# Learning to recognize features of valid textual entailments

**Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe,**
**Daniel Cer, and Christopher D. Manning**
Computer Science Department
Stanford University
Stanford, CA 94305
{wcmac, grenager, mcdm, cerd, manning}@cs.stanford.edu

## Abstract

This paper advocates a new architecture for textual inference in which finding a good alignment is separated from evaluating entailment. Current approaches to semantic inference in question answering and textual entailment have approximated the entailment problem as that of computing the best alignment of the hypothesis to the text, using a locally decomposable matching score. We argue that there are significant weaknesses in this approach, including flawed assumptions of monotonicity and locality. Instead we propose a pipelined approach where alignment is followed by a classification step, in which we extract features representing high-level characteristics of the entailment problem, and pass the resulting feature vector to a statistical classifier trained on development data. We report results on data from the 2005 Pascal RTE Challenge which surpass previously reported results for alignment-based systems.

## 1 Introduction

During the last five years there has been a surge in work which aims to provide robust textual inference in arbitrary domains about which the system has no expertise. The best-known such work has occurred within the field of question answering (Pasca and Harabagiu, 2001; Moldovan et al., 2003); more recently, such work has continued with greater focus in addressing the PASCAL Recognizing Textual Entailment (RTE) Challenge (Dagan et al., 2005) and within the U.S. Government AQUAINT program. Substantive progress on this task is key to many text and natural language applications. If one could tell that *Protestors chanted slogans opposing a free trade agreement* was a match for *people demonstrating against free trade*, then one could offer a form of semantic search not available with current keyword-based search. Even greater benefits would flow to richer and more semantically complex NLP tasks.

Because full, accurate, open-domain natural language understanding lies far beyond current capabilities, nearly all efforts in this area have sought to extract the maximum mileage from quite limited semantic representations. Some have used simple measures of semantic overlap, but the more interesting work has largely converged on a graph-alignment approach, operating on semantic graphs derived from syntactic dependency parses, and using a locally-decomposable alignment score as a proxy for strength of entailment. (Below, we argue that even approaches relying on weighted abduction may be seen in this light.) In this paper, we highlight the fundamental semantic limitations of this type of approach, and advocate a multi-stage architecture that addresses these limitations. The three key limitations are an *assumption of monotonicity*, an *assumption of locality*, and a *confounding of alignment and evaluation of entailment*.

We focus on the PASCAL RTE data, examples from which are shown in table 1. This data set contains pairs consisting of a short text followed by a one-sentence hypothesis. The goal is to say whether the hypothesis follows from the text and general background knowledge, according to the intuitions of an intelligent human reader. That is, the standard is not whether the hypothesis is logically entailed, but whether it can reasonably be inferred.

## 2 Approaching a robust semantics

In this section we try to give a unifying overview to current work on robust textual inference, to present fundamental limitations of current methods, and then to outline our approach to resolving them. Nearly all current textual inference systems use a single-stage matching/proof process, and differ

| ID | Text | Hypothesis | Entailed |
|---|---|---|---|
| 59 | Two Turkish engineers and an Afghan translator kidnapped in December were freed Friday. | translator kidnapped in Iraq | no |
| 98 | Sharon warns Arafat could be targeted for assassination. | prime minister targeted for assassination | no |
| 152 | Twenty-five of the dead were members of the law enforcement agencies and the rest of the 67 were civilians. | 25 of the dead were civilians. | no |
| 231 | The memorandum noted the United Nations estimated that 2.5 million to 3.5 million people died of AIDS last year. | Over 2 million people died of AIDS last year. | yes |
| 971 | Mitsubishi Motors Corp.'s new vehicle sales in the US fell 46 percent in June. | Mitsubishi sales rose 46 percent. | no |
| 1806 | Vanunu, 49, was abducted by Israeli agents and convicted of treason in 1986 after discussing his work as a mid-level Dimona technician with Britain's Sunday Times newspaper. | Vanunu's disclosures in 1968 led experts to conclude that Israel has a stockpile of nuclear warheads. | no |
| 2081 | The main race track in Qatar is located in Shahaniya, on the Dukhan Road. | Qatar is located in Shahaniya. | no |

Table 1: Illustrative examples from the PASCAL RTE data set, available at *http://www.pascal-network.org/Challenges/RTE*. Though most problems shown have answer *no*, the data set is actually balanced between *yes* and *no*.

mainly in the sophistication of the matching stage. The simplest approach is to base the entailment prediction on the degree of semantic overlap between the text and hypothesis using models based on bags of words, bags of $n$-grams, TF-IDF scores, or something similar (Jijkoun and de Rijke, 2005). Such models have serious limitations: semantic overlap is typically a symmetric relation, whereas entailment is clearly not, and, because overlap models do not account for syntactic or semantic structure, they are easily fooled by examples like ID 2081.

A more structured approach is to formulate the entailment prediction as a graph matching problem (Haghighi et al., 2005; de Salvo Braz et al., 2005). In this formulation, sentences are represented as normalized syntactic dependency graphs (like the one shown in figure 1) and entailment is approximated with an alignment between the graph representing the hypothesis and a portion of the corresponding graph(s) representing the text. Each possible alignment of the graphs has an associated score, and the score of the best alignment is used as an approximation to the strength of the entailment: a better-aligned hypothesis is assumed to be more likely to be entailed. To enable incremental search, alignment scores are usually factored as a combination of local terms, corresponding to the nodes and edges of the two graphs. Unfortunately, even with factored scores the problem of finding the best alignment of two graphs is NP-complete, so exact computation is intractable. Authors have proposed a variety of approximate search techniques. Haghighi et al. (2005) divide the search into two steps: in the first step they consider node scores only, which relaxes the problem to a weighted bipartite graph matching that can be solved in polynomial time, and in the second step they add the edges scores and hillclimb the alignment via an approximate local search.

A third approach, exemplified by Moldovan et al. (2003) and Raina et al. (2005), is to translate dependency parses into neo-Davidsonian-style quasi-logical forms, and to perform weighted abductive theorem proving in the tradition of (Hobbs et al., 1988). Unless supplemented with a knowledge base, this approach is actually isomorphic to the graph matching approach. For example, the graph in figure 1 might generate the quasi-LF *rose(e1), nsubj(e1, x1), sales(x1), nn(x1, x2), Mitsubishi(x2), dobj(e1, x3), percent(x3), num(x3, x4), 46(x4)*. There is a term corresponding to each node and arc, and the resolution steps at the core of weighted abduction theorem proving consider matching an individual node of the hypothesis (e.g. *rose(e1)*) with something from the text (e.g. *fell(e1)*), just as in the graph-matching approach. The two models become distinct when there is a good supply of additional linguistic and world knowledge axioms—as in Moldovan et al. (2003) but not Raina et al. (2005). Then the theorem prover may generate intermediate forms in the proof, but, nevertheless, individual terms are resolved locally without reference to global context.

Finally, a few efforts (Akhmatova, 2005; Fowler et al., 2005; Bos and Markert, 2005) have tried to translate sentences into formulas of first-order logic, in order to test logical entailment with a theorem prover. While in principle this approach does not suffer from the limitations we describe below, in practice it has not borne much fruit. Because few problem sentences can be accurately translated to logical form, and because logical entailment is a strict standard, recall tends to be poor.

The simple graph matching formulation of the problem belies three important issues. First, the above systems assume a form of upward monotonicity: if a good match is found with a part of the text, other material in the text is assumed not to affect the validity of the match. But many situations lack this upward monotone character. Consider variants on ID 98. Suppose the hypothesis were *Arafat targeted for assassination*. This would allow a perfect graph match or zero-cost weighted abductive proof, because the hypothesis is a subgraph of the text. However, this would be incorrect because it ignores the modal operator *could*. Information that changes the validity of a proof can also exist outside a matching clause. Consider the alternate text *Sharon denies Arafat is targeted for assassination*.[1]

The second issue is the assumption of locality. Locality is needed to allow practical search, but many entailment decisions rely on global features of the alignment, and thus do not naturally factor by nodes and edges. To take just one example, dropping a restrictive modifier preserves entailment in a positive context, but not in a negative one. For example, *Dogs barked loudly* entails *Dogs barked*, but *No dogs barked loudly* does not entail *No dogs barked*. These more global phenomena cannot be modeled with a factored alignment score.

The last issue arising in the graph matching approaches is the inherent confounding of alignment and entailment determination. The way to show that one graph element does not follow from another is to make the cost of aligning them high. However, since we are embedded in a search for the lowest cost alignment, this will just cause the system to choose an alternate alignment rather than recognizing a non-entailment. In ID 152, we would like the hypothesis to align with the first part of the text, to be able to prove that civilians are not members of law enforcement agencies and conclude that the hypothesis does not follow from the text. But a graph-matching system will to try to get non-entailment by making the matching cost between *civilians* and *members of law enforcement agencies* be very high. However, the likely result of that is that the final part of the hypothesis will align with *were civilians* at the end of the text, assuming that we allow an alignment with "loose" arc correspondence.[2] Under this candidate alignment, the lexical alignments are perfect, and the only imperfect alignment is the subject arc of *were* is mismatched in the two. A robust inference guesser will still likely conclude that there is entailment.

We propose that all three problems can be resolved in a two-stage architecture, where the alignment phase is followed by a separate phase of entailment determination. Although developed independently, the same division between alignment and classification has also been proposed by Marsi and Krahmer (2005), whose textual system is developed and evaluated on parallel translations into Dutch. Their classification phase features an output space of five semantic relations, and performs well at distinguishing entailing sentence pairs.

Finding aligned content can be done by any search procedure. Compared to previous work, we emphasize structural alignment, and seek to ignore issues like polarity and quantity, which can be left to a subsequent entailment decision. For example, the scoring function is designed to encourage antonym matches, and ignore the negation of verb predicates. The ideas clearly generalize to evaluating several alignments, but we have so far worked with just the one-best alignment. Given a good alignment, the determination of entailment reduces to a simple classification decision. The classifier is built over features designed to recognize patterns of valid and invalid inference. Weights for the features can be hand-set or chosen to minimize a relevant loss function on training data using standard techniques from machine learning. Because we already have a complete alignment, the classifier's decision can be con-

[1] This is the same problem labeled and addressed as *context* in Tatu and Moldovan (2005).

[2] Robust systems need to allow matches with imperfect arc correspondence. For instance, given *Bill went to Lyons to study French farming practices*, we would like to be able to conclude that *Bill studied French farming* despite the structural mismatch.

ditioned on arbitrary *global* features of the aligned graphs, and it can detect failures of monotonicity.

## 3 System

Our system has three stages: linguistic analysis, alignment, and entailment determination.

### 3.1 Linguistic analysis

Our goal in this stage is to compute linguistic representations of the text and hypothesis that contain as much information as possible about their semantic content. We use *typed dependency graphs*, which contain a node for each word and labeled edges representing the grammatical relations between words. Figure 1 gives the typed dependency graph for ID 971. This representation contains much of the information about words and relations between them, and is relatively easy to compute from a syntactic parse. However many semantic phenomena are not represented properly; particularly egregious is the inability to represent quantification and modality.

We parse input sentences to phrase structure trees using the Stanford parser (Klein and Manning, 2003), a statistical syntactic parser trained on the Penn TreeBank. To ensure correct parsing, we preprocess the sentences to collapse named entities into new dedicated tokens. Named entities are identified by a CRF-based NER system, similar to that described in (McCallum and Li, 2003). After parsing, contiguous collocations which appear in WordNet (Fellbaum, 1998) are identified and grouped.

We convert the phrase structure trees to typed dependency graphs using a set of deterministic hand-coded rules (de Marneffe et al., 2006). In these rules, heads of constituents are first identified using a modified version of the Collins head rules that favor semantic heads (such as lexical verbs rather than auxiliaries), and dependents of heads are typed using *tregex* patterns (Levy and Andrew, 2006), an extension of the *tgrep* pattern language. The nodes in the final graph are then annotated with their associated word, part-of-speech (given by the parser), lemma (given by a finite-state transducer described by Minnen et al. (2001)) and named-entity tag.

### 3.2 Alignment

The purpose of the second phase is to find a good partial alignment between the typed dependency graphs representing the hypothesis and the text. An alignment consists of a mapping from each node (word) in the hypothesis graph to a single node in the text graph, or to null.[3] Figure 1 gives the alignment for ID 971.

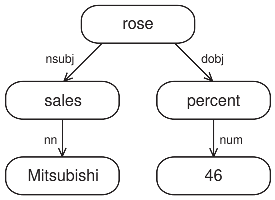
The space of alignments is large: there are $O((m+1)^n)$ possible alignments for a hypothesis graph with $n$ nodes and a text graph with $m$ nodes. We define a measure of alignment quality, and a procedure for identifying high scoring alignments. We choose a locally decomposable scoring function, such that the score of an alignment is the sum of the local node and edge alignment scores. Unfortunately, there is no polynomial time algorithm for finding the exact best alignment. Instead we use an incremental beam search, combined with a node ordering heuristic, to do approximate global search in the space of possible alignments. We have experimented with several alternative search techniques, and found that the solution quality is not very sensitive to the specific search procedure used.

Our scoring measure is designed to favor alignments which align semantically similar subgraphs, irrespective of polarity. For this reason, nodes receive high alignment scores when the words they represent are semantically similar. Synonyms and antonyms receive the highest score, and unrelated words receive the lowest. Our hand-crafted scoring metric takes into account the word, the lemma, and the part of speech, and searches for word relatedness using a range of external resources, including WordNet, precomputed latent semantic analysis matrices, and special-purpose gazettes. Alignment scores also incorporate local edge scores, which are based on the shape of the paths between nodes in the text graph which correspond to adjacent nodes in the hypothesis graph. Preserved edges receive the highest score, and longer paths receive lower scores.

### 3.3 Entailment determination

In the final stage of processing, we make a decision about whether or not the hypothesis is entailed by the text, conditioned on the typed dependency graphs, as well as the best alignment between them.

[3] The limitations of using one-to-one alignments are mitigated by the fact that many multiword expressions (e.g. named entities, noun compounds, multiword prepositions) have been collapsed into single nodes during linguistic analysis.




**Alignment**

| | | |
|---|---|---|
| rose | → | fell |
| sales | → | sales |
| Mitsubishi | → | Mitsubishi_Motors_Corp. |
| percent | → | percent |
| 46 | → | 46 |

Alignment score: $-0.8962$

**Features**

| | |
|---|---|
| Antonyms aligned in pos/pos context | − |
| Structure: main predicate good match | + |
| Number: quantity match | + |
| Date: text date deleted in hypothesis | − |
| Alignment: good score | + |

Entailment score: $-5.4262$

Figure 1: Problem representation for ID 971: typed dependency graph (hypothesis only), alignment, and entailment features.

Because we have a data set of examples that are labeled for entailment, we can use techniques from supervised machine learning to learn a classifier. We adopt the standard approach of defining a featural representation of the problem and then learning a linear decision boundary in the feature space. We focus here on the learning methodology; the next section covers the definition of the set of features.

Defined in this way, one can apply any statistical learning algorithm to this classification task, such as support vector machines, logistic regression, or naive Bayes. We used a logistic regression classifier with a Gaussian prior parameter for regularization. We also compare our learning results with those achieved by hand-setting the weight parameters for the classifier, effectively incorporating strong prior (human) knowledge into the choice of weights.

An advantage to the use of statistical classifiers is that they can be configured to output a probability distribution over possible answers rather than just the most likely answer. This allows us to get confidence estimates for computing a confidence weighted score (see section 5). A major concern in applying machine learning techniques to this classification problem is the relatively small size of the training set, which can lead to overfitting problems. We address this by keeping the feature dimensionality small, and using high regularization penalties in training.

## 4 Feature representation

In the entailment determination phase, the entailment problem is reduced to a representation as a vector of 28 features, over which the statistical classifier described above operates. These features try to capture salient patterns of entailment and non-entailment, with particular attention to contexts which reverse or block monotonicity, such as negations and quantifiers. This section describes the most important groups of features.

**Polarity features.** These features capture the presence (or absence) of linguistic markers of negative polarity contexts in both the text and the hypothesis, such as simple negation (*not*), downward-monotone quantifiers (*no*, *few*), restricting prepositions (*without*, *except*) and superlatives (*tallest*).

**Adjunct features.** These indicate the dropping or adding of syntactic adjuncts when moving from the text to the hypothesis. For the common case of restrictive adjuncts, dropping an adjunct preserves truth (*Dogs barked loudly* $\models$ *Dogs barked*), while adding an adjunct does not (*Dogs barked* $\not\models$ *Dogs barked today*). However, in negative-polarity contexts (such as *No dogs barked*), this heuristic is reversed: adjuncts can safely be added, but not dropped. For example, in ID 59, the hypothesis aligns well with the text, but the addition of *in Iraq* indicates non-entailment.

We identify the "root nodes" of the problem: the root node of the hypothesis graph and the corresponding aligned node in the text graph. Using dependency information, we identify whether adjuncts have been added or dropped. We then determine the *polarity* (negative context, positive context or restrictor of a universal quantifier) of the two root nodes to generate features accordingly.

**Antonymy features.** Entailment problems might involve antonymy, as in ID 971. We check whether an aligned pairs of text/hypothesis words appear to be antonymous by consulting a pre-computed list of about 40,000 antonymous and other contrasting pairs derived from WordNet. For each antonymous pair, we generate one of three boolean features, indicating whether (i) the words appear in contexts of matching polarity, (ii) only the text word appears in a negative-polarity context, or (iii) only the hypothesis word does.

**Modality features.** Modality features capture simple patterns of modal reasoning, as in ID 98, which illustrates the heuristic that possibility does not entail actuality. According to the occurrence (or not) of predefined modality markers, such as *must* or *maybe*, we map the text and the hypothesis to one of six modalities: *possible*, *not possible*, *actual*, *not actual*, *necessary*, and *not necessary*. The text/hypothesis modality pair is then mapped into one of the following entailment judgments: *yes*, *weak yes*, *don't know*, *weak no*, or *no*. For example:

$$(\textit{not possible} \models \textit{not actual})? \quad \Rightarrow \quad \textit{yes}$$
$$(\textit{possible} \models \textit{necessary})? \quad \Rightarrow \quad \textit{weak no}$$

**Factivity features.** The context in which a verb phrase is embedded may carry semantic presuppositions giving rise to (non-)entailments such as *The gangster tried to escape* $\not\models$ *The gangster escaped*. This pattern of entailment, like others, can be reversed by negative polarity markers (*The gangster managed to escape* $\models$ *The gangster escaped* while *The gangster didn't manage to escape* $\not\models$ *The gangster escaped*). To capture these phenomena, we compiled small lists of "factive" and non-factive verbs, clustered according to the kinds of entailments they create. We then determine to which class the parent of the text aligned with the hypothesis root belongs to. If the parent is not in the list, we only check whether the embedding text is an affirmative context or a negative one.

**Quantifier features.** These features are designed to capture entailment relations among simple sentences involving quantification, such as *Every company must report* $\models$ *A company must report* (or *The company*, or *IBM*). No attempt is made to handle multiple quantifiers or scope ambiguities. Each quantifier found in an aligned pair of text/hypothesis words is mapped into one of five quantifier categories: *no*, *some*, *many*, *most*, and *all*. The *no* category is set apart, while an ordering over the other four categories is defined. The *some* category also includes definite and indefinite determiners and small cardinal numbers. A crude attempt is made to handle negation by interchanging *no* and *all* in the presence of negation. Features are generated given the categories of both hypothesis and text.

**Number, date, and time features.** These are designed to recognize (mis-)matches between numbers, dates, and times, as in IDs 1806 and 231. We do some normalization (e.g. of date representations) and have a limited ability to do fuzzy matching. In ID 1806, the mismatched years are correctly identified. Unfortunately, in ID 231 the significance of *over* is not grasped and a mismatch is reported.

**Alignment features.** Our feature representation includes three real-valued features intended to represent the quality of the alignment: *score* is the raw score returned from the alignment phase, while *goodscore* and *badscore* try to capture whether the alignment score is "good" or "bad" by computing the sigmoid function of the distance between the alignment score and hard-coded "good" and "bad" reference values.

## 5 Evaluation

We present results based on the First PASCAL RTE Challenge, which used a development set containing 567 pairs and a test set containing 800 pairs. The data sets are balanced to contain equal numbers of *yes* and *no* answers. The RTE Challenge recommended two evaluation metrics: raw accuracy and confidence weighted score (CWS). The CWS is computed as follows: for each positive integer $k$ up to the size of the test set, we compute accuracy over the $k$ most confident predictions. The CWS is then the average, over $k$, of these partial accuracies. Like raw accuracy, it lies in the interval [0, 1], but it will exceed raw accuracy to the degree that predictions are well-calibrated.

Several characteristics of the RTE problems should be emphasized. Examples are derived from a broad variety of sources, including newswire; therefore systems must be domain-independent. The inferences required are, from a human perspective, fairly superficial: no long chains of reasoning are involved. However, there are "trick" questions expressly designed to foil simplistic techniques. The definition of entailment is informal and approximate: whether a competent speaker with basic knowledge of the world would typically infer the hypothesis from the text. Entailments will certainly depend on linguistic knowledge, and may also depend on world knowledge; however, the scope of required

| Algorithm | RTE1 Dev Set | | RTE1 Test Set | |
|---|---|---|---|---|
| | Acc | CWS | Acc | CWS |
| Random | 50.0% | 50.0% | 50.0% | 50.0% |
| Jijkoun et al. 05 | 61.0% | 64.9% | 55.3% | 55.9% |
| Raina et al. 05 | 57.8% | 66.1% | 55.5% | 63.8% |
| Haghighi et al. 05 | — | — | 56.8% | 61.4% |
| Bos & Markert 05 | — | — | 57.7% | 63.2% |
| Alignment only | 58.7% | 59.1% | 54.5% | 59.7% |
| Hand-tuned | 60.3% | 65.3% | **59.1%** | **65.0%** |
| Learning | 61.2% | 74.4% | **59.1%** | **63.9%** |

Table 2: Performance on the RTE development and test sets. CWS stands for confidence weighted score (see text).

world knowledge is left unspecified.[4]

Despite the informality of the problem definition, human judges exhibit very good agreement on the RTE task, with agreement rate of 91–96% (Dagan et al., 2005). In principle, then, the upper bound for machine performance is quite high. In practice, however, the RTE task is exceedingly difficult for computers. Participants in the first PASCAL RTE workshop reported accuracy from 49% to 59%, and CWS from 50.0% to 69.0% (Dagan et al., 2005).

Table 2 shows results for a range of systems and testing conditions. We report accuracy and CWS on each RTE data set. The baseline for all experiments is random guessing, which always attains 50% accuracy. We show comparable results from recent systems based on lexical similarity (Jijkoun and de Rijke, 2005), graph alignment (Haghighi et al., 2005), weighted abduction (Raina et al., 2005), and a mixed system including theorem proving (Bos and Markert, 2005).

We then show results for our system under several different training regimes. The row labeled "alignment only" describes experiments in which all features except the alignment score are turned off. We predict entailment just in case the alignment score exceeds a threshold which is optimized on development data. "Hand-tuning" describes experiments in which all features are on, but no training occurs; rather, weights are set by hand, according to human intuition. Finally, "learning" describes experiments in which all features are on, and feature weights are trained on the development data. The figures reported for development data performance therefore reflect overfitting; while such results are not a fair measure of overall performance, they can help us assess the adequacy of our feature set: if our features have failed to capture relevant aspects of the problem, we should expect poor performance even when overfitting. It is therefore encouraging to see CWS above 70%. Finally, the figures reported for test data performance are the fairest basis for comparison. These are significantly better than our results for alignment only (Fisher's exact test, $p < 0.05$), indicating that we gain real value from our features. However, the gain over comparable results from other teams is not significant at the $p < 0.05$ level.

A curious observation is that the results for hand-tuned weights are as good or better than results for learned weights. A possible explanation runs as follows. Most of the features represent high-level patterns which arise only occasionally. Because the training data contains only a few hundred examples, many features are active in just a handful of instances; their learned weights are therefore quite noisy. Indeed, a feature which is expected to favor entailment may even wind up with a negative weight: the modal feature *weak yes* is an example. As shown in table 3, the learned weight for this feature was strongly negative — but this resulted from a single training example in which the feature was active but the hypothesis was not entailed. In such cases, we shouldn't expect good generalization to test data, and human intuition about the "value" of specific features may be more reliable.

Table 3 shows the values learned for selected feature weights. As expected, the features *added adjunct in* all *context*, *modal yes*, and *text is factive* were all found to be strong indicators of entailment, while *date insert*, *date modifier insert*, *widening from text to hyp* all indicate lack of entailment. Interestingly, *text has neg marker* and *text & hyp diff polarity* were also found to disfavor entailment; while this outcome is sensible, it was not anticipated or designed.

## 6 Conclusion

The best current approaches to the problem of textual inference work by aligning semantic graphs,

[4]Each RTE problem is also tagged as belonging to one of seven *tasks*. Previous work (Raina et al., 2005) has shown that conditioning on task can significantly improve accuracy. In this work, however, we ignore the task variable, and none of the results shown in table 2 reflect optimization by task.

| Feature class & condition | | weight |
|---|---|---|
| Adjunct | added adjunct in *all* context | 1.40 |
| Date | date mismatch | 1.30 |
| Alignment | good score | 1.10 |
| Modal | yes | 0.70 |
| Modal | no | 0.51 |
| Factive | text is factive | 0.46 |
| … | … | … |
| Polarity | text & hyp same polarity | −0.45 |
| Modal | don't know | −0.59 |
| Quantifier | widening from text to hyp | −0.66 |
| Polarity | text has neg marker | −0.66 |
| Polarity | text & hyp diff polarity | −0.72 |
| Alignment | bad score | −1.53 |
| Date | date modifier insert | −1.57 |
| Modal | weak yes | −1.92 |
| Date | date insert | −2.63 |

Table 3: Learned weights for selected features. Positive weights favor entailment. Weights near 0 are omitted. Based on training on the PASCAL RTE development set.

using a locally-decomposable alignment score as a proxy for strength of entailment. We have argued that such models suffer from three crucial limitations: an assumption of monotonicity, an assumption of locality, and a confounding of alignment and entailment determination.

We have described a system which extends alignment-based systems while attempting to address these limitations. After finding the best alignment between text and hypothesis, we extract high-level semantic features of the entailment problem, and input these features to a statistical classifier to make an entailment decision. Using this multi-stage architecture, we report results on the PASCAL RTE data which surpass previously-reported results for alignment-based systems.

We see the present work as a first step in a promising direction. Much work remains in improving the entailment features, many of which may be seen as rough approximations to a formal monotonicity calculus. In future, we aim to combine more precise modeling of monotonicity effects with better modeling of paraphrase equivalence.

## Acknowledgements

We thank Anna Rafferty, Josh Ainslie, and particularly Roger Grosse for contributions to the ideas and system reported here. This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program.

## References

E. Akhmatova. 2005. Textual entailment resolution via atomic propositions. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005*.

J. Bos and K. Markert. 2005. Recognising textual entailment with logical inference. In *EMNLP-05*.

I. Dagan, O. Glickman, and B. Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. An inference model for semantic entailment and question-answering. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*.

C. Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press.

A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi, and J. Stephan. 2005. Applying COGEX to recognize textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

A. Haghighi, A. Ng, and C. D. Manning. 2005. Robust textual inference via graph matching. In *EMNLP-05*.

J. R. Hobbs, M. Stickel, P. Martin, and D. D. Edwards. 1988. Interpretation as abduction. In *26th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 95–103, Buffalo, New York.

V. Jijkoun and M. de Rijke. 2005. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenge Workshop on Recognising Textual Entailment, 2005*, pages 73–76.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *LREC 2006*.

E. Marsi and E. Krahmer. 2005. Classification of semantic relations by humans and machines. In *Proceedings of the ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL 2003*.

G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing in English. In *Natural Language Engineering*, volume 7(3), pages 207–233.

D. Moldovan, C. Clark, S. Harabagiu, and S. Maiorano. 2003. COGEX: A logic prover for question answering. In *NAACL-03*.

M. Pasca and S. Harabagiu. 2001. High performance question/answering. In *SIGIR-01*, pages 366–374.

R. Raina, A .Ng, and C. D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*.

M. Tatu and D. Moldovan. 2005. A semantic approach to recognizing textual entailment. In *HLT/EMNLP 2005*, pages 371–378.