Robust Textual Inference using Diverse Knowledge Sources

Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova
Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

We present a machine learning approach to robust textual inference, in which parses of the text and the hypothesis sentences are used to measure their asymmetric "similarity", and thereby to decide if the hypothesis can be inferred. This idea is realized in two different ways. In the first, each sentence is represented as a graph (extracted from a dependency parser) in which the nodes are words/phrases, and the links represent dependencies. A learned, asymmetric, graph-matching cost is then computed to measure the similarity between the text and the hypothesis. In the second approach, the text and the hypothesis are parsed into the logical formula-like representation used by (Harabagiu et al., 2000). An abductive theorem prover (using learned costs for making different types of assumptions in the proof) is then applied to try to infer the hypothesis from the text, and the total "cost" of proving the hypothesis is used to decide if the hypothesis is entailed.

1 Introduction

Below, we illustrate our methods with the following toy example of entailment:

```
TEXT: Chris purchased a BMW.
HYPOTHESIS: Chris bought a car.
```

Using relationships derived from syntactic dependencies, we can represent the text and hypothesis sentences equivalently as either a directed graph, or as a set of logical terms, as shown in Figure 1 and Section 3.1. In the graph, a vertex typically represents a word, but can also represent a phrase that is interpreted as a single entity. Labeled edges represent syntactic and semantic relationships tagged by various modules. The logical formula is derived by constructing a term for each node in the graph, and representing the dependency links with appropriately shared arguments. After presenting the inference methods, we show how the representations over which they work are derived from plain text.

Figure 1: Example graph matching ($\alpha = 0.55$) for example pair in Section 2. Dashed lines represent mapping.

2 Entailment by graph matching

We take the view that a hypothesis can be inferred from the text when the cost of matching the hypothesis graph to the text graph is low. For the remainder of this section, we outline a model for assigning a match cost to graphs.

For hypothesis graph H, and text graph T, a *matching* M is a mapping from the vertices of H to those of T; we allow nodes in H to map to a fictitious NIL vertex if necessary. Suppose the cost of matching M is $\text{Cost}(M)$. Then we define the cost of matching H to T: $\text{MatchCost}(H,T) = \min_M \text{Cost}(M)$.

One simple cost model is given by the normalized sum of costs $\text{SubCost}(v, M(v))$ for substituting each vertex v in H for $M(v)$ in T:

$$\text{Cost}(M) = \frac{1}{Z} \sum_{v \in H_V} w(v)\ \text{SubCost}(v, M(v)) \qquad (1)$$

Here, $w(v)$ represents the weight or relative importance for vertex v, and $Z = \sum w(v)$ is a normalization constant. In our implementation, the weight of each vertex was based on the part-of-speech tag of the word or the type of named entity, if applicable. For hypothesis vertex v and text vertex $M(v)$, the substitution cost (in $[0, 1]$) is progressively higher for the following conditions:

- v and $M(v)$'s stem and POS / only stem match
- v is a synonym / hypernym of $M(v)$ (*WordNet*)
- v and $M(v)$'s stems are similar according to the word similarity modules (described later).

As (Punyakanok et al., 2004) demonstrated, models which also match syntactic relationships between words can outperform bag-of-words models for TREC QA answer extraction. As in (1), we can measure how relationally similar H and T are by a normalized sum of costs for substituting each edge relation (v, v') in H with the edge relation $(M(v), M(v'))$ in T. We assign a substitution cost for edge (v, v') in H based on the following conditions on path length:

- $M(v)$ is a parent/ancestor of $M(v')$
- $M(v)$ and $M(v')$ share a parent/ancestor

As in the vertex case we have weights for each hypothesis edge, $w(e)$, based upon the edge's label; typically subject and object relations are more important to match than others. Our final matching cost is given by a convex mixture of the vertex and relational match costs:
$\text{Cost}(M) = \alpha\text{VertexCost}(M) + (1-\alpha)\text{RelationCost}(M)$.

Notice that minimizing $\text{Cost}(M)$ is computationally hard since $\text{RelationCost}(M) = 0$ if and only if H is isomorphic to a subgraph of T. As an approximation, we can efficiently find the matching M^* which minimizes $\text{VertexCost}(\cdot)$ using the Hungarian method (Kuhn, 1955); we then perform local greedy hillclimbing search, beginning from M^*, to approximate the minimal matching.

3 Abductive theorem proving

This method works with a logical formula-like representation (Harabagiu et al., 2000) of the syntactic dependencies in the text and hypothesis sentences. The basic idea is that a hypothesis that can be logically derived from the text is entailed by it. Such a logical derivation is called a "proof" of the hypothesis.

The logical formulae capture only the syntactic dependencies in the sentences. Consequently, several entailed hypotheses that require semantic rewrites (such as "a *BMW* is a *car*") can be derived from the corresponding text formulae only by using additional assumptions in the proof. We do not use explicit logical axioms ("rules") for these assumptions; instead, each assumption that unifies one term in the hypothesis with another in the text is assigned a cost based on the judged plausibility of that assumption. This cost is computed using particular features of the assumption.

Using such a cost model, the inference procedure searches for a minimum cost proof for the hypothesis. The hypothesis is judged to be entailed from the text if it has a proof with cost below a certain learned threshold value.

We also provide a procedure to learn good costs for assumptions from a training set containing examples of entailed and non-entailed hypotheses.

3.1 Representation

For the example, the following logical representation is derived, with each number/letter representing a constant:

```
T: Chris(1) BMW(2) purchased(3,1,2)
H: Chris(x) car(y) bought(e,x,y)
```

Each predicate and each argument is also annotated with other linguistic information not shown here (such as semantic roles and named entity tags) for use in assigning costs to assumptions.

3.2 Inference

For our representation, proof steps that unify one term from the text with one term of the hypothesis suffice. We allow any pair of terms to unify with each other, with a cost assigned by the *assumption cost model*. We relax the requirements for logical unification in several ways, adding cost penalties for each such relaxation:

1. Terms with different predicates can be unified; the cost penalty is obtained using the term similarity measures (described later) and the linguistic annotations on the predicates.
2. The terms can have differing number of arguments, and the arguments of one term can be matched with those of the other term in any order. Each argument matching is assigned a cost based on the compatibility of the annotations of those arguments. A term pair might be unified in many ways corresponding to different argument matchings.
3. Constants can be unified with each other at an appropriate cost. This cost is precomputed for all constant pairs in a particular example, and is lowered for specific pairs—such as when there is possible coreference or appositive reference.

We developed a specialized abductive theorem prover to discover the minimum cost proof using uniform cost search. For our running example, the minimum cost proof unifies `BMW` with `car`, and `purchased` with `bought`, at small costs.

3.3 Learning good costs for assumptions

Given a training set of labeled text-hypothesis pairs (such as the RTE development set), we propose a learning algorithm that tries to learn good assumption costs.[1]

4 Producing representations and similarities for inference

4.1 Syntactic processing

The first steps of the front-end deal with tokenization and parsing. Beyond this base level, the performance of the inference methods depends critically on our ability to identify similarities and differences between our fairly syntactic representations of the text and the hypothesis. This is largely dependent on being able to perform normalization and enrichment tasks that will reveal essential similarities, and on having good measures of lexical semantic similarity between words and larger units.

---

[1] Details are omitted here due to space constraints. See (Raina et al., 2005) for details.

Dataset	`General`		`ByTask`	
	Accuracy	CWS	Accuracy	CWS
DevSet1	64.8%	0.778	65.5%	0.805
DevSet2	52.1%	0.578	55.7%	0.661
DevSet1 + DevSet2	58.5%	0.679	60.8%	0.743
Test set	56.2%	0.620	55.2%	0.686

Table 1: Accuracy and confidence weighted score (CWS) on RTE datasets.

We do deterministic tokenization and then use full sentence parsing to reveal syntactic dependencies. The parser used was a variant of (Klein and Manning, 2003). The most important addition was training on an extra dozen sentences that gave the parser some exposure to topics in the news in 2005 rather than only those appearing in 1989. Exploiting headedness relations and hand-written pattern-matching rules, the parse tree is converted into a set of typed dependencies between words, representing grammatical relations (like subject and object) and other modifier dependencies, including such things as appositives, negations, and temporal modifiers. This is the basis of the graph structure in Figure 1. Various collapsings are then done to normalize and improve this dependency representation. Prepositions and possessive *'s* are changed from being vertices to relation names, and coordinations explicitly represent the conjuncts. A conditional random field (Lafferty et al., 2001) named entity recognition system is run to identify seven classes (Person, Organization, Location; Percent, Time, Money, and Date). The first three are collapsed into single nodes tagged NNP (proper noun) prior to parsing, while the latter four are grouped after parsing, but before the conversion to a dependency representation, and their values are normalized into a canonical form using hand-written regular expressions. This includes representing approximate and relative quantities (*around $40* and *less than 2 dollars*) as well as exact amounts. At the same time, we also collapse collocations, which are found in WordNet, like *back off* and *throw up* to a single node.

4.2 Additional dependencies between nodes

We augment the syntactic dependency graph with semantic role arcs using a Propbank-trained semantic role labeler (Toutanova et al., 2005). For each verb, we added edges between that verb and the head word of each of its arguments, and labeled the edges with the appropriate semantic role. This allowed us to add relations (between words) that were not captured by surface syntax, and also to classify modifying phrases as temporal, locative, and other categories. We added coreference relations between noun phrases and named entities using a maximum entropy coreference classifier modeled after (Soon et al., 2001).

Task	`General`		`ByTask`	
	Accuracy	CWS	Accuracy	CWS
CD	79.3%	0.903	84.0%	0.926
IE	47.5%	0.493	55.0%	0.590
IR	56.7%	0.590	55.6%	0.604
MT	46.7%	0.480	47.5%	0.479
PP	58.0%	0.623	54.0%	0.535
QA	48.5%	0.478	43.9%	0.466
RC	52.9%	0.523	50.7%	0.480

Table 2: Accuracy and confidence weighted score (CWS) split by task on the RTE test set.

4.3 Methods for discovering term similarity

As in other work, e.g., (Moldovan et al., 2000), we relied on WordNet (Miller, 1995) heavily for lexical knowledge. The `WordNet::Similarity` module (Pedersen et al., 2004) was used to compute a symmetric similarity score between two phrases. If the queried phrases are listed as antonyms in WordNet, the match is given a very high cost in the inference procedures. Derivational forms in WordNet are used to detect nominalized events and modify the representation (e.g., *murder of police officer* entails *police officer killed*). WordNet does not include prepositions. We semi-automatically constructed a matrix of preposition similarity values using synonyms (e.g., *over* and *above*) and antonyms (e.g., *over* and *under*). Synonyms were found by grouping prepositions into clusters. Antonym pairs were added manually. Finally, we compiled a list of 206 countries and their derivatives manually (e.g., *Philippines - Filipino*), and collected a list of 276 frequently occurring acronyms in a large corpus, and recorded their expansions.

The inference procedures require considerable semantic knowledge to infer some rewrites using just phrasal dependencies; for example, *won victory in Presidential election* might entail *became President*. We attempted to discover such rewrites by looking for similarly placed phrases in a large corpus, using a backed-off modification of the similarity measure described in (Lin and Pantel, 2001).

Sometimes both of these methods are too precise. Words that are used in the same context often do not have explicit relationships between them; for instance *marathon* and *run* clearly have a semantic relationship not considered in the WordNet hierarchy. To overcome this we used `Infomap`,[2] an open-source implementation of Latent Semantic Analysis (Deerwester et al., 1990), to score words according to distributional similarity (measured using the British National Corpus). To further exploit distributional similarity, we also implemented a measure of similarity that is computed as the ratio between the number of search results from `google.com` for two phrases when queried separately and in combination.

---

[2] Available at `http://infomap.stanford.edu`.

Text	Hypothesis	Our answer	Conf	Comments
A Filipino hostage in Iraq was released.	A Filipino hostage was freed in Iraq. *(TRUE)*	True	0.61	Verb rewrite is handled. Phrasal ordering does not affect cost.
The government announced last week that it plans to raise oil prices.	Oil prices drop. *(FALSE)*	False	0.69	High cost given for substituting word for its antonym.
Shrek 2 rang up $92 million.	Shrek 2 earned $92 million. *(TRUE)*	False	0.51	Collocation “rang up” is not known to be similar to “earned”.
Sonia Gandhi can be defeated in the next elections in India by BJP.	Sonia Gandhi is defeated by BJP. *(FALSE)*	True	0.66	“can be” does not indicate the complement event occurs.
Fighters loyal to Moqtada al-Sadr shot down a U.S. helicopter Thursday in the holy city of Najaf.	Fighters loyal to Moqtada al-Sadr shot down Najaf. *(FALSE)*	True	0.67	Should recognize non-Location cannot be substituted for Location.
C and D Technologies announced that it has closed the acquisition of Datel, Inc.	Datel Acquired C and D technologies. *(FALSE)*	True	0.59	Failed to penalize switch in semantic role structure enough

Table 3: Analysis of results on some RTE examples.

5 Results and analysis

Our overall system is a combination of the two systems described in Sections 2 and 3. Each system produces a real number score that is normalized to have zero mean and unit variance, and then converted to a confidence value using the cumulative distribution function for a normal distribution. These individual scores are then linearly combined using logistic regression, with the weights trained on the RTE development sets. The first version (called `General`) trained one set of weights for all RTE tasks; the second version (called `ByTask`) trained separate weights per task. All parameters except the classifier weights were identical.

Table 1 reports the performance of our final classifiers on different datasets. Table 2 shows the performance separately on each task in the test set.

A random guessing baseline achieves accuracy 50% and confidence weighted score (CWS) 0.50. Our test set accuracy is only a few points above random guessing; however, the CWS is significantly higher. Thus, our predictions are well-calibrated and more robust; this is probably because our learning and classifier combination procedures maximize the likelihood of the full predicted distribution rather than just a binary accuracy value.

Table 3 has an analysis of some examples from the RTE datasets. The term similarity routines seemed most important for good performance, while many of the other modules are useful in specific cases. Many of the language resources used were sparse (e.g., antonyms in WordNet); high-recall resources would be extremely beneficial.

Acknowledgments

This work was supported by the Advanced Research and Development Activity (ARDA)’s Advanced Question Answering for Intelligence (AQUAINT) program.

References

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

S. Harabagiu, M. Pasca, and S. Maiorano. 2000. Experiments with Open-Domain Textual Question Answering. *COLING 2000*.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. *ACL-2003*, 423–430.

H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML-2001*.

D. Lin and P. Pantel. 2001. DIRT: Discovery of Inference Rules from Text. *KDD 2001*.

G. Miller. 1995 WordNet: A lexical database. *Communications of the ACM*, 38(11):39-41.

D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. 2000 The structure and performance of an open-domain question answering system. *ACL-2000*, 563–570.

T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::Similarity - Measuring the relatedness of concepts. *AAAI-2004*.

V. Punyakanok, D. Roth, and W. Yih. 2004 Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics*.

R. Raina, A. Y. Ng, and C. D. Manning 2005 Robust textual inference via abduction and learning. (Unpublished manuscript).

W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

K. Toutanova, A. Haghighi, and C. D. Manning 2005. Joint learning improves semantic role labeling. *ACL-2005*.