Path Analysis for Refining Verb Relations

Timothy Chklovski and Patrick Pantel
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
{timc, pantel}@isi.edu

ABSTRACT

Link discovery is the process of identifying complex patterns from (multi)-relational data. The quality of link discovery outputs depends on the quality of the underlying data. In this paper, we discuss a method for refining multi-relational data. We treat the data as a graph and apply global link analysis to refine the graph. Specifically, we re-estimate the presence of a relation between a pair of nodes from the evidence provided by multiple indirect paths between the nodes. Our approach applies to a variety of relations: transitive symmetric, transitive asymmetric, and relations inducing equivalence classes. We present preliminary results on a semantic network called VERBOCEAN, which contains 22,306 relations between 3,477 verbs.

Categories and Subject Descriptors

I.2.4 **[Knowledge Representation Formalisms and Methods]**: *Semantic networks*

General Terms

Algorithms, Theory, Verification.

Keywords

Global link analysis, semantic networks, plausible inference, graph refinement.

1. INTRODUCTION

Link discovery is the process of identifying complex patterns from (multi-)relational data, which can be conceptualized as a graph where entities are nodes and relations between pairs of entities are edges. The quality of link discovery output depends on the quality of the underlying data, which is often noisy. The data is typically extracted on a per link basis (i.e., a link between two nodes is determined without regard to other nodes). Yet, a global view of the graph may provide additional information to refine local decisions by identifying inconsistencies, updating confidences in links and suggesting new links.

Noisy or incomplete graphs are encountered in many areas, including semantic networks, social networks, citation analysis, and coreference resolution. A global view of such graphs can be used to refine them. For example, in semantic networks, observing "discover *happens-before* refine," and "refine *happens-before* exploit" provides evidence for "discover *happens-before* exploit," because the relation *happens-before* is transitive. In social networks, observing "E_1 *has-coworker* E_2" (essentially stating that E_1 and E_2 are in the same equivalence class with respect to employer) and "E_1 *works-for O*" provides evidence for "E_2 *works-for O*."

In this paper, we investigate an approach to graph refinement, in which nodes represent entities and links represent relations between entities, using a global analysis relying on link semantics. Our approach is based on the observation that some paths (chains of relations) between a pair of nodes x_i and x_j imply the existence or absence of a particular relation between x_i and x_j. Despite each individual path being noisy, multiple indirect paths can provide sufficient evidence for adding, removing, or altering a relation between two nodes. As illustrated by the above examples, inference of a relation based on the presence of a certain path often relies on transitivity of the individual relations that make up the path, or on some relations on the path indicating membership in the same equivalence class.

As a testbed, we use VERBOCEAN, a broad-coverage noisy network of semantic relations between verbs extracted by mining the Web. We demonstrate refinements offered by our approach on VERBOCEAN's two transitive asymmetric relations and on two relations which induce equivalence classes between elements.

The remainder of this paper is organized as follows. Section 2 describes VERBOCEAN and Section 3 presents our filtering algorithm. Preliminary results are presented in Section 4 and finally, we conclude with a discussion and future work.

2. VERBOCEAN

We apply our global link analysis approach to VERBOCEAN, a resource of lexical semantics with potential applications to a variety of natural language tasks ranging from question answering and information retrieval to document summarization and machine translation. VERBOCEAN has been extracted from the web, as described in [1], and outlined in the Appendix.

For our purposes, VERBOCEAN is a graph of semantic relations between verbs, with 3,477 verbs (nodes) and 22,306 relations (edges). As shown in Table 1, VERBOCEAN contains five types of relations: *similarity, strength, antonymy, enablement,* and *happens-before*. Senses are not discriminated and an edge indicates that the relation is believed to hold between some senses of the verbs in this relation. *Similarity* is a relation that suggests

LinkKDD04 workshop, KDD2004, August 22–25, 2004, Seattle, WA.

Table 1. Types, examples and frequencies of 22,306 semantic relations in VerbOcean.

Semantic Relation	*Example*	*Transitive*	*Symmetric*	*Num in VerbOcean*
similarity	produce :: create	Y	Y	11,515
strength	wound :: kill	Y	N	4,220
antonymy	open :: close	N	Y	1,973
enablement	fight :: win	N	N	393
happens-before	buy :: own; marry :: divorce	Y	N	4,205

Table 2. Precision of semantic relations in VerbOcean on a random sample of 100 pairs.

Semantic Relation	*System Tags*	*Tags Correct*	*Preferred Tags Correct*
similarity	41	63.4%	40.2%
strength	14	75.0%	75.0%
antonymy	8	50.0%	43.8%
enablement	2	100%	100%
happens-before	17	67.6%	55.9%

two nodes are likely to be in the same equivalence class, although polysemy and sense drift make it only weakly transitive. The *strength* relation is a subtype of *similarity*. Unlike *similarity*, it is an asymmetric relation. *Strength*, or *stronger-than*, holds whenever a verb denotes a more intense, thorough, comprehensive or absolute action. Being narrower than *similarity*, it seems to be more strongly transitive in practice, and we treat it as transitive. *Antonymy*, a dual of similarity, is symmetric but not transitive. In treating antonymy, we leverage the observation that being the antonym of an antonym suggests similarity, e.g., *adore opposite-of despise*, *despise opposite-of love,* and *adore* is similar to *love*. *Enablement* is rare and is not considered here. Finally, *happens-before* is a transitive asymmetric temporal relation between verbs.

When outputting VerbOcean relations, the extraction algorithm enforces link-level unidirectionality of asymmetric links between two nodes (i.e., it ensures that if there is an edge x_i *happens-before* x_j, then there is no edge x_j *happens-before* x_i). Larger cycles, however, are possible, as extraction is strictly local. The extraction algorithm also outputs at most one of *similarity, strength, antonymy, enablement*. The most strongly manifested of *similarity, antonymy,* and *enablement* is preferred, and *strength*, if detected, is preferred over *similarity*. When *strength* relation is output, an edge explicitly indicating *similarity* is not created. *Happens-before*, being complimentary to other relations, is allowed to be present along with another relation between a pair of nodes. For example, "wrap" and "unwrap" are antonyms, and also "wrap *happens-before* unwrap."

Precision of the detected relations, based on evaluating a random sample of 100 verb pairs for which extraction of a relation is attempted, is given in Figure 2. The *Tags Correct* column represents the percentage of verb pairs whose system output relations were deemed correct. The *Preferred Tags Correct* column gives the percentage of verb pairs whose system output relations matched exactly the human's preferred relations. Precision of relations varies, with asymmetric relations generally being more precise (possibly because asymmetric relations, when being detected, must pass an additional test for the ratio of frequency in the forward vs. reverse direction).

3. GLOBAL REFINEMENT

Our approach uses a global view of the graph to refine a relation between a given pair of nodes x_i and x_j. The refinement relies on multiple paths between x_i and x_j. The analysis processes triples < x_i, r, x_j> consisting of nodes x_i and x_j and relation r. The analysis outputs the relation r, its opposite (which we will denote r'), or neither. The opposite of *happens-before* and *stronger-than* is the same relation in the reverse direction, and the opposite of *similarity* is *antonymy* and vice versa.

In the remainder of this section, we explain how triples are selected and processed to refine the graph, what paths are used as evidence, and present the statistical model for combining evidence from multiple paths.

3.1 Testing for relations

In this section, we present our method for extracting the set of triples <x_i, r, x_j> to process. Although more extensive sets of triples can be considered, we focus on the set of triples <x_i, r, x_j> such that r is one of {*similarity, antonymy, happens-before* and *stronger-than*} and the relation r is present between x_i and x_j in the original graph.

We now introduce some notation. Let $R_{i,j}$ denote the event that the relation r is present between nodes x_i and x_j in the original graph (i.e., the graph indicates the presence of the relation r between x_i and x_j, but it might be an error). Let $r_{i,j}$ denote the event that the relation r actually holds between x_i and x_j. Let $C_{i,j}$ denote an acyclic path from x_i to x_j of (possibly distinct) relations $\{R_{i,i+1} .. R_{j-1,j}\}$. For example, the path "x_1 *happens-before* x_2 *happens-before* x_3" can be written as $C_{1,3}$. If the edges of $C_{i,j}$ indicate the relation r between the nodes x_i and x_j, we say that $C_{i,j}$ *indicates* $r_{i,j}$.

Given a triple <x_i, r, x_j>, we identify the set $\boldsymbol{C}$ of all paths $C_{i,j}$ which may have arbitrary intermediate nodes but which must match one of the allowed sequences of relations (allowed sequences of relations for every r are described in Section 3.2).

For each $C_{i,j}$ in $\boldsymbol{C}$, we compute its score, which is the estimated probability that $r_{i,j}$ holds given the observation of edges of $C_{i,j}$. The statistical model for estimating these probabilities is given in Section 3.3.1. Longer paths and paths made up of less reliable edges will have lower scores.

Next, we filter $\boldsymbol{C}$ to form the set $\boldsymbol{C'}$ of paths which indicate $r_{i,j}$ and which have no common intermediate nodes. This is done using a greedy approach by processing all paths in $\boldsymbol{C}$ in order of decreasing score, placing each in $\boldsymbol{C'}$ iff it does not share any intermediate nodes with any path already in $\boldsymbol{C'}$.

Next, the total score $Sc(r_{i,j})$ for presence of $r_{i,j}$ is calculated from the scores of the nonintersecting, nonoverlapping paths in $\boldsymbol{C'}$, as described in Section 3.3.2. The score $Sc(r'_{i,j})$ for $r'_{i,j}$, the opposite of $r_{i,j}$ is calculated similarly (i.e., by combining scores from paths in the other direction in the case of asymmetric transitive relations, and by finding paths for *antonymy* in the case of *similarity*).

Finally, the scores are combined to make the final decision. In order to output $r_{i,j}$, two conditions must be met:

$$Sc(r_{i,j}) > \sigma_1$$

and

$$\frac{Sc(r_{i,j})}{Sc(r'_{i,j})} > \sigma_2$$

Currently, $\sigma_1 = 0.6$ and $\sigma_2 = 1.7$, as determined by inspection of system scores for some specific instances. Similarly, $r'_{i,j}$ is output if

$$Sc(r'_{i,j}) > \sigma_1$$

and

$$\frac{Sc(r'_{i,j})}{Sc(r_{i,j})} > \sigma_2$$

If neither pair of conditions is met, neither $r_{i,j}$ nor $r'_{i,j}$ are output (i.e., the edge is removed from the graph).

3.2 Paths considered

The enabling observation behind our approach is that in a graph in which edges have certain properties such as transitivity, some paths $C_{i,j}$ indicate the presence of a relation between the first node x_i and the last node x_j. In the paths we consider, we rely on two kinds of inferences: transitivity and equivalence. However, we do not consider paths composing the two types of inference (e.g. "*similar, happens-before, happens-before*"). Also, we do not consider very long paths, as they tend to become unreliable (recall that Figure 2 shows the observed precision of a given edge). In the preliminary investigations we report, the set of paths to consider was not rigorously motivated. Rather, we aimed to cover some common cases. Refining the sets of paths is a possible fruitful direction for future work.

3.2.1 Transitive asymmetric relations

In identifying paths which indicate *happens-before*, a transitive asymmetric relation, the following 6 path types are considered:

"happens-before"
"happens-before, happens-before"
"happens-before, happens-before, happens-before"
"happens-before, happens-before, happens-before, happens-before"
"similar, happens-before"
"happens-before, similar"

For the relation *stronger-than*, we consider similar paths, with *stronger-than* in place of *happens-before*.

3.2.2 Similarity

In paths indicating a given relation, we treat *similarity* as a weak form of equivalence. That is, we avoid chaining several *similarity* relations to avoid drift caused by polysemy and meaning drift. When assessing indication of *similarity* itself, using paths of *similarity* can be viewed as both transitivity and equivalence.

The *antonymy*, or *opposite-of* relation is a special case – a path of two oppositions indicates equivalence and a path of three oppositions indicates opposition again.

For presence of *similarity*, we consider all 13 paths with less that three edges which indicate *similarity*:

"similar"
"stronger-than"
"weaker-than"
"opposite-of, opposite-of"
nine paths of with two edges, each edge from the set {*similar, stronger-than, weaker-than*}

Note that in detecting *similarity* with paths of length two, we rely on the *strength* relation, which is a subtype of *similarity*.

3.2.3 Opposition

For presence of opposition, the following 8 path types were considered (all paths with fewer than three edges and one with three edges) indicating *opposition*:

"opposite-of"
"opposite-of, similar"
"opposite-of, stronger-than"
"opposite-of, weaker-than"
"similar, opposite-of"
"stronger-than, opposite-of",
"weaker-than, opposite-of",
"opposite-of, opposite-of, opposite-of"

3.3 Statistical model for combining evidence

Our goal is to estimate the validity of inferring $r_{1,n}$ given a collection of m paths $C_{1,n}{}^1$, $C_{1,n}{}^2$, …, $C_{1,n}{}^m$:

$$P(r_{1,n} \mid C_{1,n}{}^1, C_{1,n}{}^2, ..., C_{1,n}{}^m) \qquad (1)$$

Then, $P(r_{i,j} \mid R_{i,j})$ denotes the probability that the relation is actually true given that it was detected.

3.3.1 Estimating the implication of a single path

We first estimate the implication $r_{1,n}$ of a single path $C_{1,n}$:

$$P(r_{1,n} \mid C_{1,n})$$

For the case where $C_{1,n}$ consists of exactly one edge, this is simply:

$$P(r_{1,2} \mid R_{1,2})$$

For paths with more than one edge, notice that $r_{1,n}$ is 1 if all edges of the path hold (the likelihood of which can be estimated from the observed edges in the graph). Otherwise, if one or more of the edges do not hold, $r_{1,n}$ is independent of whether the individual observed edges $R_{i,j}$ in the path hold or not:

$$P(r_{1,n} \mid R_{1,2} .. R_{n-1,n}) = \begin{cases} 1 & \text{if } r_{1,2} .. r_{n-1,n} \\ P(r_{1,n}) & \text{otherwise} \end{cases}$$

We can estimate the probability of $r_{1,2}, \ldots, r_{n\text{-}1,n}$ given $R_{1,2}, \ldots, R_{n\text{-}1,n}$, yielding:

$$P(r_{1,n} \mid R_{1,2} .. R_{n-1,n}) =$$
$$P(r_{1,2} .. r_{n-1,n} \mid R_{1,2} .. R_{n-1,n}) + (1 - P(r_{1,2} .. r_{n-1,n} \mid R_{1,2} .. R_{n-1,n}))P(r_{1,n})$$

Assuming that a path of true edges $r_{1,2}, \ldots, r_{n\text{-}1,n}$ is conditionally independent of observed edges $R_{1,2}, \ldots, R_{n\text{-}1,n}$ gives:

$$P(r_{1,n} \mid R_{1,2} .. R_{n-1,n}) =$$
$$\prod\nolimits_{i=1..n-1} P(r_{i,i+1} \mid R_{i,i+1}) + \left(1 - \prod\nolimits_{i=1..n-1} P(r_{i,i+1} \mid R_{i,i+1})\right)P(r_{1,n})$$

which can be rewritten as:

$$P(r_{1,n} \mid R_{1,2} .. R_{n-1,n}) = P(r_{1,n}) + (1 - P(r_{1,n})) \cdot \prod\nolimits_{i=1..n-1} P(r_{i,i+1} \mid R_{i,i+1})$$

$P(r_{1,n})$ and $P(r_{i,i+1} \mid R_{i,i+1})$ can be estimated empirically by manually tagging the relations $R_{i,j}$ in a graph as correct or incorrect (i.e. they are the probability of an edge being labeled as such by a human judge and the system precision for this type of edge, respectively).

3.3.2 Combining estimates from multiple paths

To compute Eq. 1, we must combine evidence from multiple paths. Some paths may contribute to the presence of the implication $r_{1,n}$ while others contribute to the presence of its opposite (i.e. for the asymmetric relations, the presence of the relation in the reverse direction, and for *similarity* the presence of *antonymy*).

As the simplest case, consider $C_{1,n}{}^{a}$ and $C_{1,n}{}^{b}$, two independent paths (paths not sharing any intermediate nodes) connecting nodes x_1 to x_n and implying the same relation $r_{1,n}$. We wish to compute the probability of the implication given the two paths:

$$P(r_{1,n} \mid C_{1,n}{}^{a}, C_{1,n}{}^{b}) \qquad (2)$$

For short, let C_a and C_b denote $C_{1,n}{}^{a}$ and $C_{1,n}{}^{b}$, respectively, and r denote $r_{1,n}$.

Rewriting Eq. 2 using Bayes theorem, yields:

$$P(r \mid C_a, C_b) = \frac{P(r)}{P(C_a, C_b)} P(C_a, C_b \mid r)$$

However, it is unclear how to compute the joint probability of C_a and C_b. We sidestep the issue by assuming that C_a and C_b are independent. We get the following score (no longer a true probability):

$$\overline{P}(r \mid C_a, C_b) = \frac{P(r)}{P(C_a)P(C_b)} P(C_a \mid r)P(C_b \mid r) \qquad (3)$$

Using Bayes again, we get:

$$\overline{P}(r \mid C_a, C_b) = \frac{1}{P(r)} \frac{P(r)}{P(C_a)} P(C_a \mid r) \frac{P(r)}{P(C_b)} P(C_b \mid r)$$
$$= \frac{P(r \mid C_a)P(r \mid C_b)}{P(r)}$$

Generalizing from two paths to m, we arrive at the score which we use in place of Eq. 1:

$$\overline{P}(r \mid C_a \ldots C_m) = \frac{\prod_{i=a \ldots m} P(r \mid C_i)}{P(r)^{m-1}} .$$

The independence assumption made in Eq. 3 does not actually hold, and for large m can yield scores in excess of 1,000 rather than scores in the interval [0, 1]. In the case when zero paths are found in one direction, the score is simply *P(r)*, the probability of observing *r* with no additional evidence.

4. PRELIMINARY RESULTS

In this section, we present some preliminary results on refining the semantic verb relations in VerbOcean. We first examine how well our method identifies incorrect relations in VerbOcean. Then, we inspect certain system outputs and present a discussion and error analysis.

4.1 Refining VerbOcean

Chklovski and Pantel evaluated the semantic relations in VerbOcean, which contains 22,306 relations between 3,477 verbs, by randomly sampling 100 highly correlated verb pairs (see the Appendix) and presenting the classifications to two human judges [1]. Table 2 shows the results.

Of the 100 pairs, 66 were identified to have a relation. We applied our refinement algorithm to VerbOcean and inspected the output. On the 37 relations that VerbOcean got wrong, our system identified six of them. On the remaining 29 that VerbOcean got correct, only one was identified as incorrect (false positive). Hence, on the task of identifying incorrect relations in VerbOcean, our system has a precision of 85.7%, where precision is defined as the percentage of correctly identified erroneous relations. However, it only achieved a recall of 16.2%, where recall is the percentage of erroneous relations that our system identified. Table 3 presents the relations that were refined by our system. The first two columns show the verb pair while the next two columns show the original relation in VerbOcean

4.2 Discussion

Based on inspection of some faulty refinements, the failures seem to stem noise in the supporting paths or from reliance on paths which do not actually indicate the relation (e.g. a path consisting of *similarity* and *stronger than* edges often fails to indicate the *strength* relation in cases we inspected).

We highlight some noteworthy aspects of our approach and illustrate prospects of extending it with some refinements the system suggested given a certain triple as input.

Our algorithm currently tests only the triples already in the graph. Thus, the algorithm was able to revise the relation “doom *opposite-of* complicate” to “doom *similar* complicate.” Yet, the revision did not affect the precision in the evaluation, because the algorithm did not detect the more specific relation “doom *stronger-than* complicate,” indicated by two human judges. If the algorithm was extended to test the strength relation whenever it detected similarity, the triple “doom *stronger-than* complicate” would have been tested. This test would have found three paths supporting the stronger-than relation:

Table 3. Seven relations in VerbOcean refined by our system.

Verb 1	*Verb 2*	*VerbOcean Relation*	*Refinement Relation*	*Judge 1 Relation*	*Judge 2 Relation*	*Judge 3 Relation*
attach	use	*happens-before* *similar*	*similar*	*none*	*none*	*none*
bounce	get	*stronger than (reverse)*	*stronger than*	*none*	*none*	*none*
dispatch	defeat	*opposite*	*none*	*none*	*none*	*happens-before*
doom	complicate	*opposite*	*similar**	*none*	*stronger-than*	*stronger-than*
flatten	level	*stronger than*	*no relation**	*similar*	*similar*	*similar*
outlaw	codify	*similar*	*opposite*	*none*	*none*	*opposition*
privatize	improve	*happens-before*	*none*	*happens-before*	*happens-before*	*happens-before*

* sources of potential improvements discussed in Section 4.2

DOOM *stronger-than* DELAY *stronger-than* COMPLICATE;
DOOM *similar* UNDERMINE *stronger-than* COMPLICATE;
DOOM *stronger-than* CRIPPLE *stronger-than* DISRUPT
stronger-than COMPLICATE;

there are also no paths at all supporting the stronger-than relation in the opposite direction. The three above paths have scores 0.61, 0.54, 0.49, respectively and a combined score of 12.13 which has a ratio of 105.44 over the score in the reverse direction 0.12 (which is the estimated prior that the relation holds given no additional evidence).

Another observation suggested by the preliminary results concerns the behavior of the algorithm when testing the *strength* relation. In the case of the triple <flatten, *strength*, level>, *stronger-than* relation was detected in both directions, causing the algorithm to output "no relation." In fact, because *strength* (in either direction) is a subclass of *similarity*, it may have been appropriate to output *similarity*, or at least test the triple <flatten, *similarity*, level> (which would find five paths supporting similarity and no paths supporting opposition, leading to scores of 25.1 and 0.07 for *similarity* and *opposition*, respectively, causing *similarity* to be detected. The above observations suggest changes to the algorithm which it would be important to evaluate more rigorously in future work.

Although in our evaluation we have only paid attention to outputting presence or absence of relations, the approach can potentially also update confidences in particular edges. For example, the original graph contains the following relations

DESTROY *stronger-than* DAMAGE (with score of 16.3)
DESTROY *stronger-than* ENGULF (with score of 14.0)
DESTROY *stronger-than* ATTACK (with score of 12.4)

Turning to graph-level analysis, however, uncovers five additional paths for "damage," none for "engulf," and one additional path for "attack."

A useful observation in future work on the system is that the actual decisions made by the algorithm depend on how scores are calculated for sets of paths, and what thresholds are chosen. We illustrate how scores stack up in a specific case in which *similarity* is correctly revised to *antonymy*. Consider the (incorrect) edge

REVEAL *similar* HIDE

present in the original graph. Applying graph-level analysis yields no additional paths supporting similarity, but discovers two paths supporting opposition:

REVEAL *opposite-of* CONCEAL *similar* HIDE
REVEAL *similar* DISCLOSE *opposite-of* HIDE

Both of these paths have a score of 0.37, and their combined score is 1.81. The score of the single similarity path is 0.63, and the ratio of the opposition to similarity score is 2.85, yielding reversal of similarity to opposition. Whatever scoring model is used, the outputted decision will stay the same given similar input (e.g. two *opposite-of* and *similar* paths vs. one *similar* path). Identifying the preferred way to treat such situations can be approached as a supervised learning problem.

Beyond testing more triples and improving handling of evidence combination, possible extensions to the algorithm include more elaborate types of inference from graph structure, for example treating absence of certain paths as counter-evidence. Suppose that relations A *opposite-of* B and A *similar* A' were detected, but the relation A' *opposite-of* B was not detected. Then, there is an "absent path":

A *similar* A' *opposite-of* B.

The absence of this path suggests absence of A *opposite-of* B.

Our work can be viewed as performing inferences over graphs in which the edges have particular properties and semantics. The problem is related to inference over graphical models (with the model potentially containing bidirectional edges and cycles); as such, recent work on Markov logic networks [2], may prove relevant and useful. Also, refining the *similarity* relation in particular is related to the rich body of work on classical graph clustering.

5. CONCLUSIONS

We presented a method for refining multi-relational data by applying global link analysis, leveraging multiple noisy paths. We re-estimated the presence of a relation between a pair of nodes from the evidence provided by multiple indirect paths between the nodes. Our approach applies to a variety of relation types:

transitive symmetric, transitive asymmetric, and relations inducing equivalence classes.

Preliminary results suggest that the system may be able to identify incorrect edges with high accuracy. Although the current system fails to consistently correct these identified edges, the experiments yielded several insights applying which may improve the performance. For example, evidence of both *stronger-than* and inverse *stronger-than* relations should conclude *similarity* instead of neither since *stronger-than* is a subset of *similarity*. For the same reason, the system should make a *strength* test when *antonymy* is changed to *similarity*.

Although the relations in VERBOCEAN have a weight based on the strength of the lexico-syntactic pattern matches on the Web, our approach does not currently leverage these weights. One future direction is to use the weights when factoring the contributions of edges within paths.

Successful link discovery is highly dependent of the quality of the (multi)-relational data to which it is applied. However, this data is often noisy. With the work presented here, we hope to motivate research in automatic methods to refine the quality of multi-relational data.

6. REFERENCES

[1] Chklovski, T., and Pantel, P. VERBOCEAN: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain, July 25-26, 2004, to appear.

[2] Domingos, P., and Richardson M. Markov Logic: A Unifying Framework for Statistical Relational Learning. In *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields*, 2004. Banff, Canada: IMLS. To appear.

[3] Lin, D. and Pantel, P. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360, 2001.

APPENDIX

VERBOCEAN (Chklovski and Pantel 2004) is a semantic network of relations between verb pairs. Currently, it contains 22,306 relations between 3,477 verbs. The relation types are: *similarity, strength, antonymy, enablement,* and *happens-before*.

The relations were identified by testing one pair of verbs at a time without reference to any other verbs (i.e. no global analysis). Specifically, relations were identified by querying a Web index (Google) with 35 lexico-syntactic patterns; for example, for the verb pair *discover* and *patent*, if "*discovered* and then *patented*" (the pattern "*Xed and then Yed*") was abnormally frequent on the web, the relation *happens-before* would be identified between *discover* and *patent*. The evidence from the 35 patterns were combined, using a measure similar to mutual information to account for both the frequency of the verbs as well as for the frequency of the pattern, to yield the strongest relation.

Creating VERBOCEAN was computationally intensive, requiring more than 30 queries per verb pair. In all, 10^6 queries to a positional index of more than 7×10^{11} words on the Web were issued. This volume is not nearly sufficient to test the millions of possible pairs of verbs. Instead, detection of relations was attempted only between 29,165 pairs of strongly associated verbs (the associated verbs were extracted from a smaller, 3.5×10^8 word parsed corpus, as described in [3]). As a result, 22,306 edges (relations) were extracted between 18,496 node pairs.