Global Path-Based Refinement of Noisy Graphs
Applied to Verb Semantics

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. Recently, researchers have applied text- and web-mining algorithms to mine semantic resources. The result is often a noisy graph of relations between words. We propose a mathematically rigorous refinement framework, which uses path-based analysis, updating the likelihood of a relation between a pair of nodes using evidence provided by multiple indirect paths between the nodes. Evaluation on refining temporal verb relations in a semantic resource called VERBOCEAN showed a 16.1% error reduction after refinement.

1 Introduction

Increasingly, researchers are creating broad-coverage semantic resources by mining text corpora [1][5] and the Web [2][6]. These resources typically consist of a noisy collection of relations between words. The data is typically extracted on a per link basis (i.e., the relation between two nodes is determined without regard to other nodes). Yet, little work has taken a global view of the graph of relations, which may provide additional information to refine local decisions by identifying inconsistencies, updating confidences in specific edges (relations), and suggesting relations between additional pairs of nodes.

For example, observing the temporal verb relations “discover happens-before refine” and “refine happens-before exploit” provides evidence for the relation “discover happens-before exploit,” because the happens-before relation is transitive.

We conceptualize a semantic resource encoding relations between words as a graph where words are nodes and binary relations between words are edges. In this paper, we investigate the refinement of such graphs by updating the confidence in edges 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 presence or absence of a particular direct 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 earlier example, inferring a relation based on the presence of an indirect path relies on the semantics of the links that make up the path, like transitivity or equivalence classes.

As an evaluation and a sample practical application, we apply our refinement framework to the task of refining the temporal precedence relations in VERBOCEAN, a broad-coverage noisy network of semantic relations between verbs extracted by mining the Web [2]. Examples of new edges discovered (added) by applying the
framework include: “ascertain happens-before evaluate”, “approve happens-before back”, “coat happens-before bake”, “plan happens-before complete”, and “interrogate happens-before extradite”.

Examples of edges that are removed by applying our framework include: “induce happens-before treat”, “warm happens-before heat”, “halve happens-before slice”, and “fly happens-before operate”.

Experiments show that our framework is particularly good at filtering out the incorrect temporal relations in VERBOCEAN. Removing incorrect relations is particularly important for inference systems.

2 VerbOcean

We apply our path-based refinement framework to VERBOCEAN [2], a web-extracted lexical semantics resource with potential applications to a variety of natural language tasks such as question answering, information retrieval, document summarization, and machine translation. VERBOCEAN is a graph of semantic relations between verbs, with 3,477 verbs (nodes) and 22,306 relations (edges). Although the framework applies whenever some paths through the graph imply presence or absence of a relation, for the evaluation we focus on the temporal precedence relation in VERBOCEAN, and, in an ancillary role, on the similarity relation. Senses are not discriminated and an edge indicates that the relation is believed to hold between some senses of the verbs in this relation.

The five semantic relations present in VERBOCEAN are presented in Table 1. Temporal precedence (happens-before) is a transitive asymmetric temporal relation between verbs. Similarity is a relation that suggests two nodes are likely to be in the same equivalence class, although polysemy makes it only weakly transitive.

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

| Semantic Relation  | Example       | Transitive | Symmetric | # in VERBOCEAN |
|--------------------|---------------|------------|-----------|----------------|
| temporal precedence| marry :: divorce | Y          | N         | 4,205          |
| similarity         | produce :: create | Y          | Y         | 11,515         |
| strength           | wound :: kill  | Y          | N         | 4,220          |
| antonymy           | open :: close  | N          | Y         | 1,973          |
| enablement         | fight :: win   | Y          | N         | 393            |

In VERBOCEAN, asymmetric relations between two nodes are enforced to be unidirectional (i.e., presence of an edge $x_i$ happens-before $x_j$ guarantees absence of an edge $x_j$ happens-before $x_i$). Larger, inconsistent loops are possible, however, as extraction is strictly local. Taking advantage of the global picture to refine the edges of the graph can improve quality of the resource, helping performance of any algorithms or applications that rely on the resource.
3 Global Refinement

Our approach relies on a global view of the graph to refine a relation between a given pair of nodes \( x_i \) and \( x_j \), based on multiple indirect paths between the two nodes. The analysis processes triples \(<x_i, r, x_j>\) for the relation \( r \) to output \( r \), its opposite (which we will denote \( q \)), or neither. The opposite of happens-before is the same relation in the reverse direction (happens-after). The refinement is based on evidence provided by indirect paths, over a probabilistic representation of the graph.

Section 3.1 introduces the steps of the refinement, Section 3.2 details which paths are used as evidence, and Section 3.3 derives the statistical model used for combining evidence from multiple unreliable paths.

3.1 Overview of the Refinement Algorithm

We first 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 (perhaps spuriously) the presence of the relation \( r \) between \( x_i \) and \( x_j \). Let \( r_{i,j} \) denote the relation \( r \) actually holding between \( x_i \) and \( x_j \). Let \( \psi_{i,j} \) 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 “\( x_1 \) similar \( x_2 \) happens-before \( x_3 \)” can be denoted \( \psi_{1,3} \). If the edges of \( \psi_{i,j} \) indicate the relation \( r \) between the nodes \( x_i \) and \( x_j \), we say that \( \psi_{i,j} \) indicates \( r_{i,j} \).

Given a triple \(<x_i, r, x_j>\), we identify the set \( \Psi_r^{full} \) of all paths \( \psi_{i,j} \) such that \( \psi_{i,j} \) indicates \( r_{i,j} \) and \( \psi_{i,j} \)'s sequence of relations \( \{R_{i,i+1} .. R_{j-1,j}\} \) matches one of the allowed sequences. That is, we only consider certain path types. The restriction on types of paths considered is introduced because identifying and processing all possible paths indicating \( r_{i,j} \) is too demanding computationally in a large non-sparse graph. The path types considered are detailed in Section 3.2. Note that the intermediate nodes of paths can range over the entire graph.

For each \( \psi_{i,j} \) in the above set \( \Psi_r^{full} \), we compute the estimated probability that \( r_{i,j} \) holds given the observation of (relations that make up) \( \psi_{i,j} \). Each edge in the input graph is treated as a probabilistic one, with probabilities \( P(r_{i,j}) \) and \( P(r_{i,j}|R_{i,j}) \) estimated from human judgments on a representative sample. Generally, longer paths and paths made up of less reliable edges will have lower probabilities. Section 3.3 presents the full model for estimating these probabilities.

Next, we form the set \( \Psi_r \) by selecting from \( \Psi_r^{full} \) only the paths which have no common intermediate nodes. This is done greedily, processing all paths in \( \Psi_r^{full} \) in order of decreasing score, placing each in \( \Psi_r \) iff it does not share any intermediate nodes with any path already in \( \Psi_r \). This is done to avoid double-counting the available evidence in our framework, which operates assuming conditional independence of paths.

Next, we compute \( P(r_{i,j}|\Psi_r) \), the probability of \( r_{i,j} \) given the evidence provided by the paths in \( \Psi_r \). The model for computing this is described in Section 3.3. Similarly, \( \Psi_q \) and \( P(q_{i,j}|\Psi_q) \) are computed for \( q_{i,j} \) the opposite of \( r_{i,j} \). Next, the evidence for \( r \) and \( q \) are reconciled by computing \( P(r_{i,j}|\Psi_r, \Psi_q) \) and, similarly, \( P(q_{i,j}|\Psi_r, \Psi_q) \).

Finally, the more probable of the two relations \( r_{i,j} \) and \( q_{i,j} \) is output if its probability exceeds a threshold value \( P_{min} \) (i.e., \( r_{i,j} \) is output if \( P(r_{i,j}|\Psi_r, \Psi_q) > P(q_{i,j}|\Psi_r, \Psi_q) \) and \( P(r_{i,j}|\Psi_r, \Psi_q) > P_{min} \)). In Section 4.2, we experiment with varying values of \( P_{min} \).
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 $\Psi_{i,j}$ 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. Also, we do not consider very long paths, as they tend to become unreliable due to accumulation of chance of false detection of each edge and sense drift in each intermediate node. 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.

For the presence of happens-before, a transitive asymmetric relation, we considered all 11 path types of length 3 or less which imply happens-before between the end nodes based on transitivity and equivalence:

- “happens-before”
- “happens-before, similar”
- “similar, happens-before”
- “happens-before, happens-before”
- “similar, happens-before, similar”
- “happens-before, similar, similar”
- “happens-before, happens-before, similar”
- “happens-before, similar, happens-before”
- “similar, happens-before, happens-before”
- “happens-before, happens-before, happens-before”

3.3 Statistical Model for Combining Evidence

This section presents a rigorous derivation of the probabilistic model for computing and combining probabilities with which indirect paths indicate a given edge.

3.3.1 Estimating from a Single Path

We first derive probability of $r_{1,n}$ given single path $\psi_{1,n}$:

$$P(r_{1,n} | \psi_{1,n})$$

If $n$ is 2, i.e. $\psi_{1,n}$ has only one edge $R_{1,2}$, we have simply the probability that the edge actually holds given its presence in the graph:

$$P(r_{1,2} | \psi_{1,2}) = P(r_{1,2} | R_{1,2})$$ (1)

Otherwise, $\psi_{1,n}$ has intermediate nodes, in which case $P(r_{1,n} | \psi_{1,n})$ can be estimated as follows:

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

$$P(r_{1,n} | R_{1,2}, ..., R_{n-1,n}, \neg(r_{1,2}, ..., r_{n-1,n}))P(\neg(r_{1,2}, ..., r_{n-1,n}) | R_{1,2}, ..., R_{n-1,n})$$

Because $r_{1,n}$ is conditionally independent from $R_{i,i+1}$ given $r_{i,i+1}$ or $\neg r_{i,i+1}$, we can simplify:

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

$$P(r_{1,n} | \neg(r_{1,2}, ..., R_{n-1,n}))P(\neg(r_{1,2}, ..., R_{n-1,n}) | R_{1,2}, ..., R_{n-1,n})$$

Assuming independence of a given relation $r_{i,i+1}$ from all edges in $\psi_{1,n}$ except for the edge $R_{i,i+1}$ yields:
Let $P_{\text{match}}$ denote the probability that there is no significant shift in meaning at a given intermediate node. Then, assume that path $r_{1,2}, \ldots, r_{n-1,n}$ indicates $r_{1,n}$ iff the meanings at $n-2$ intermediate nodes match:

$$P(r_{1,n} | r_{1,2}, \ldots, r_{n-1,n}) = P_{\text{match}}^{n-2}$$

Also, when one or more of the relations $r_{i,i+1}$ do not hold, nothing is generally implied about $r_{1,n}$, thus

$$P(r_{1,n} | -(r_{1,2}, \ldots, r_{n-1,n})) = P(r_{1,n})$$

Plugging these in, we have:

$$P(r_{1,n} | \psi_{1,n}) = P_{\text{match}}^{n-2} \prod_{i=1 \ldots n-1} P(r_{i+1} | R_{i,i+1}) + P(r_{1,n}) \left[ 1 - P_{\text{match}}^{n-2} \prod_{i=1 \ldots n-1} P(r_{i+1} | R_{i,i+1}) \right]$$

which can be rewritten as:

$$P(r_{1,n} | \psi_{1,n}) = P(r_{1,n}) + (1 - P(r_{1,n})) P_{\text{match}}^{n-2} \prod_{i=1 \ldots n-1} P(r_{i+1} | R_{i,i+1})$$

where the prior $P(r_{1,n})$ and the conditional $P(r_{i+1} | R_{i,i+1})$ can be estimated empirically by manually tagging the relations $R_{i,j}$ in a graph as correct or incorrect. $P(r_{1,n})$ is the probability that an edge will be labeled with relation $r$ by a human judge, and $P(r_{i+1} | R_{i,i+1})$ is the precision with which the system could identify $R$. While $P_{\text{match}}$ can be estimated empirically we have not done so. We experimentally set $P_{\text{match}} = 0.9$.

### 3.3.2 Combining Estimates from Multiple Paths

In this subsection we derive an estimate of the validity of inferring $r_{1,n}$ given the set $\Psi_r$ of $m$ paths $\psi_{1,n}^1, \psi_{1,n}^2, \ldots, \psi_{1,n}^m$:

$$P(r_{1,n} | \psi_{1,n}^1, \psi_{1,n}^2, \ldots, \psi_{1,n}^m)$$

In the case of zero paths, we use simply $P(r_{1,n})=P(r)$, the probability of observing $r$ between a pair of nodes from a sample set with no additional evidence. The case of one path has been treated in the previous section. In the case of multiple paths, we derive the expression as follows (omitting for convenience subscripts on paths, and distinguishing them by their superscripts). We assume conditional independence of any two paths $\psi^i$ and $\psi^j$ given $r$ or $\neg r$. Using Bayes’ rule yields:

$$P(r_{1,n} | \psi^1, \ldots, \psi^m) = \frac{P(r) P(\psi^1, \ldots, \psi^m | r)}{P(\psi^1, \ldots, \psi^m)} = \frac{P(r) \prod_{i=1 \ldots m} P(\psi^i | r)}{P(\psi^1, \ldots, \psi^m)}$$

---

1 This is not the case for paths in which the value of one edge, given the other edges, is correlated with the value of the end-to-end relation. The exception does not apply for happens-before edges if there are other happens-before edges in the path, nor does it ever apply for any similar edges.

2 Here and afterward, the denominators must be non-zero; they are always so when we apply this model.
The above denominator can be rewritten as:
\[
P(y^1, \ldots, y^n) = P(r)P(y^1 \mid r) + P(\neg r)P(y^1 \mid \neg r) = P(r) \prod_{k=1}^m P(y^k \mid r) + P(\neg r) \prod_{k=1}^m P(y^k \mid \neg r)
\]  
(5)

Using Bayes’ rule again, the expressions in the above products can be rewritten as follows:
\[
P(y^k \mid r) = \frac{P(r \mid y^k)P(y^k)}{P(r)}
\]  
(6)

\[
P(y^k \mid \neg r) = \frac{P(\neg r \mid y^k)P(y^k)}{P(\neg r)} = \frac{(1 - P(r \mid y^k))P(y^k)}{1 - P(r)}
\]  
(7)

Substituting into Eq. 5 the Eqs. 6 and 7 yields:
\[
P(y^1, \ldots, y^n) = P(r) \prod_{k=1}^m P(y^k \mid r) + P(\neg r) \prod_{k=1}^m P(y^k \mid \neg r) = P(r) \prod_{k=1}^m \left( \frac{P(r \mid y^k)P(y^k)}{P(r)} \right) + (1 - P(r)) \prod_{k=1}^m \left( \frac{(1 - P(r \mid y^k))P(y^k)}{1 - P(r)} \right)
\]

Using the above for the denominator of Eq. 4, using Eq. 6 in the numerator of Eq. 4, and simplifying, we have:
\[
P(r \mid y^1, \ldots, y^n) = \frac{P(r) \prod_{k=1}^m P(y^k \mid r)}{P(y^1, \ldots, y^n)} = \frac{\prod_{k=1}^m P(r \mid y^k)}{P(r) + \prod_{k=1}^m \left( \frac{(1 - P(r \mid y^k))}{1 - P(r)} \right)}
\]

which can be rewritten as
\[
P(r \mid y^1, \ldots, y^n) = \frac{\prod_{k=1}^m P(r \mid y^k)}{\prod_{k=1}^m P(r \mid y^k) + \prod_{k=1}^m \left( \frac{P(r)}{1 - P(r)} \right)}
\]  
(8)

where \(P(r \mid y^k)\) is as in Eq. 2 and \(P(r)\) can be estimated empirically.

### 3.3.3 Estimating from Supporting and Opposing Paths

Recall that \(q\) denotes the opposite of \(r\). The previous section has shown how to compute \(P(r \mid \Psi_r)\) and, similarly, \(P(q \mid \Psi_q)\). We now derive how to estimate \(r\) given both \(\Psi_r, \Psi_q\):
\[
P(r \mid \Psi_r, \Psi_q)
\]  
(9)

We assume that \(r\) and \(q\) are disjoint, \(P(r,q)=P(rq)=P(q|r)=0\). We also assume that \(q\) is conditionally independent from \(\Psi_r\) given \(\neg r\), i.e.,
\[ P(q \mid r, \Psi_r) = P(q \mid r) \] and \[ P(q \mid r, \Psi_q) = P(q \mid r, \Psi_q), \] and similarly
\[ P(r \mid q, \Psi_q) = P(r \mid q) \] and \[ P(r \mid q, \Psi_q) = P(r \mid q, \Psi_q). \]

We proceed by deriving the following, each consequent relying on the previous result:

**Lemma 1.** From \(P(r \mid q) = 0\), we observe:
\[ P(q) = P(r)P(q \mid r) + P(\neg r)P(q \mid r) = P(\neg r)P(q \mid r) \]

Solving for \(P(q \mid r)\), we obtain:
\[ P(q \mid r) = \frac{P(q)}{P(\neg r)} \quad (10) \]

**Lemma 2.** Using an approach similar to that of Lemma 1 and noting that \(P(q \mid r, \Psi r) = P(q \mid r) = 0\) yields:
\[ P(q \mid \Psi_r) = P(r \mid \Psi_r)P(q \mid r, \Psi_r) + P(\neg r \mid \Psi_r)P(q \mid r, \Psi_r) = 0 + P(\neg r \mid \Psi_r)P(q \mid r, \Psi_r) \]

Invoking the assumption \(P(q \mid r, \Psi_r) = P(q \mid r)\), we can simplify:
\[ P(q \mid \Psi_r) = P(\neg r \mid \Psi_r)P(q \mid r) \]

Substituting the result of Lemma 1 (Eq. 10) into the above yields:
\[ P(q \mid \Psi_r) = \frac{P(\neg r \mid \Psi_r)P(q)}{P(\neg r)} \quad (11) \]

And thus
\[ P(\neg q \mid \Psi_r) = \frac{P(\neg r) - P(\neg r \mid \Psi_r)P(q)}{P(\neg r)} \quad (12) \]

**Lemma 3.** We derive \(P(r \mid \neg q, \Psi_r)\), using \(P(\neg q \mid r, \Psi r) = 1\):
\[ P(r \mid \neg q, \Psi_r) = \frac{P(r, \neg q, \Psi_r)}{P(\neg q, \Psi_r)} = \frac{P(r, \neg q, \Psi_r)P(\Psi_r)}{P(\neg q, \Psi_r)P(\Psi_r)} = \frac{P(r \mid \Psi_r)}{P(\neg q \mid \Psi_r)} \]

Substituting the result of Lemma 2 (Eq. 12) into the above yields:
\[ P(r \mid \neg q, \Psi_r) = \frac{P(\neg r)P(r \mid \Psi_r)}{P(\neg r) - P(\neg r \mid \Psi_r)P(q)} \quad (13) \]

Similarly,
\[ P(q \mid \neg r, \Psi_q) = \frac{P(\neg q)P(q \mid \Psi_q)}{P(\neg q) - P(\neg q \mid \Psi_q)P(r)} \quad (14) \]
**Theorem 3**

\[
P(r \mid \Psi_r, \Psi_q) = \frac{P(-r)P(r \mid \Psi_r)P(-q \mid \Psi_q)}{(1 - P(r))(1 - P(q)) - (P(r \mid \Psi_r) - P(r))(P(q \mid \Psi_q) - P(q))}
\]

\(P(r \mid \Psi_r, \Psi_q)\) can be derived using the above Lemmas, as follows:

\[
P(r \mid \Psi_r, \Psi_q) = P(q \mid \Psi_r, \Psi_q)P(r \mid q, \Psi_r, \Psi_q) + P(-q \mid \Psi_r, \Psi_q)P(r \mid -q, \Psi_r, \Psi_q)
\]

The assumption \(P(r \mid q) = 0\) implies \(P(r \mid q, \Psi_r, \Psi_q) = 0\). Also, since \(r\) is conditionally independent of \(\Psi_q\) given \(\neg q\), we have \(P(r \mid -q, \Psi_r, \Psi_q) = P(r \mid -q, \Psi_r)\). Thus, we can simplify:

\[
P(r \mid \Psi_r, \Psi_q) = P(-q \mid \Psi_r, \Psi_q)P(r \mid -q, \Psi_r) = (1 - P(q \mid \Psi_r, \Psi_q))P(r \mid -q, \Psi_r)
\]

Similarly,

\[
P(q \mid \Psi_r, \Psi_q) = P(-r \mid \Psi_r, \Psi_q)P(q \mid -r, \Psi_r) = (1 - P(r \mid \Psi_r, \Psi_q))P(q \mid -r, \Psi_r)
\]

Substituting, Eq. 16 into Eq. 15 yields:

\[
P(r \mid \Psi_r, \Psi_q) = (1 - (1 - P(r \mid \Psi_r, \Psi_q))P(q \mid -r, \Psi_r))P(r \mid -q, \Psi_r)
\]

\[
= P(r \mid -q, \Psi_r)(1 - P(q \mid -r, \Psi_r)) + P(r \mid \Psi_r, \Psi_q)P(q \mid -r, \Psi_q)P(r \mid -q, \Psi_r)
\]

Solving for \(P(r \mid \Psi_r, \Psi_q)\), we get:

\[
P(r \mid \Psi_r, \Psi_q) = \frac{P(r \mid -q, \Psi_r) - P(r \mid -q, \Psi_r)P(q \mid -r, \Psi_q)}{1 - P(r \mid -q, \Psi_r)P(q \mid -r, \Psi_q)}
\]

Expanding and simplifying, we establish our Theorem 1:

\[
P(r \mid \Psi_r, \Psi_q) = \frac{P(-r)P(r \mid \Psi_r)P(-q \mid \Psi_q)}{(1 - P(r))(1 - P(q)) - (P(r \mid \Psi_r) - P(r))(P(q \mid \Psi_q) - P(q))}
\]

### 4 Experimental Results

In this section, we evaluate our refinement framework on the temporal precedence relations discovered by VERBOCEAN, and present some observations on applying the refinement to other VERBOCEAN relations.

#### 4.1 Experimental Setup

Following Chklovski and Pantel [2], we studied 29,165 pairs of verbs obtained from a paraphrasing algorithm called DIRT [4]. We applied VERBOCEAN to the 29,165 verb pairs, which tagged each pair with the semantic tag *happens-before*, *happens-after* and *no temporal precedence*.

VERBOCEAN actually produces additional relations such as *similarity*, *antonymy*, *strength* and *enablement*. For our purposes, we only consider the temporal relations.
For our experiments, we randomly sampled 1000 of these verb pairs, and presented them to two human judges (without revealing the VERBOCEAN tag). The judges were asked to classify each pair among the following tags:

- Happens-before with entailment
- Happens-before without entailment
- Happens-after with entailment
- Happens-after without entailment
- Another semantic relation
- No semantic relation

For the purposes of our evaluation, tags \( a \) and \( b \) align with VERBOCEAN’s happens-before tag, tags \( c \) and \( d \) align with the happens-after tag, and tags \( e \) and \( f \) align with the no temporal relation tag\(^4\). The Kappa statistic \([7]\) for the task was \( \kappa = 0.78 \).

### 4.2 Refinement Results

Table 2 shows the overall accuracy of VERBOCEAN tags on the 1000 verb pairs randomly sampled from DIRT. Each row represents a different refinement. The number in parentheses is \( P_{min} \), the threshold value for the strength of the relation from Section 3.1. As the threshold is increased, the refinement algorithm requires greater evidence (more supporting paths and absence of opposing evidence) to trigger a temporal relation between a pair of verbs.

Table 2. Accuracy (95\% confidence) of VERBOCEAN on a random sample of 1000 verb pairs tagged by two judges

|                      | Accuracy       |
|----------------------|----------------|
|                      | Judge1 | Judge2 | Total    |
| Unrefined            | 80.7\% | 74.8\% | 77.7\% ± 2.0\% |
| Refined (0.5)        | 66.0\% | 63.7\% | 64.8\% ± 2.6\% |
| Refined (0.66)       | 75.4\% | 71.7\% | 73.5\% ± 2.4\% |
| Refined (0.9)        | 83.1\% | 77.2\% | 80.2\% ± 2.1\% |
| Refined (0.95)       | 84.5\% | 78.0\% | 81.3\% ± 1.9\% |
| Refined (Combo)*     | 86.8\% | 81.3\% | 84.0\% ± 2.4\% |

* Combo combines the no temporal relation from the 0.5 and the happens-before and happens-after from the 0.9 and 0.95 refinements, where the reported accuracy is computed on the subset of 716 verb pairs for which the algorithm is most confident.

Table 3 shows the reassignments due to refinement. At the 0.5 level, the refinement left 76 of 81 relations unchanged, revising 3 to happens-after and 2 to no temporal relation. Similarly, only two of the original happens-after relations were changed with refinement. However, of the 849 originally tagged no temporal relation, the

\(^4\) In future work, we plan to use the judges’ classifications to evaluate the extraction of entailment relations using VERBOCEAN.
Table 3. Allocation change between semantic tags due to refinement

|                  | Happens-Before | Happens-After | No Temporal Relation |
|------------------|----------------|--------------|----------------------|
| Unrefined        | 81             | 70           | 849                  |
| Refined (0.5)    | 190            | 180          | 630                  |
| Refined (0.66)   | 118            | 124          | 758                  |
| Refined (0.9)    | 53             | 66           | 881                  |
| Refined (0.95)   | 40             | 46           | 914                  |

Fig. 1. Refinement precision on each semantic tag

Refinement moved 113 to happens-before and 109 to happens-after. The precision of the 0.5 refinement on the no temporal relation tag increased by 4%; however, the precision on the temporal relations decreased by 5.7%. At the 0.95 refinement level, 54 of the 81 relations originally tagged happens-before and 45 of the 70 relations originally tagged happens-after were changed to no temporal relation. Only 34 of the 849 no temporal relations were changed. At this level, the precision of no temporal relation tag decreased by 0.8% and the temporal relations’ precision increased by 4%.

Hence, at the 0.5 level, pairs classified as no temporal relation were improved while at the 0.95 level, pairs classified as a temporal relation were improved. To leverage benefits of the two, we applied both the 0.5 and 0.95 level refinements and kept happens-before and happens-after classifications from the 0.95 level, and kept the no temporal relation classification from the 0.5 level. 5 284 verb pairs were left unclassified. On the 716 classified verb pairs, refinement improved accuracy by 6.3%.

Fig. 2. Refinement recall on each semantic tag

5 This combination is guaranteed to be free of conflicts in classification because it is impossible for a relation to be classified as temporal at the 0.95 threshold level while being classified as non-temporal at the 0.5 level.
Figures 1 and 2 illustrate the refinement precision and recall for each semantic tag. Both annotators have agreed on 819 verb pairs, and we examined performance on these. Figure 3 shows a higher precision on these pairs as compared to the overall set, illustrating that what is easier for the annotators is easier for the system.

### 4.3 Observations on Refining Other Relations

We have briefly investigated refining other semantic relations in VERBOCEAN. The extent of the evaluation was limited by availability of human judgments. We randomly sampled 100 pairs from DIRT and presented the classifications to three human judges for evaluation [2].

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 4 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.

| Verb 1 | Verb 2 | VERBOCEAN Relation | Refinement Relation | Judge 1 Relation | Judge 2 Relation | Judge 3 Relation |
|--------|--------|---------------------|--------------------|------------------|------------------|------------------|
| attach | use    | happens-before      | similar            | none             | none             | none             |
| bounce | get    | weaker than         | 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   |

* only revision of relation to its opposite or “none” was attempted here

### 4.4 Discussion

Our evaluation focused on the presence or absence of relations after refinement, without exploiting the fact that our framework also updates confidences in a given relation. The additional information about confidence can benefit probabilistic inference approaches (e.g., [3]).

Possible extensions to the algorithm include a more elaborate inference from graph structure, for example treating the absence of certain paths as counter-evidence. Suppose that relations A happens-before B and A similar A’ were detected, but the relation A’ happens-before B was not. Then, the absence of a path
A similar A' happens-before B suggests the absence of A happens-before B.

Other important avenues of future work include applying our framework to other relations (e.g., strength in VERBOCEAN) and to better characterize the refinement thresholds.

5 Conclusions

We presented a method for refining edges in graphs by leveraging the semantics of multiple noisy paths. We re-estimated the presence of an edge 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. We applied our model to refining temporal verb relations in a semantic resource called VERBOCEAN. Experiments showed a 16.1% error reduction after refinement. On the 72% refinement decisions that it was most confident, the error reduction was 28.3%.

The usefulness of a semantic resource is highly dependent on its quality, which is often poor in automatically mined resources. With graph refinement frameworks such as the one presented here, many of these resources may be improved automatically.

References

1. Berland, M. and E. Charniak, 1999. Finding parts in very large corpora. In *ACL-1999*. pp. 57-64. College Park, MD.
2. Chklovski, T., and Pantel, P. 2004. 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.
3. Domingos, P. and Richardson, M. 2004. Markov Logic: A unifying framework for statistical relational learning. In *Proceedings of ICML Workshop on Statistical Relational Learning and its Connections to Other Fields*. Banff, Canada.
4. Lin, D. and Pantel, P. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343-360.
5. Pantel, P. and Ravichandran, D. 2004. Automatically labeling semantic classes. In *Proceedings HLT/NAACL-04*. pp. 321-328. Boston, MA.
6. Shinzato, K. and Torisawa, K. 2004. Acquiring hyponymy relations from web documents. In *Proceedings of HLT-NAACL-2004*. pp. 73-80. Boston, MA.
7. Siegel, S. and Castellan Jr., N. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill.