

[Kremer et al.2014] Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden, April. Association for Computational Linguistics.

[Kruszewski et al.2015] Germán Kruszewski, Denis Paperno, and Marco Baroni. 2015. Deriving boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.

[Lesk1986] Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26. ACM.

[Levy and Goldberg2014] Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

[Levy et al.2014] Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for open ie propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Lewis and Steedman2013] Mike Lewis and Mark Steedman. 2013. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

[Lewis2014] Mike Lewis. 2014. *Combined Distributional and Logical Semantics*. Ph.D. thesis, University of Edinburgh.

[Lin and Pantel2001] Dekang Lin and Patrick Pantel. 2001. Dirt: Discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

[Marelli et al.2014] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1314.

[McCarthy and Navigli2007] Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic, June. Association for Computational Linguistics.

[Melamud et al.2013] Oren Melamud, Jonathan Berant, Ido Dagan, Jacob Goldberger, and Idan Szpektor. 2013. A two level model for context sensitive inference rules. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1331–1340, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Melamud et al.2015] Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado, June. Association for Computational Linguistics.

[Pavlick et al.2015a] Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Benjamin Van Durme, and Chris Callison-Burch. 2015a. Adding semantics to data-driven paraphrasing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1512–1522, Beijing, China, July. Association for Computational Linguistics.

[Pavlick et al.2015b] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015b. 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 (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia, June. Association for Computational Linguistics.

[Rocktäschel et al.2015] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado, May–June. Association for Computational Linguistics.

[Schoenmackers et al.2010] Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, MA, October. Association for Computational Linguistics.

[Szpektor et al.2007] Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463, Prague, Czech Republic, June. Association for Computational Linguistics.

[Voorhees and Tice2000] Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 200–207. ACM.

[Weisman et al.2012] Hila Weisman, Jonathan Berant, Idan Szpektor, and Ido Dagan. 2012. Learning verb inference rules from linguistically-motivated evidence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 194–204, Jeju Island, Korea, July. Association for Computational Linguistics.

[Zeichner et al.2012] Naomi Zeichner, Jonathan Berant, and Ido Dagan. 2012. Crowdsourcing inference-rule evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 156–160, Jeju Island, Korea, July. Association for Computational Linguistics.