Learning Relation Entailment Graphs
CSE 515 Final Report

Mitchell Koch, Justin Huang

Friday, June 14, 2013

Abstract

Relation entailment is an interesting problem for question answering tasks. Not every question can be answered directly from text, making inference using implications important. Additionally, in information extraction systems like Open IE [3], it would be beneficial to have entailment information in order to give more results for relation queries.

We approached the problem of relation entailment for question answering using two methods. One is using Bayes net structure learning to learn a graph of relation influences given query results, and another is leveraging the WordNet resource, mapping relation strings to WordNet senses in order to use entailments in WordNet.

Using these methods to expand relation queries in order to improve queries, we found that Bayes net structure learning reduced precision, because we were not able to find a way to bias the directionality of edges in a way appropriate for the problem. Using WordNet sense entailments with a logistic regression classifier to select which paths are likely, we were able to improve the results returned while maintaining high precision of query results.

1 Introduction and Motivation

Relation extraction systems are able to find examples of predicates between arguments in text. A goal of relation extraction is question answering, and for this, queries must be run over results from relation extraction.

Can learning relation entailment graphs based on WordNet or Bayes net structure learning improve relation query performance?

Expanding relation queries is useful in any relation extraction context. This could be over surface forms (relation strings), as in Open IE [3] or structured relations from Freebase. a We focused on Open IE, where the relations and arguments are all strings from sentences, but these methods could also apply in a structured context.

previous relation entailment work [2, 1], on work in relation extraction using matrix factorization [6], as well as on related work in probabilistic modeling of relations between entities [8, 7].

Freebase using distant supervision as in [5]

1.1 Database Query Task

We define the following database query task to evaluate a relation entailment graph. Let R be a set of relation strings, and A be a set of argument strings. We are given a database D of facts, where each fact is of the form $r(x, y)$ for some $r \in R$ and $x, y \in A$.

Given a query for a relation r with arguments x and y, we can expand the query to additionally search for all $r'(x, y)$ such that there exists an edge from r' to r in the entailment graph. An example of this process is shown in figure 1. We can then evaluate the correctness of the returned results.

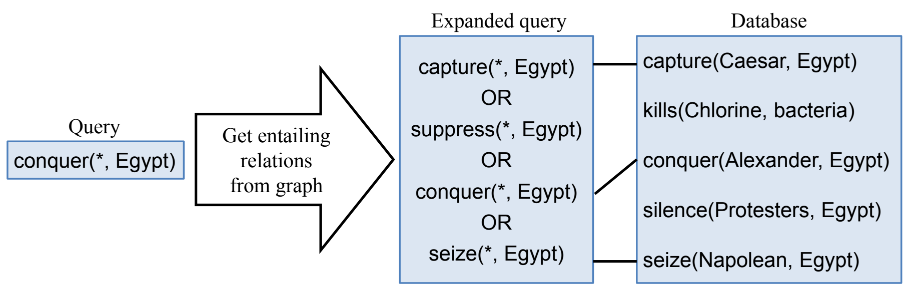

Figure 1: An example of a query being expanded.

2 Methods and Algorithms

2.1 Constraints on WordNet

WordNet [4] is a hand-crafted resource that distinguishes between different word senses, and provides synonyms and entailments between them. We denote a WordNet sense with a number where *#1* is the most common. For example *note#4* is the fourth most common sense of "note," which means "to write down". Each WordNet sense has its sense number, a count indicating how often that sense is used, and a probability, which is the count divided by the sum of counts for all senses of the same word.

2.1.1 WordNet entailment graph

A troponym is a specialization of a word, e.g., "to fly" is a troponym of "to travel." We say that WordNet sense w_1 entails w_2 if w_1 is a synonym of w_2, if w_1 is a troponym of w_2 in WordNet, or if there exists some w_3 such that

w_1 entails w_3 and w_3 entails w_2. A path in the entailment graph is defined as the sequence of troponyms between w_1 and w_2. The entailment graph is disconnected. Figure 2 shows one component of the graph, and figure 3 shows an example of a path in the graph, from *note#4* to *write#2*.

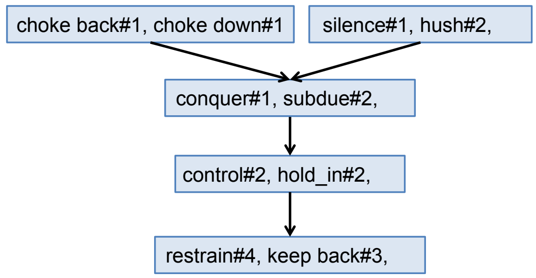

Figure 2: Component of the WordNet entailment graph with the first sense of conquer, *conquer#1*. Boxes represent synonym sets, arrows represent entailments.

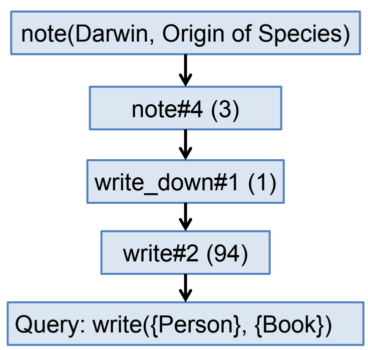

Figure 3: Example path in entailment graph

Although WordNet already defines an entailment graph between verb senses, one hurdle to using it is determining what sense to use. For example, given the verb "take," is it *take#21* (take by force) or *take#2* (take time)? Each sense will lead to very different entailments. First we assume a string can take on any of its WordNet senses, which may lead to noisier entailments. Then we train a logistic regression classifier to help filter out noisy entailments.

2.1.2 Logistic regression classifier

For a certain set of queries, we use all possible entailments from WordNet on the database query task, and label each fact as relevant to the query or not relevant to the query. This data is subsequently used to train the logistic regression model. For each result, we record the path that was taken in the entailment graph and produce the following features:

1. Path length
2. Average sense number
3. Average WordNet probability
4. Maximum sense number
5. Minimum WordNet probability

The length of the path is how many nodes there were in the path. For example, in figure 3, the length is 3. Because WordNet senses are ranked in order frequency, we use the average and maximum sense numbers as features, the idea being that an uncommon WordNet sense (e.g., *take#42*) could be a weak link in the entailment path. A similar intuition holds for the probabilities of the WordNet senses.

The output of logistic regression is a set of weights w for each feature. Let x be the vector of features described above. Then the probability of the label we would like to predict is given by:

$$p(x) = \frac{e^{w \cdot x}}{1 + e^{w \cdot x}}$$

2.2 Bayesian Network Structure Learning

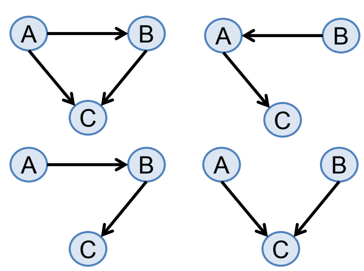

Figure 4: Examples of possible network structures

3 Experiments and Results

We evaluated our relation entailment graphs on the database query task, as described in section 1.1. The set of queries we chose to evaluate with are listed in table 1.

Table 1: Benchmark queries for database query task.

Query
has-written(type: Person, type: Book)
play(Tom Hanks, ?)
conquer(?, Egypt)
killed(?, Voldemort)
grown-in(coffee, type: Country)

Our baseline is a system that does no query expansion, only searching the database for an exact string match of the relation phrase. We compared this to systems that expanded queries based on the entailment graphs from 1) all of WordNet, 2) WordNet with logistic regression, and 3) Bayes net structure learning.

We labeled the results from the full WordNet system, and trained the logistic regression model on the labeled data using 10-fold cross validation.

The input to the structure learning was a matrix of counts, with relation phrases for rows and entity pairs for columns. However, we were constrained by high memory usage when running structure learning on graphs with more than 150 relation phrases. As a result, for some queries, we had to randomly sample 150 relation phrases. The search was done using 100 samples of Markov Chain Monte Carlo (MCMC), with a burn-in of 10 samples.

Our results are summarized in table 2. Structure learning had no effect on the results, because none of the entailments learned through structure learning applied to the five queries listed in table 1.

Table 2: Results on database query task for queries in table 1.

System	# results returned	Precision
Baseline	115	83.48%
Structure learning	115	83.48%
WordNet	1105	86.85%
WordNet + logistic regression	1035	91.50%

These results show that using entailments from WordNet improved recall substantially. We were surprised that using WordNet entailments also increased precision, but that likely stems from our choice of queries. For example, while not many facts in the database specifically say a person “has written” a book, many facts say a person “writes” a book. Since we consider those to be correct answers, those results push precision up. Adding the logistic regression classifier presented us with a precision/yield tradeoff.

To see the results of structure learning, we also tried a set of 10 benchmark queries that had entries in the entailment graph generated from structure learning. The queries are listed in table 3, and our results are shown in table 4.

Table 3: Additional benchmark queries to evaluate structure learning.

Query
be-one-of(coffee, type:Country)
work-on(type:Person, type:Book)
be-solidly-establish in(coffee, type:Country)
believe-in(type:Person, type:Book)
play-major-economic-role-in(coffee, type:Country)
be-come-in(type:Person, type:Book)
discuss-character-of(Tom Hanks, ?)
be-originally-find-in(coffee, type:Country)
tell(Tom Hanks, ?)
endorse(Tom Hanks, ?)

Table 4: Results on database query task for queries in figure 3.

System	# results returned	Precision
Baseline	55	98.18%
Structure learning	73	76.71%
WordNet	108	56.48%
WordNet + logistic regression	58	91.38%

This set of results shows that our structure learning will increase the number of results returned, but at a high cost to precision. These results also show that for the type of more specific queries presented in table 3, almost all the extra results given by WordNet entailments are bad. The classifier partially recovered the loss in precision.

4 Discussion and Conclusion

Our results suggest that the choice of benchmark queries can make a big difference in the results. However, we noticed that using WordNet with a classifier does seem to increase yield at an acceptable level of precision. Although it's hard to say what queries should be used in our experiment, the queries in table 1 are arguably more common than those in table 3.

References

[1] J. Berant, I. Dagan, and J. Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 610–619, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[2] J. Berant, I. Dagan, and J. Goldberger. Learning entailment relations by global graph structure optimization. *Comput. Linguist.*, 38(1):73–111, Mar. 2012.

[3] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, Dec. 2008.

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

[5] R. Hoffmann, C. Zhang, X. Ling, L. S. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550, 2011.

[6] S. Riedel, L. Yao, B. M. Marlin, and A. McCallum. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, June 2013.

[7] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, UAI'02, pages 485–492, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[8] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *NIPS*, 2003.