# Word and tree-based similarities for textual entailment

**Frank Schilder**
R&D
Thomson Legal & Regulatory
610 Opperman Drive
Eagan MN 55123, USA
Frank.Schilder@Thomson.com

**Bridget Thomson McInnes**
Department of Computer Science
and Engineering
University of Minnesota
200 Union Street SE
Minneapolis MN 55455, USA
bthomson@cs.umn.edu

## Abstract

To calculate the textual entailment between two sentences in the RTE data, we use a word-based similarity combined with a tree-based similarity approach. For each approach, we experiment with two different metrics. In order to combine the four different metrics we use an SVM trained on the development set. Our results show an overall accuracy of 0.5437-0.555.

## 1 Introduction

The textual entailment challenge is defined as the task of determining whether the hypothesis $H$ is entailed by a longer text $T$. The definition of textual entailment is based on the human capability in reading $T$ and inferring that $H$ is most likely true. The textual entailment relation holds, for example, for the following pair:

$T$**:** The settlement must be approved by Citigroup's board of directors and the board of Regents of the University of California, the lead plaintiff for investors in the case.

$H$**:** The settlement is pending approval by Citigroup's directors and the board of Regents at the University of California.

The textual entailment relation does not hold, on the other hand, for the following pair:

$T$**:** Arsenal sneaked a 1-0 victory over Birmingham at Highbury, taking advantage of Stephen Clemence's own-goal in the 81st minute.

$H$**:** Arsenal lost to Birmingham.

For this year's challenge, a development data set of 800 pairs was provided by the organizers. This set is divided into four subsets. The subsets are created for different applications and partly taken from the output from actual NLP systems. The four application settings are information extraction (IE), information retrieval (IR), question answering (QA), and multi-document summarization (SUM).

Our approach is partly motivated by the four different application settings. We hypothesized that the subtasks IE and IR are better addressed by a word-based similarity approach, whereas QA and SUM would benefit more from a tree-based approach that takes the tree structure of the generated parse tree into account. Based on this assumption, we developed two word-based similarity and two tree- based similarity approaches. Given the similarity scores for each of these metrics we used an SVM to compute the best combination of the computed metrics, the subtasks and the lengths of $T$ and $H$.

## 2 Semantic similarity approaches

In this section, we discuss two word-based semantic similarity metrics to identify the similarity between two sentences as a function of the similarity between the words in the sentence. How similarities of words can be measured are described in section 2.1 before we describe in more detail the two metrics between

sentences we developed based on (Corley and Mihalcea, 2005).

### 2.1 Word-based approaches to semantic similarity

There exist a number of *similarity measures* which rely on the position of words in an *is-a* hierarchy and *relatedness measures* which do not rely on such a hierarchy. For the word-based similarity approach, we chose to look at seven measures. The similarity measures used can be grouped into two categories: path length based and information content (IC) based. The path length based measures used are: Wu & Palmer, Leacock & Chodorow, and Path. The IC measures used are: Resnik, Jiang & Conrath, and Lin. The relatedness measure used is Lesk.

The (Wu and Palmer, 1994) similarity measure defined in Equation 1 measures the depth of two concepts in WordNet and the depth of their least common subsumer (LCS). The LCS is the most specific concept two concepts share as an ancestor.

$$sim_{wup} = \frac{2 * depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \quad (1)$$

The (Leacock and Chodorow, 1998) measure defined in Equation 2 is the negative log of the shortest path between two concepts in a taxonomy (in this case WordNet, (Fellbaum, 1998)) divided by twice the total depth of the taxonomy ($D$).

$$sim_{lch} = -log\frac{minpath(c_1, c_2)}{2 * D} \quad (2)$$

The Path measure (Pedersen et al., 2004) is a simple measure that determines the similarity between two concepts in WordNet by counting the number of nodes between them.

The (Resnik, 1995) measure defined in Equation 3 is based on the information content. It is the negative log of the probability of the concepts which is define as the information content of the LCS of the two concepts.

$$sim_{res} = IC(lcs(c_1, c_2) = -log(P(lcs(c_1, c_2))) \quad (3)$$

The (J. Jiang, 1997) measure defined in Equation 4 is based on the IC of the two concepts as defined by (Resnik, 1995). The measure is modified to include the length of the path between the two concepts.

$$sim_{jcn} = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))} \quad (4)$$

The (Lin, 1998b) measure defined in Equation 5 is also based on (Resnik, 1995) IC but is modified to include the IC of the two concepts.

$$sim_{lin} = \frac{2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (5)$$

The (Lesk, 1986) measure is based on the overlap between the words in the two concepts definitions (in this case WordNet glosses).

### 2.2 Computing word similarity for sentences

In this section, we propose two metrics to determine the similarity between two sentences based on the similarity between the words in the sentences. The first is a non-symmetric metric that computes the Euclidean distance between a vector of similarity scores and the origin (OVA). The second one is a symmetric metric that computes the Euclidean distance between two vectors containing similarity scores (TVA).

#### 2.2.1 Origin vector approach

In the origin based approach (OVA), for each word in the text sentence, we obtain the maximum similarity score between the text word and the words with the same corresponding part-of speech in the hypothesis sentence using the similarity and relatedness measure described above. The similarity scores are obtained using the WordNet::Similarity package (Pedersen et al., 2004). The measures are defined for concept-concept similarity but can be used for word-word similarity as well. WordNet is a lexical reference system which models the hierarchical relationships between words. The similarity package requires that each word has a predefined WordNet sense. To obtain these, we use the WordNet::SenseRelate::AllWords (Michelizzi, 2005) package which assigns a WordNet sense to every content word in a sentence. This approach performed better than using just the first sense of the word.

WordNet is only defined for noun and verbs. Hence, for adjectives and adverbs, we perform an exact lexical matching giving them a maximum similarity score of one for a match and zero otherwise. This equation is similar to the *maxSim* equation proposed by (Corley and Mihalcea, 2005).

These computations give us a vector of maximum similarity scores which we weight by multiplying the maximum similarity score for the text word with its Inverse Document Frequency (IDF), as defined in Equation 6 where $t_k$ is a word in the text sentence (cf. (Corley and Mihalcea, 2005)).

$$sim_{t_k} = \frac{maxSim(t_k) * IDF(t_k)}{\sum_{t_i} maxSim(t_i) * IDF(t_i)} \quad (6)$$

The Euclidean distance using Equation 7 is calculated between the similarity vector and the origin . The closer to the origin the vector is the less similar the text and hypothesis sentences are from one another.

$$simScore(T_i, T_j) = \sqrt{\sum_{t_k} sim_{t_k}^2} \quad (7)$$

### 2.2.2 Two vector approach

In the two vector approach (TVA), we create a similarity vector for the text sentence as described in our previous approach. We then perform the same calculations only reversing the text and hypothesis sentences, obtaining a similarity vector for the hypothesis sentence. We then calculate the Euclidean distance using Equation 8 between the text vector ($t$) and the hypothesis vector ($h$). The size of the $t$ and $h$ vectors should be equal, as required by the metric. Therefore, we pad the smaller of the two vectors with zero's. The smaller the distance between the two vectors the more likely it is that they are related.

$$simScore(T_i, T_j) = \sqrt{\sum_{w_k} (sim_{t_k} - sim_{h_k})^2} \quad (8)$$

## 3 Tree-based approaches to semantic similarity

In order to compute the similarity between the parse trees, we used two tree similarity measures found in the literature. The first one is based on the algorithm developed by (Shasha and Zhang, 1989) and has already been used in the context of the textual entailment challenge by (Kouylekov and Magnini, 2005). The second one is an approximate tree similarity metric proposed by (Augsten et al., 2005). For the remainder of this paper, these metrics will be called *fdist* and *adist*, respectively. The tree structures used for these similarity metrics were derived from dependency parses we obtained from running the MiniPar parser on all $T - H$ pairs (Lin, 1998a).

In the following sections, we will briefly describe the two algorithms. However, due to the limited space we refer the reader to the detailed description of these algorithms in the literature.

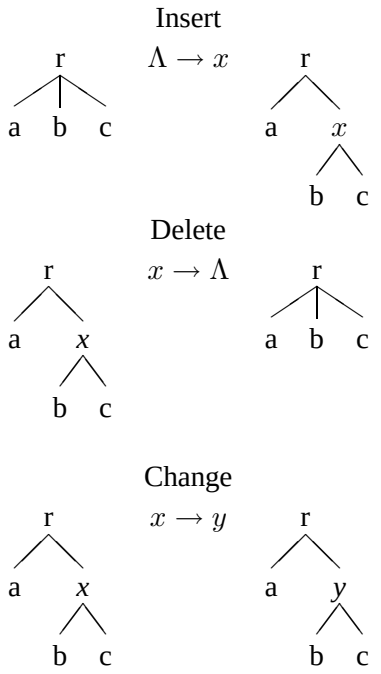


Figure 1: The three editing options

### 3.1 *fdist*

The original tree edit distance algorithm allows for three different operations in order to transform a tree $t$ into a tree $h$: (a) delete, (b) insert and (c) change. Figure 1 depicts these three operations. The weights of the penalties imposed by the three different operations are defined in table 1.[1]

[1]The function $nodes(T)$ generates the set of all nodes and leaves in a tree $T$. The function mi-penalty(x,y) gives back a penalty based on the mutual information for the words $x$ and $y$.

$$
\begin{array}{lcl}
X \rightarrow \Lambda & : & 0 \\
\Lambda \rightarrow x & : & \begin{cases} 0 & \text{if } x \in nodes(H) \\ 5 & \text{otherwise} \end{cases} \\
x \rightarrow y & : & mi - penalty(x, y)
\end{array}
$$

Table 1: Penalty for editing operations

Note that our approach differs from the one proposed by (Kouylekov and Magnini, 2005). Instead of relying on the *IDF* measures, we used mutual information (mi) as an information theoretic measure for the similarity between words to be changed. This measure has shown good results for the automatic discovery of synonyms (Turney, 2001).

Instead of mining the web, as (Turney, 2001) did, we extracted the mutual information for word pairs from a large news corpus. Incorporating this measure into the change penalty of the tree similarity algorithm will allow us to model the transformation from $T$ to $H$ based on the shared information of the words. The function $mi - penalty$ is defined in a way that words that normally occur in similar contexts will get a low penalty (e.g. *bought*/*purchased*), whereas words that do not share the same context will get a very high penalty.

While experimenting with the tree-edit distance algorithm, we noticed that one drawback of this approach is that the operations are carried out without taking the context into account. Passive sentence or transposed sentences would get an unusual high penalty. Hence, we were looking for a tree similarity algorithm that focuses more on the actual structure of the tree.

### 3.2 *adist*

The approximate tree edit distance algorithm proposed by (Augsten et al., 2005) has a number of advantages over the original tree edit distance algorithm by (Shasha and Zhang, 1989). Differences in the actual tree structure become more pronounced and it is computationally far less expensive.

The algorithm uses so-called $p, q$-grams which are computed on the basis of an extended tree $\mathrm{T}^{p,q}$ that contains extra empty nodes (see figure 2). The $p, q$-grams are derived from all possible subtrees of $\mathrm{T}^{p,q}$ given an anchor node that has $p-1$ ancestors and $q$ children. In the extended tree in figure 2, the node

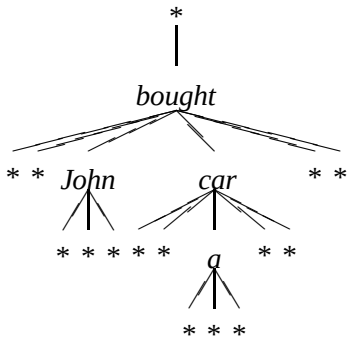


Figure 2: $p, q$-extended tree with $p = 2, q = 3$

*bought* would be a possible anchor that has $*$ as ancestor and combined with the bag of all possible $q$ children produces the following $p, q$-grams:
$\{[*, bought, *, *, John], [*, bought, *, John, car],$
$[*, bought, John, car, *], [*, bought, car, *, *]\}$.
By generating all possible $p, q$-grams for two trees we obtain a profile $\mathrm{P}^{p,q}(t)$ for a given tree $t$. Given a profile for each tree, the similarity score for two trees $t, h$ is computed as follows:

$$adist(t, h) = 1 - 2\frac{\mid \mathrm{P}^{p,q}(t) \cap \mathrm{P}^{p,q}(h) \mid}{\mid \mathrm{P}^{p,q}(t) \cup \mathrm{P}^{p,q}(h) \mid} \quad (9)$$

## 4 Combining word-based and tree-based approaches

After obtaining all the scores for the word-based and the tree-based approaches, we used these features for training an SVM. In addition to the similarity scores, we added the length of the text and hypothesis as well as the subtask as additional features. We used WEKA for training the SVM using the standard polynomial kernel provided by this machine learning environment (Witten and Frank, 2005).

## 5 Results

We submitted two runs varying the two different word-based methods. The first run uses the word-based origin vector approach (OVA) combined with the two tree-based approaches *adist* and *fdist*. The second run uses the word-based two vector approach (TVA) again combined with the two tree- based approaches. The results for the test set can be found in figure 2.

The results are within the range of accuracy results obtained by systems for last year's challenge

| | **All** | IE | IR | QA | SUM |
|---|---|---|---|---|---|
| RUN 1 | | | | | |
| Acc | **0.5437** | 0.405 | 0.55 | 0.605 | 0.615 |
| RUN 2 | | | | | |
| Acc | **0.5550** | 0.425 | 0.58 | 0.590 | 0.625 |

Table 2: Results for test set Run1 and Run 2

(i.e. best participating system: 0.586). In particular, the results for the subtasks QA and SUM seem to be encouraging. On the other hand, the result for IE is far below the baseline (i.e. 0.5).

## 6 Analysis

Our assumption that the word-based approaches combined with the tree-based ones would lead to an improvement of the entire accuracy did turn out to be only partly true. Moreover, the results for the test set turned out to be quite a bit lower than what we had obtained for the training set. The accuracy for the training set can be found in figure 3.

| | **All** | IE | IR | QA | SUM |
|---|---|---|---|---|---|
| RUN 1 | | | | | |
| Acc | **0.634** | 0.600 | 0.550 | 0.635 | 0.830 |
| RUN 2 | | | | | |
| Acc | **0.650** | 0.605 | 0.585 | 0.655 | 0.815 |

Table 3: Results for development set

A closer look reveals that the drop in performance is highly subtask-dependent. Whereas IE and SUM show a very high drop in accuracy (i.e. up to 0.215), IR and QA showed virtually no or only a modest decrease in accuracy (i.e. 0.064).

The bad performance for the IE subtask may result from the fact that the tree-based approaches match the root of the parse trees which works fine for the following training example, where $T$ does not entail $H$:

$T$**:** ECB spokeswoman, Regina Schueller, declined to comment on a report in Italy's La Repubblica newspaper that the ECB council will discuss Mr. Fazio's role in t he takeover fight at its Sept. 15 meeting.

| Approach | Acc. |
|---|---|
| OVA | 0.5162 |
| TVA | 0.5225 |
| ADIST | 0.5812 |
| FDIST | 0.6050 |
| A-FDIST | 0.6275 |
| ADIST + FDIST + OVA (RUN 1) | 0.6337 |
| ADIST + FDIST + TVA (RUN 2) | 0.6500 |

Table 4: Individual results for the development set

$H$**:** Regina Shueller works for Italy's La Repubblica newspaper.

The test set, on the other hand, seemed to contain many example where the $H$ had to be inferred from a subtree of the parse tree (e.g. a relative clause).

$T$**:** The British ambassador to Egypt, Derek Plumbly, told Reuters on Monday that authorities had compiled the list of 10 based on lists from tour companies and from families whose relatives have not been in contact since the bombings.

$H$**:** Derek Plumbly resides in Egypt.

Since *told* and *reside* bear a high penalty, our system does not derive the entailment relation via the tree-based approaches nor do the word-based approaches establish a semantic similarity between this very short $H$ and the relatively long $T$.

The prediction we made that a combination of different approaches to semantic similarity would be beneficial was not true for the test set, even though the results for the development set seemed to indicate that. The results shown in table 4 describe an improvement of system's accuracy when different approaches are combined.

The same numbers obtained after submitting our two runs, however, show a different picture for the test set. The results of the individual approaches in table 5 prove that a combination of word-based and tree-based similarity measures did actually hurt accuracy. The combined tree-based approach (A-FDIST) would have produced a better performance for the test set (i.e. 0.5627).

Analyzing the results for the training set, we noticed that the length feature for $T$ and $H$ alone could

give surprisingly good results. A tree stump trained only on these two features obtained an accuracy of 0.5738. This seems to be an artifact that should be avoided by producing balanced training and test sets wrt. sentence length in the future.

| Approach | Acc. | IE | IR | QA | SUM |
|---|---|---|---|---|---|
| OVA | 0.513 | 0.490 | 0.505 | 0.520 | 0.535 |
| TVA | 0.515 | 0.485 | 0.550 | 0.495 | 0.530 |
| ADIST | 0.543 | 0.500 | 0.495 | 0.580 | 0.595 |
| FDIST | 0.539 | 0.490 | 0.570 | 0.485 | 0.610 |
| A-FDIST | 0.563 | 0.450 | 0.605 | 0.590 | 0.620 |
| RUN 1 | 0.544 | 0.405 | 0.550 | 0.605 | 0.615 |
| RUN 2 | 0.555 | 0.425 | 0.580 | 0.590 | 0.625 |

Table 5: Individual results for the test set

## References


Nikolaus Augsten, Michael H. Böhlen, and Johann Gamper. 2005. Approximate matching of hierarchical data using pq-grams. In *Proceedings of Very Large Data Bases (VLDB) Conference*, pages 301–312. ACM Press.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages pp. 13–18, Ann Arbor, Michigan.

Christiane Fellbaum. 1998. *WordNet – An electronic lexical database*. MIT Press, Cambridge, Massachusetts and London, England.

D. Conrath J. Jiang. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on International Conference on Research in Computational Linguistics, Taiwan*, pages pp. 19–33.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In Oren Glickman Ido Dagan and Bernardo Magnini, editors, *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

C. Leacock and M. Chodorow. 1998. Combining local context and wordnet similarity for word sense identifi - cation. pages 265–283.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM Press.

Dekang Lin. 1998a. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May.

Dekang Lin. 1998b. An information-theoretic defi nition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA.

J. Michelizzi. 2005. Semantic relatedness applied to all words sense disambiguation. Master's thesis, University of Minnesota, Duluth, Duluth.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Demonstration Papers*, pages 38–41, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453.

D. Shasha and K. Zhang. 1989. Fast parallel algorithms for the unit cost editing distance between trees. In *SPAA '89: Proceedings of the first annual ACM symposium on Parallel algorithms and architectures*, pages 117–126, New York, NY, USA. ACM Press.

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

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection.