# Toward Dependency Path based Entailment

**Rodney D. Nielsen, Wayne Ward and James H. Martin**
Center for Spoken Language Research
Institute of Cognitive Science
Department of Computer Science
University of Colorado, Boulder
Rodney.Nielsen, Wayne.Ward, James.Martin@Colorado.edu

## Abstract

We present our submission to the RTE2 challenge which takes steps in the direction of dynamically entailing hypotheses via their dependency paths. We evaluate semantic similarity between sentences utilizing corpus co-occurrence estimates of various dependency path features and show a 2.7% improvement on the RTE1 dataset.

## 1 Introduction

Determining whether the propositions in one text fragment are entailed by those in another fragment is important to numerous natural language processing applications. Consider the intelligent tutoring setting, where it is critical for the tutor to assess which propositions in the desired answer are entailed by the student's answer and, conversely, whether each proposition in the student's answer is entailed by a combination of world knowledge and the tutor subject matter. Truly effective interaction and pedagogy is only possible if the automated tutor can assess this entailment at a relatively fine grain of detail.

The PASCAL Recognizing Textual Entailment (RTE) challenge (Dagan et al., 2005) has brought the issue of textual entailment before a broad community of researchers in a task independent fashion. This task requires systems to make judgments as to whether a human reading a text fragment *t* would typically consider it to entail a second, hypothesis, text fragment *h*. This paper describes our submission to the 2006 RTE challenge. We first outline the goal of our dependency path based entailment and describe related prior work. Then we describe our current implementation in section 3. We then present our results on the 2006 RTE challenge, followed by a discussion of related issues in section 5.

## 2 Dependency Path-based Entailment

Dekang Lin and Patrick Pantel (2001) propose extracting paraphrases or discovering inference rules from text (DIRT) by extending Harris' Distributional Hypothesis, which states that words that occur in the same contexts tend to be similar. Specifically, they extend this hypothesis to dependency paths in a MiniPar (Lin, 1993) parse tree, stating that paths that occur in similar contexts tend to have similar meanings. The contexts for these paths include the dependency nodes at each end of the path. If two paths occur in a meaningful number of similar contexts, they interpret the paths as providing an inference rule, which in their context is similar but not identical to a paraphrase. They use point-wise mutual information to decide whether these inference rules are statistically meaningful and run their system on 1 GB of newspaper text, producing a large number of inference rules.

Multiple entries in the 2005 RTE challenge attempted to use these rules to improve their entailment predictions, but noted that the existing database of rules provided inadequate coverage (e.g., Braz et al., 2005; Raina et al., 2005). Our goal is to implement a technique that improves this coverage. Specifically, rather than generating

these rules a priori, we interpret the RTE text-hypothesis pair as providing potential inference rules whose validity is to be determined. In this setting, the task is to align the terms in *t* and *h* which act as the inference rule context and then verify that the path(s) between the context terms in *h* are entailed by the corresponding path(s) in *t*.

Ultimately our aim is to extend the DIRT approach and implement it as a dynamic system. Context word alignment will be performed using a mixture of techniques (c.f., Turney et al., 2003) including PMI-IR (Turney, 2001), WordNet and thesaurus expansion, and Latent Semantic Analysis (Deerwester et al., 1990). Given a highly probable alignment between two pairs of words (or phrases), $w_1$ and $w_2$ from *h* aligning with $w_1$' and $w_2$' from *t* respectively, we will determine the dependency paths from $w_1$ to $w_2$, $p_h$, and from $w_1$' to $w_2$', $p_t$. These dependency paths will be expanded to their minimal meaning-retaining surface forms, $s_h$ and $s_t$ respectively, by adding back necessary terms that are not directly on the dependency path (e.g., negation operators such as 'not' that are not directly on the path will be added back to the minimal meaning-retaining surface forms). These surface forms will then be used in queries to generate a likelihood estimate for $s_t$ entailing $s_h$, which is a necessary condition to show that *t* entails *h*. Entailment is assumed to hold if all of the words or phrases in *h* are aligned reasonably well with those in *t* or they are part of the paths connecting these phrases and these paths show a high likelihood of entailment.

## 3 Initial Implementation

In this section, we detail the status of our current implementation, which is perhaps a first order approximation of the goal described in the preceding section, but nonetheless provided a significant improvement in accuracy over our baseline system. We cast the problem as a classification task and generate features related to word, phrase and dependency path similarity. Our features are based primarily on corpus co-occurrence statistics, so we first describe the corpora and information retrieval engine we utilized. Then we describe our features, followed by an outline of our classification approach and the training dataset.

### 3.1 Corpora

The features described in the following subsections are based on document co-occurrence counts. Rather than use the web as our corpus, as was done by Turney (2001) and Glickman et al. (2005), we use three publicly available corpora totaling 7.4M articles and 2.6B indexed terms.

**English Gigaword:** English Gigaword (Graff, 2003) is newspaper text from five sources ranging from 1995-2004. It consists of about 5.7M news articles and 2.1B words on a wide variety of subjects. This resulted in documents with an average of around 375 indexed tokens. This corpus comprises 77% of our total documents and 83% of the total indexed words.

**Reuters Corpus Volume 1:** The Reuters corpus (Lewis et al., 2004) consists of one year of Reuters newswire from 1996-1997. It provided 0.8M articles and 0.17B indexed words, averaging 213 words per article.

**TIPSTER:** The three volume TIPSTER corpus[1] includes documents from a variety of sources, including newspaper text, and ranges from 1987-1992. It provided 0.9M articles and 0.26B indexed words, averaging 291 words per article.

### 3.2 Query Engine

The above corpora were indexed and searched using Lucene.[2] Two indices were created, the first using the `StandardAnalyzer` and the second adding the `PorterStemFilter`. Both indices excluded only three words, {a, an, the}. However, when referring to content words in the feature descriptions that follow, Lucene's standard stop-word list was utilized, with the exception of removing the words *no* and *not*.

### 3.3 Features

We generate features that loosely assess the overall quality of word and phrase alignments. Information from these sets of features will in the future be used to perform a formal word alignment.

**Unigram Word Alignment:** The first set of features correspond with the lexical entailment calculations in (Turney, 2001; Glickman et al., 2005). Here a lexical entailment likelihood is derived from point-wise mutual information, common

[1] http:// www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A
[2] http://lucene.apache.org/

terms are factored out and maximum likelihood estimates are made based on corpus co-occurrence statistics. For a single content word $w$ from $h$, their methods estimate the probability of entailment as

$$P(Tr_w = 1 \mid t) \approx \max_{v \in t} P(Tr_w = 1 \mid v) \approx \max_{v \in t} \frac{n_{w,v}}{n_v} \quad (1)$$

where $v$ represents a word in $t$, $n_v$ is the number of documents in which $v$ occurs, $n_{w,v}$ is the number of documents in which $w$ and $v$ co-occur, and the truth value or entailment of $w$ is assumed to be determined primarily by the single aligned word from $t$ that maximizes this estimate.

Glickman et al. then estimate the probability of entailment for $h$ as the product of the probabilities of each of its content words $w$ being entailed:

$$P(Tr_h = 1 \mid t) = \prod_{w \in h} \max_{v \in t} \frac{n_{w,v}}{n_v} \quad (2)$$

One weakness of this product of the maximum likelihood estimates (MLEs) is that longer hypotheses result in lower entailment probabilities. Glickman et al. noted that a tf-idf weighted average approach resulted in lower accuracy in the 2005 RTE challenge dataset. It could be argued that the more someone says or writes, the more likely they are to say something inaccurate, but we believe this to be an anomaly that is perhaps specific to the way the RTE1 dataset was created. Therefore, in addition to the product we include features for the average and the geometric mean of the MLEs. We also include a feature for the worst non-zero MLE, believing that one very poorly entailed word could imply the entire hypothesis is not entailed.

Since these estimates are subject to significant variance depending on, among other things, $n_v$ – the number of documents in which the entailing word occurs, we include a number of features to expose this information to the classifier. We include $n_v$ for the lowest non-zero MLE, the largest $n_v$ with a zero MLE, and the smallest $n_v$ with a nonzero MLE. We also provide the classifier with the count of words $w$ that do not co-occur with any $v$, (i.e., the number of words $w$ in $h$ where $n_{w,v}$ is zero for all $v$ in $t$) and the count of words $w$ that do co-occur with at least one $v$.

The features described above are listed in Table 1, but in a slightly more general form as they are repeated in 24 additional contexts described in the following paragraphs.

| **Core Repeated Features** |
|---|
| Product of MLEs |
| Average of MLEs |
| Geometric Mean of MLEs |
| Worst Non-Zero MLE |
| Entailing Ngrams for the Lowest Non-Zero MLE |
| Largest Entailing Ngram Count with a Zero MLE |
| Smallest Entailing Ngram Count with a Non-Zero MLE |
| Count of Ngrams in $h$ that do not Co-occur with any Ngrams from $t$ |
| Count of Ngrams in $h$ that do Co-occur with Ngrams in $t$ |

Table 1: Core Features (see text for descriptions)

**Bigram Word Alignment:** The second set of word alignment-related features involves using bigrams rather than unigrams. To measure the similarity of $w_i$ and $v_j$, we perform bigram co-occurrence queries using the words on each side of $w_i$ and $v_j$ as contexts; again, using document hit counts to calculate the MLE for $w_i$

$$\mathrm{MLE}(w_i) = \max_{v \in t} \frac{1}{k} \left( \begin{array}{l} \delta_1 \dfrac{n_{w_{i-1}w_i, w_{i-1}v_j}}{n_{w_{i-1}v_j}} + \delta_2 \dfrac{n_{w_i w_{i+1}, v_j w_{i+1}}}{n_{v_j w_{i+1}}} \\ \quad + \delta_3 \dfrac{n_{v_{j-1}w_i, v_{j-1}v_j}}{n_{v_{j-1}v_j}} + \delta_4 \dfrac{n_{w_i v_{j+1}, v_j v_{j+1}}}{n_{v_j v_{j+1}}} \end{array} \right) \quad (3)$$

where the $\delta_h$ are 1 if the term is included in the average and 0 otherwise – to avoid severe penalties where one or more queries result in very low MLEs, we only average across the highest $k$ of the four MLEs up to the first value with zero co-occurrences or the first value with more than 30 co-occurrences.

**Average Word Alignment:** The third set of word alignment-related features is derived by generating averages for each $w$ across the information used in the first two sets, assuming both query procedures result in positive MLEs, otherwise it uses the unigram information. For each $w$, the co-occurrence count and entailing ngram count is taken from the query with the largest MLE.

**Stem-based Word Alignment:** The fourth-sixth sets of word alignment-related features replicate the first-third sets, but are based on the Porter stems of the words rather than their surface forms.

**Bag-of-Dependencies:** This set of features treats $h$ and $t$ each as a bag of MiniPar dependencies (i.e., independent parent-child node pairs). We refer to a MiniPar node as a *Word Node* if it includes a <word> element that maps to a word (not punctuation) in the original text fragment. We refer to a node, $w_p$, as a *Word Parent* of a node $w_c$, if

(a) it is a Word Node and it is referenced as $w_c$'s MiniPar parent, (b) it is a Word Parent of $w_q$, $w_q$ is the node reference as the parent of $w_c$, and $w_q$ is not a Word Node, (c) it is a Word Node and it references $w_c$ as a MiniPar antecedent, or (d) it is a Word Parent of $w_r$, $w_r$ references $w_c$ as an antecedent, and $w_r$ is not a Word Node. While a node will never have more than one MiniPar parent, it will have two Word Parents, for example, when it is the subject of two verbs.

We generate MLEs for all, $w_c$, content words in $h$ that have a Word Parent. Let $<w_c, w_p>$ be the bigram containing $w_c$ and $w_p$ in the same order they occur in $h$. Similarly, let $<v_c, v_p>$ be a surface ordered bigram containing any $v_c$ from $t$ and one of its Word Parents $v_p$. The MLE for $<w_c, w_p>$ is made based on document co-occurrence counts, using the $<v_c, v_p>$ bigram that maximizes the MLE, as shown in equation 4. This calculation disregards the actual dependency type. The queries use the Porter stem-based index.

$$\mathrm{MLE}\left(\left\langle w_c, w_p \right\rangle\right) = \max_{\langle v_c, v_p \rangle \in t_d} \frac{n_{\langle w_c, w_p \rangle, \langle v_c, v_p \rangle}}{n_{\langle v_c, v_p \rangle}} \quad (4)$$

The final MLE associated with a given $w_c$ is the average MLE calculated by equation 4 for all of its parents $w_p$. The MLEs for all the $w_c$ in $h$ are then combined to generate a set of features that parallels those discussed above in the bag-of-words feature sets (see Table 1). These features indicate the likelihood that the dependencies in $h$ are entailed by those in $t$.

**Dependency Path Based Entailment:** The rest of the paragraphs in this subsection describe our first steps toward implementing the Dependency Path Based Entailment approach. All queries in these sections were run against the Porter stems.

**Descendent Relation Features:** Given a dependency node $w_c$ and its Word Parent $w_p$, let $w_c$ be defined as a *Word Child* of $w_p$. The descendent relation features for a node $w_p$ are the same MLE and ngram count features as described in the Bag-of-Dependencies above, but include just the dependencies with each of $w_p$'s Word Children $w_c$ in the bag. The MLE for $w_c$ is recursively computed from the descendent relation MLE of its children; this bag of MLEs is averaged (or unitized) before being combined with the MLEs of $w_p$'s other children. These features are not used by the classifier directly, but are used repeatedly to generate the feature sets described below.

**Combined Verb Descendent Relations:** This set of features is generated by combining the descendent relation features of each verb in $h$. For each of these verbs, we process all dependency paths that include content words. Before combining the features for all verbs, their individual MLE features are unitized as discussed above under Descendent Relation Features.

**Worst Verb Descendent Relations:** This is simply the set of features associated with the verb that has the lowest MLE value.

**Combined Subject Descendent Relations:** This set of features parallels the Combined Verb Descendent Relations features. It is calculated by combining the Descendent Relation Features for all the Word Children $w_c$ of verbs where $w_c$ has the `subj` relation with its parent verb.

**Worst Subject Descendent Relations:** The set of features used in the preceding paragraph that had the lowest MLE value.

**Combined Subject-to-Verb Relations:** This set of features is based on the same Word Children $w_c$ as the Combined Subject Descendent Relations. Here the features are constructed from the dependencies between the subjects and the verbs, rather than the dependencies between the subjects and their child nodes.

**Worst Subject-to-Verb Relations:** The set of features from the preceding paragraph associate with the lowest MLE.

**Object, pcomp-n, and Other Relations:** The same four sets of features that are generated for the subject are also constructed for the object, the head nouns of other prepositional phrases having dependencies with the verb, and all other content word types having a dependency link to the verbs.

**Other Features:** We also provide the classifier with features that indicate the RTE task type (IR, IE, QA, or SUM), the number of content words in $h$, the number of content words in $t$ and the number of verbs in $h$.

### 3.4 Classification Approach

We used a mixture of experts as our classifier, combining the unweighted votes and probability estimates of a variety of classifiers, all within the Weka machine learning package (Witten and Frank, 2000). We trained separate classifiers for

the document summarization, SUM, subset of the data, since this resulted in better performance during cross-validation on our training sets. Each individual classifier was also tuned somewhat based on training set cross-validation.[3] Our first submission made decisions based on the average probability of the classifiers. Where classifiers output almost strictly 0 and 1 probability estimates (SMO and VotedPerceptron), we normalized these estimates to be consistent with the classifiers' accuracy on training set cross-validation. Our second submission made decisions based on the majority vote among component classifiers, breaking any ties with the average probability estimate.

### 3.5 Training Set

We trained our IE,IR,QA classifier strictly on the associated RTE2 training data, but trained our SUM classifier utilizing both the RTE2 SUM training data and the RTE1 CD training and test sets, since cross-validation on the training data suggested better performance taking this approach.

## 4 Results

Table 2 shows our results on each subset of the data for each of our two submissions. For comparison, Table 3 shows results from cross-validation on our training sets, results when training and testing on the RTE1 (2005) datasets, and the best accuracy results for a full submission by anyone at the RTE1 challenge (Dagan et al., 2005).

## 5 Discussion

**Comparison with RTE1 Submissions:** As can be seen in Table 3, the system described here outperformed the submission with the best accuracy at the RTE1 challenge by 2.7%. Part of the reason for this is because we trained separate classifiers for the CD and non-CD portions of the dataset. In cross-validation on the RTE1 training set, we see an absolute increase of 1.6% in the error rate when combining all of the examples into a single dataset. It is interesting to note that of the RTE1 participants that reported task-specific information at a level of detail sufficient to determine their CD versus non-CD accuracy, the average non-CD accuracy was essentially at chance, as shown in the last line of Table 3. The best non-CD accuracy was only 52.8%, where the accuracy for our system was 3.1% higher on that section of the data.

| Run | IE | | IR | | QA | | SUM | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| Run1 | 54.0 | 50.8 | 61.5 | 63.6 | 55.0 | 57.8 | 68.0 | 82.3 | 59.6 | 64.6 |
| Run2 | 53.5 | 50.7 | 59.5 | 64.2 | 54.0 | 57.3 | 68.0 | 82.4 | 58.8 | 64.9 |

Table 2: Accuracy and Average Precision (Run 1: average probability estimate mixture; Run 2: majority vote of component classifiers)

| Run | SUM / CD | NonSUM / NonCD | Overall |
|---|---|---|---|
| RTE2 Test Set | 68.0 | 56.8 | 59.6 |
| RTE2 Trng CV | 83.9 [4] | 63.2 | 68.4 |
| RTE1 Trng CV | 81.6 | 55.0 | 59.6 |
| RTE1 Test Set | 84.7 | 55.9 | 61.3 |
| Best RTE1 Submission | 83.3 | 52.8 | 58.6 |
| Ave RTE1 Submission | 75.2 | 49.8 | 54.5 |

Table 3: Accuracy (ave. probability estimate mixture)

**Comparison with Training Cross-Validation:** Comparing our RTE2 test results with those for cross-validation on the training data shows a significant decline in accuracy. This is true for both the SUM subset, which had a 15.9% decline[4], and the non-SUM subset, which saw a 6.4% decline. The majority of this decline is not due to over-fitting the training data. Using component classifiers that are not tuned to the training data leads to only a 2% decrease in accuracy on the non-SUM portion and effectively no decrease on the SUM portion of the training set. Additionally, many of the good classifiers performed close to the accuracy of the mixture of experts. We hypothesize that most of the decrease in performance on the test set is due to differences in the entailment pairs, but we do not want to examine the test set and bias our future results.

**Feature Analysis:** Preliminary feature ablation studies based on training set cross-validation suggest that nearly all of our features might be helping the accuracy in some context. The core repeated feature from Table 1 that appeared to have the most significant positive effect on accuracy was the average MLE. Removing this feature from all

[3] {IE,IR,QA} classifiers with tuned parameters: AdaBoost (I=50), ADTree (I=7), ClassificationViaRegression, DecisionTable (X=2, -I), JRip (O=6), LogitBoost (I=2), MultiBoost (I=35), RandomCommittee (I=65), RandomForest (I=100, K=24), SimpleLogistic (H=300), SMO (N=1), SMO (N=1, C=0.73), VotedPerceptron
SUM classifiers with tuned parameters: AdaBoost, ADTree, ClassificationViaRegression, DecisionTable (X=2, -I), JRip, LMT, LogitBoost, RandomCommittee (I=100), RandomForest (I=129, K=24), RepTree, SMO

[4] This is based on the average performance over the RTE1 CD data and the RTE2 SUM training data, which could not be differentiated in the Weka output.

of the feature sets resulted in a decrease in performance of 4% (24/600 additional misclassifications) for the {IE,IR,QA} portion of the dataset. The best single feature appears to be the stem unigram average MLE. Training a linear classifier on just this feature results in a decrease in accuracy of only 5% (30/600) relative to the tuned mixture of experts.

On the other hand, feature analysis of the SUM portion of the dataset suggested that virtually all of the feature sets were irrelevant. Multiple features individually performed very close to the same accuracy as the mixture of experts. Again, the best feature appears to be the stem unigram average MLE.

## 6 Summary and Future Work

We presented a dependency path based entailment approach and our initial implementation which takes steps in this direction. Our results are very promising, showing a 2.7% improvement over the best accuracy for all full submissions on the RTE1 dataset.

Future work includes implementing the full system described in Section 2. We intend to utilize additional training data, perhaps following the approach proposed by Burger and Ferro (2005) to use news article headlines and their opening paragraphs as entailment pairs. We also plan to work on our mixture of experts to verify the effect of adding other classifiers, among other issues.

## Acknowledgements

We would like to thank Graeme Hirst for advice on this project and Steven Bethard for support in using various tools.

## References

Braz, Rodrigo de Salvo, Girju, Roxana, Punyakanok, Vasin, Roth, Dan, and Sammons, Mark. (2005). An Inference Model for Semantic Entailment in Natural Language. In *Proc. of the PASCAL Recognizing Textual Entailment Challenge Workshop*.

Burger, John and Ferro, Lisa. (2005). Generating an Entailment Corpus from News Headlines. *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54, Ann Arbor, June 2005.

Dagan, Ido, Glickman, Oren, and Magnini, Bernardo. (2005). The PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the PASCAL Recognizing Textual Entailment Challenge Workshop*.

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

Glickman, Oren and Dagan, Ido, and Koppel, Moshe. (2005). Web Based Probabilistic Textual Entailment. In *Proceedings of the PASCAL Recognizing Textual Entailment Challenge Workshop*.

Graff, David (2003). *English Gigaword*. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR*, 5:361-397.

Lin, D. (1993). Principle-Based Parsing Without Over-Generation. In *Proceedings of ACL-93*. pp. 112-120. Columbus, OH.

Lin, Dekang and Pantel, Patrick. (2001). Discovery of inference rules for Question Answering. In Natural Language Engineering, 7(4):343-360.

Raina, Rajat, Haghighi, Aria, Cox, Christopher, Finkel, Jenny, Michels, Jeff, Toutanova, Kristina, MacCartney, Bill, de Marneffe, Marie-Catherine, Manning, Christopher D., and Ng, Andrew Y. (2005). Robust Textual Inference using Diverse Knowledge Sources. In *Proc. of the PASCAL Recognizing Textual Entailment Challenge Workshop*.

Turney, Peter D. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, 491–502.

Turney, P.D., Littman, M.L., Bigham, J., and Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. *Proceedings of RANLP*, 482-489. Borovets, Bulgaria.

Witten, Ian H. and Frank Eibe. (2000). *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.