Recognizing Textual Entailment
Is lexical similarity enough?

Valentin Jijkoun and Maarten de Rijke

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: jijkoun,mdr@science.uva.nl

Abstract. We describe the system we used at the PASCAL-2005 Recognizing Textual Entailment Challenge. Our method for recognizing entailment is based on calculating "directed" sentence similarity: checking the directed "semantic" word overlap between the text and the hypothesis. We use frequency-based term weighting in combination with two different lexical similarity measures.
Although one version of the system shows significant improvement over randomly guessing decisions (with an accuracy score of 57.3), we show that this is only due to a subset of the data that can be equally well handled by simple word overlap. Furthermore, we give an in-depth analysis of the system and the data of the challenge.

1 Introduction

The Recognizing Textual Entailment (RTE) challenge, which is organized within the PASCAL network (Pascal), is a task where systems are required to detect semantic entailment between pairs of natural language sentences. For example, the sentence

- *The memorandum noted the United Nations estimated that 2.5 million to 3.5 million people died of AIDS last year*

is considered to logically entail the sentence

- *Over 2 million people died of AIDS last year.*

While the recognition of textual entailment is not an end-to-end task in itself, it is generally felt that robust entailment checkers have the potential of improving the performance of systems for a variety of end-to-end tasks, including reading comprehension, question answering, information extraction, machine translation, and paraphrase acquisition.

In principle, the RTE challenge offers opportunities for a broad spectrum of techniques, ranging from shallow baseline approaches based on word overlap and lexical similarity measures well-known from the field of information retrieval to methods based on deep natural language processing that require significant amounts of elaborate knowledge engineering. At the PASCAL-2005 RTE challenge the whole spectrum was represented; see (Dagan et al.). Our focus is on methods situated at the light-weight end of the scale. The main research aim for our participation in the PASCAL-2005 RTE challenge was to understand the potential and limitations of simple entailment checking methods based on lexical similarity. More specifically, in this paper we address the following issues:

- How well does a baseline entailment checker based on lexical similarity work? How much do similarity measures contribute to the performance?
- When determining whether a pair of sentences is a positive entailment instance, the similarity score between the two sentences needs to be above some threshold. How reliably can this threshold be estimated from development data?
- How well does our light-weight similarity measure separate positive and negative entailment examples?
- What are easy cases where lexical similarity methods are likely to succeed, and what are hard cases where they are likely to break down and where more elaborate methods are called for?

The remainder of the paper is organized as follows. In Section 2 we describe our system and provide details on the setting used for our experiments. Then, in Section 3 we compare several versions of the system and explore the contributions of its various components. In Sections 4–8 we describe more general and methodological issues, including thresholding, the distribution of positive and negative examples, and easy vs. hard cases for our system. We wrap up in Section 9.

2 System Description and Experimental Setting

At the Pascal-2005 RTE challenge, systems had to address the following task: given a pair of sentences T, H (text, hypothesis), determine whether T logically entails H and provide an estimate of the system's confidence. The example entailment pairs come from a number of natural language processing (NLP) areas: comparable documents (CD), reading comprehension (RC), question answering (QA), information extraction (IE), machine translation (MT), and paraphrase acquisition (PP). See (Dagan et al.) for further details.

To address the RTE challenge, we proceed as follows. For every text, hypothesis pair (T,H), we view each sentence as a bag of words and calculate a *directed sentence similarity score* between them. To check for entailment, we compare the score against a threshold. This method is implemented as shown in the pseudo-code in Figure 1. Essentially, for every word in the hypothesis H we find the most similar word in the text T according to the measure $\text{wordsim}(w_1,w_2)$. If such a similar word exists (i.e., *maxSim* is non-zero), we add the weighted similarity value to the total similarity score. Otherwise, we subtract the weight of the word, penalizing words in the hypothesis without matching words in the text.

The threshold for the final entailment checking is selected using the development corpus of text, hypothesis pairs (see Subsection 2.3). The confidence of a decision made by the system is determined by looking at the distance between the similarity value and the threshold. For example, for positive decisions ($sim \geq threshold$):

$$confidence = \frac{sim - threshold}{1 - threshold}$$

The algorithm is parametrized with two functions:

- $\text{weight}(w)$: the importance of the word w for the similarity identification;

```
let $T = (T_1, T_2, \ldots, T_n)$
let $H = (H_1, H_2, \ldots, H_m)$
$totalSim = 0$
$totalWeight = 0$
for $j = 1 \ldots m$ do
    $maxSim = \max_i \mathrm{wordsim}(T_i, H_j)$
    if $maxSim = 0$ then $maxSim = -1$
    $totalSim$ += $maxSim * \mathrm{weight}(H_j)$
    $totalWeight$ += $\mathrm{weight}(H_j)$
end for
$sim = totalSim / totalWeight$
if $sim \geq threshold$ then return TRUE
return FALSE
```

Fig. 1. Pseudo-code for our textual similarity method: determining whether the text T entails the hypothesis H.

– $\mathrm{wordsim}(w_1, w_2)$: the similarity between the two words w_1 and w_2 , with range $[0,1]$.

Next, we describe the choices we considered for these two functions.

2.1 Weighting words

The weighting of words with respect to importance is based on core intuitions from research in Information Retrieval, where Inverse Document Frequency (IDF) is often used as a measure of term importance; see e.g., (Baeza-Yates and Ribeiro-Neto, 1999). Recently, Monz and de Rijke (2001) used IDF for light-weight entailment checking in the setting of information fusion: merge information (i.e., text snippets) on a single topic but try to avoid redundancy, i.e., if a snippet entails another segment, only the entailing segment should be included in the fused information; in that paper, evaluation was done using a purpose-built corpus.

For our experiments in the present paper we use the normalized *inverse collection frequency* of words, calculated on a large collection of newspaper texts. That is, for a word w we compute

$$\mathrm{ICF}(w) = \log \frac{\text{\# occurences of all words}}{\text{\# occurences of } w},$$

and the actual weight of a word is calculated as normalized ICF, so that, for instance, the weight for the most frequent word (“*the*”) is 0.

2.2 Word similarity measures

We experimented with two similarity measures: Dekang Lin’s dependency-based word similarity (Lin, 1998) and the measure based on lexical chains in WordNet due to Hirst and St-Onge (1998). For both measures, words were first converted to lemmas.

We used both similiary measures for our official submission, as described in (Jijkoun and de Rijke). The dependency-based similarity measure performed somewhat better (accuracy 55.3 vs. 53.6). For this reason, we focus on Lin's dependency-based word similarity in the remainder of this paper.

2.3 Experimental setting

For the experiments described below, we used the material provided by the organizers of the Pascal-2005 RTE challenge: a development and test corpus, with 567 and 800 sentence pairs, respectively, manually annotated for logical entailment.

The evaluation measures used are accuracy (A), confidence-weighted score (CWS), as well as precision (P) and recall (R) for the entailment identification; see (Dagan et al.) for details.

3 Versions of the System

In this section we present and discuss several versions of our entailment checker. Our aim is to understand how well the lexical similarity-based system works and what the contribution of different components is, thus addressing the first of the research questions raised in the introduction.

The design of our system involves a number of important choices, whose effects are not obvious: (i) weighting words by importance, and (ii) using a word similarity measure. We want to determine whether the use of these techniques is justified.

In addition to these choices, we considered an option motivated by examples from the development corpus, like

T: Clinton's new book is not big seller here.
H: Clinton's book is a big seller.

Clearly, the text T does not entail the hypothesis H because of the presense of "*not*." We added a simple ad-hoc rule to the system, that checks for *not* or *n't* in both sentences of a pair, and rejects entailment if a particle is present in exacly one of the two sentences.

In our experiments we evaluated the following versions of the system:

- *M*: the main version, with word importance weighting, Lin's word similarity and the rule for handling *not*,
- *M-not*: the same but without the *not*-rule,
- *M-not-sim*: also without word similarity, and
- *M-not-sim-imp*: also without word weighting.

Note that the simplest version of the system, *M-not-sim-imp*, assigns entailment scores based solely on word overlap.

The results are presented in Table 1. There, we list the various flavors of our baseline system; the threshold values used as listed in row 2. Optimal thresholds were chosen so as to maximise accuracy on the development corpus.[1]

[1] As an aside, the system used to generate the official runs that were submitted for our participation in the Pascal-2005 RTE challenge (*M-not*) actually showed an accuracy score of 55.3; due

	M	*M-not*	*M-not-sim*	*M-not-sim-imp*
optimal threshold	0.4	0.4	0.2	0.3
accuracy on development corpus	58.2	56.6	57.1	57.0
accuracy on test corpus	57.3	57.1	54.4	54.3
precision on test corpus	55.1	54.7	53.0	53.3
recall on test corpus	78.8	83.5	76.3	69.3

Table 1. Accuracy, precision, and recall scores for (different flavors of) our baseline system.

Interestingly, in Table 1 we see that the more "elaborate" system *M* outperformseach of its subsystems, both on the development corpus and on the test corpus with automatically selected threshold. Looking at the accuracy scores on the test corpus, we see that each component of the main system *M* adds to the overall score, weighting helps (54.4 vs. 54.3), word similarity helps (57.1 vs. 54.4) and the *not*-rule helps (57.3 vs. 57.1). Another thing worth noting is that the simplest system, *M-not-sim-imp*, does not perform significantly better than random (which was the intention of the organizers Dagan et al.), while *M* does.

With respect to the 25 full runs submitted to the PASCAL-2005 RTE Challenge (Dagan et al.), both *M* and *M-not* (with accuracy scores of 57.3 and 57.1, respectively) perform above the median (55.2) and are only outperformed by the Web-based probabilistic system of Glickman et al. and the MT-based system of Bayer et al. (both with accuracy scores of 58.6). While this might be interpreted as a "success" for our simple methods, we interpret this outcome as an indication that deep language technology still faces very non-trivial challenges in recognizing textual entailment.

There are some further observations worth making. While differences in accuracy scores on the test corpus between the systems *M* and *M-not* are insignificant, their performance on the development corpus differs more substantially. However, in our further experiments with random splittings of the Pascal-2005 RTE collection into development and test data (see below), behavior of all versions of the system was similar on both corpora.

Summarizing our findings in this section, we claim that whereas simple word-overlap methods do not work well for the RTE task, they can be easily extended with simple weighting and word similarity measures, resulting in a system with a competitive performance.

4 Choosing a Threshold

Next, we turn to the second of our research questions from the introduction: How robust is the choice of thresholds? We approached this question from a number of angles.

to a bug, the threshold of 0.5 used there was selected based only on half of the development corpus. Had we used the entire development corpus for our official runs, the accuracy score would have been 57.1, as in Table 1, row 4.

System	Official	Min	Max	Median
M	57.3	54.9	57.8	57.0
M-not-sim-imp	54.3	52.5	56.5	55.1

Table 2. Accuracy scores based on alternative optimal thresholds: as estimated on the official development corpus (Official), and on 10 random splittings of the development and test corpus (Min, Max, Median).

To check how sensitive the different versions of the system are to varying corpora, we performed several experiments, splitting the entire collection randomly into development and test data, keeping the proportion of positive/negative examples and examples for the six subtasks as they were in the original split (i.e., in total 567 pairs for development and 800 pairs for testing). For each split and each version of the system, the optimal threshold was selected on the development data and then applied to the test data. The results, for the systems *M* and *M-not-sim-imp*, are presented in Table 2. While there is some variation in the resulting accuracy scores for *M*, all are significantly better than random at the 0.01 level (Dagan et al.). These experiments indicate that the system's behavior is consistent and that fine-tuning entailment thresholds on development data does generally produce good performance on test examples.

Our next observation concerns the performance on the development corpus vs. the performance on the test corpus: the former is not necessarily a good predictor of the latter. In particular, while simple subsystems (*M-not-sim* and *M-not-sim-imp*) perform reasonably well on the development corpus, their performance on the test corpus is substantially lower. In our experiments with random splittings, we observed a similar phenomena: whereas generally better performance on the development corpus led to better performance on the test data (with thresholds tuned on the development corpus), we were unable to establish strong statistical correlation (we used Spearman's rank correlation coefficient).

In an attempt to see how the choice of threshold depends on the choice of corpus, we looked at the performance of the versions of our system with different thresholds. Figure 2 shows the accuracy on the development and test data depending on a threshold, for the full system *M* (top) and the simplest subsystem *M-not-sim-imp* (bottom).

While for the simplest system, *M-not-sim-imp*, thresholds optimal for the development corpus are clearly suboptimal for the test corpus (the peaks in accuracy are located at different values of the threshold), for the full system, *M*, the correlation is very high. This does indeed indicate that for simple overlap (*M-not-sim-imp*) the optimal threshold is highly corpus-dependent, but that the choice is quite consistent in the more complex system (*M*). That is, *M*'s reasonable performance is not an accident.

We have not systematically investigated how the size of the development corpus affects the quality of threshold, but anecdotal evidence (the bug in our official submission, see footnote 1) suggests that the size of the development corpus is an important issue, and that at least several hundreds of pairs are necessary for training.

Finally, we hypothesize that the optimal thresholds depend on the source of the examples, i.e., they may be different for the seven subtasks (CD, IE, MT, QA, RC, PP, IR).

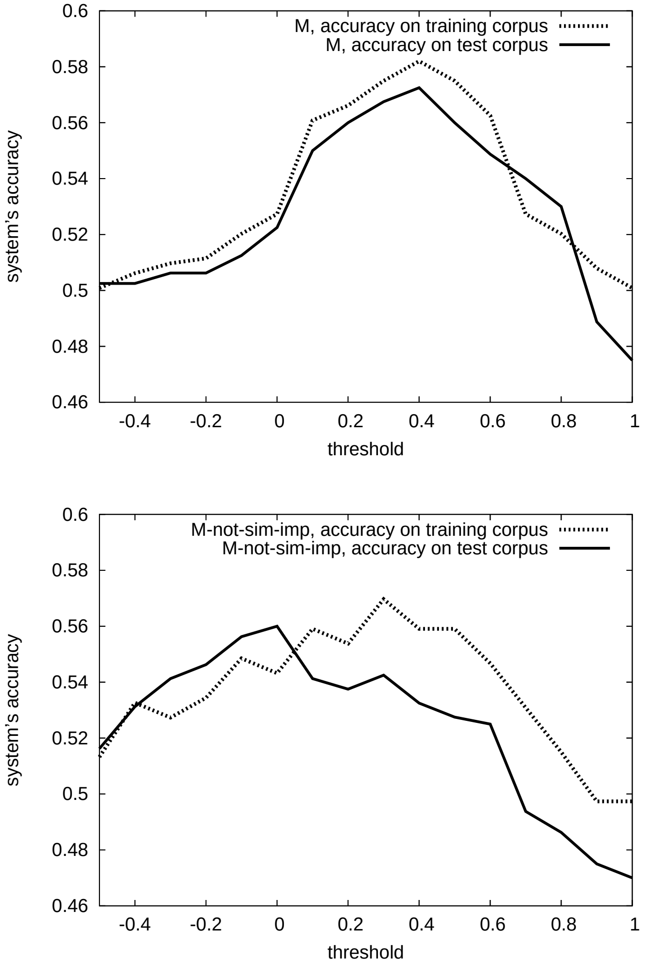

Fig. 2. (Top): thresholds on the main system *M*. (Bottom): thresholds on the simplest system *M-not-sim-imp*. The horizontal axis shows possible thresholds, and the vertical axis—accuracy of a system.

However, since currently only 50–100 entailment pairs are available for development per subtask, it is difficult to support this claim experimentally at this time.

5 The Distribution of Positive and Negative Examples

Every system that makes an entailment decision based on a threshold of some similarity score between the text and the hypothesis (e.g., most systems in the PASCAL-2005 RTE

Challenge) is based on the assumption that the similarity scores somehow separate negative and positive examples. Ideally, for a good variant of a similarity scoring method, negative examples would mostly have low scores and positive examples—mostly high scores.

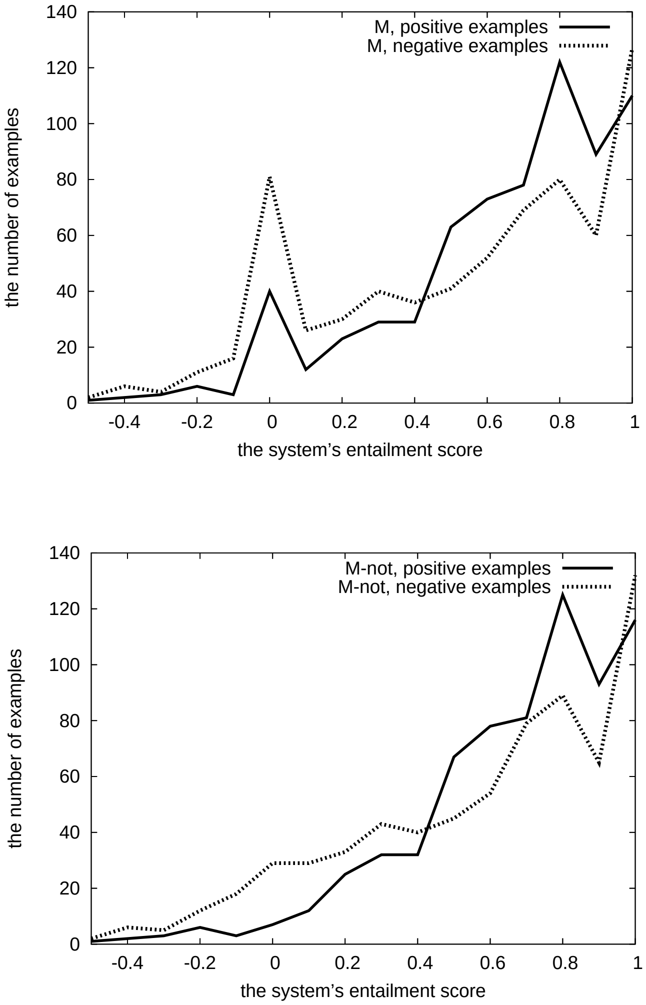

Fig. 3. Distribution of true positive and negative examples. (Top): for the full system *M*. (Bottom): for the system without the *not*-rule (*M-not*). The horizontal axis gives the possible values of the system's entailment score, and the vertical axis shows the number of pairs (positive vs. negative) with this entailment score.

To see whether this is indeed the case for our entailment checkers, we plotted the number of true positive and true negative entailment pairs that were assigned different scores by the system. Figure 3 shows the results for the systems *M* and *M-not* on the full corpus of 1367 entailment pairs.

Quite surprisingly, for the word-similarity based system *M-not* (Figure 3(Bottom)) the distributions of positive and negative examples with respect to the score of the system are very similar and the graphs have peaks near the same values. Most negative examples are "concentrated" around the same entailment score as positive examples. Moreover, there are actually more negative than positive examples with entailment score 1. This means that the system does not really manage to separate positive and negative examples, but simply uses the fact that the distribution of negatives is somewhat "flatter": the peak around score 0.8 is lower and some mass is moved left, to lower values. It seems that the "only" reason the system shows a performance that is better than random is that the distribution of the negative examples with respect to weighted word overlap is flatter than the distribution of the positive examples (except for the high peak in both distributions around score 1.0).

Note that the situation is somewhat different when we include our simple *not*-rule (Figure 3(Top)). Now, negative examples have a second clear peak around 0 (this is exactly the entailment score assigned by the *not*-rule). Apart from improving accuracy, it seems that the *not*-rule actually does something reasonable, providing for a somewhat clearer separation between positive and negative examples.

As an aside, for the other, simpler subsystems (*M-not-sim* and *M-not-sim-imp*), the slopes of the graphs are even flatter and the two curves are even closer together, making it even more difficult to separate positive and negative examples.

In sum, we conclude that in general, the similarity-based system *M* fails to actually separate positive and negative examples of the entailment pairs: their distributions with respect to the system's score are very similar. A more substantial separation is only achieved using the ad-hoc *not*-rule.

6 Easy vs. Hard Cases

Ideally, we would want to use our lexical similarity-based system to identify entailment pairs that are "hard" for a purely lexical systems, i.e., where more sophisticated analysis (syntactic relations, reasoning with world knowledge) is required. Can we use a variant of our entailment checking methods to find such "hard" cases?

Unfortunately, the answer seems to be "no." As the curves in Figure 3 indicate, there is no single region among possible entailment scores with a substantial number of TE examples and high confidence of the system (i.e., mostly positive or mostly negative examples). As mentioned previously, the distributions of positive and negative examples are fairly similar. The best observation we were able to make is that among TE pairs with scores less than 0.1 (216 pairs of 1367, or 16%), as much as 69% of the pairs were negative entailment examples. Still, we believe that the accuracy 0.69 is not high enough to consider these examples as "easy."

For now it seems that we need a different way of identifying "easy" vs. "hard" cases—a reliable category of "easy" examples is identified in the next section.

7 Performance on Different Subtasks

We also compared the performance of our entailment checking system on different subtasks, reflecting the different sources from which the entailment pairs were selected by the task organizers. The table below shows the accuracy, precision and recall of the system *M* for all subtasks:

Subtask	Accuracy	Precision	Recall
CD	84.0	84.9	82.7
IE	59.2	55.2	96.7
MT	45.8	46.8	60.0
QA	46.2	47.0	60.0
RC	52.1	51.2	92.9
PP	56.0	53.5	92.0
IR	50.0	50.0	71.1
Overall	57.3	55.0	78.8

From the table it is clear that the overall accuracy of the system is relatively high only due to the reasonable performance on the CD (comparable documents) subtask. This particular subtask appears to be quite easy for our system, whereas on other tasks the performance is not better than randomly guessing. Manual examination of the entailment candidate pairs from the CD subtask shows that the pairs usually have many words in common. Here are two examples:

(T) Voting for a new European Parliament was clouded by concerns over apathy.
(H) Voting for a new European Parliament has been clouded by apathy.
Entailment: TRUE, System's score: 0.88

(T) A small bronze bust of Spencer Tracy sold for $174,000.
(H) A small bronze bust of Spencer Tracy made $180,447.
Entailment: FALSE, System's score: 0.44

In the second example the similarity is substantially lower since the numbers (which occur relatively rarely in our newspaper collection, and thus get higher weight) are different.

In our subsequent analysis, we found that even the subsystem *M-not-sim-imp* (simple word overlap) performed well on the CD subtask, with an accuracy score of 86.0. This suggests that examples from the CD task can indeed be considered "easy" and that they probably need not be included in future editions of the RTE task.

When CD examples were removed from the development and testing corpora, the system did not perform better than random (accuracy 51.2). We interpret this as a good sign: examples from other subtasks, apparently, require other, deeper methods of entailment recognition.

8 Precision and Recall

For all subtasks, except CD, our precision scores are substantially worse than our recall scores. The system *M* judged 72% of the test pairs as positive, compared to 50% true positives in the test set. This comes as no surprise: since most examples have entailment scores larger than the selected threshold (see Figure 3), most errors are also in this “positive” area, thus most errors are false positives.

In many classification problems thresholds can be used to fine-tune the precision-recall balance, which is obviously a very useful option for any real-world application. However, we found that for our system precision on the test data cannot be improved by changing the threshold. This is due to the great uncertainty for large values of the entailment score (Figure 3) and unseparability of positive and negative examples mentioned above.

9 Conclusions

We described a system for recognizing textual entailment based on lexical similarity. Although the system performs significantly better than randomly guessing, the reasonable performance is only based on one subtask (CD, comparable documents). For this subtask even much simpler systems (viz. plain word overlap) give similar performance. For all other subtasks none of the variants of our system performed better than random. Moreover, we found that the system cannot be further tuned without overfitting, which indicates that other, deeper textual features need to be explored.

Acknowledgments

Both authors were supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. In addition, Maarten de Rijke was supported by grants from NWO, under project numbers 017.001.190, 264-70-050, 365-20-005, 612.000.106, 612.000.207, 612.066.302, 612.069.006, and 640.001.501

10 References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. MITRE’s submission to the EU Pascal RTE challenge.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognizing Textual Entailment Challenge.

Oren Glickman, Ido Dagan, and Moshe Koppel. Web based probabilistic textual entailment.

Graeme Hirst and David St-Onge. Lexical chains as representation of context for the detection and correction of malapropisms. In Fellbaum Christiane, editor, *WordNet: An electronic lexical database*. The MIT Press, 1998.

Valentin Jijkoun and Maarten de Rijke. Recognizing textual entailment using lexical similarity.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, 1998.

Christof Monz and Maarten de Rijke. Light-weight entailment checking for computational semantics. In *Proceedings of the Workshop on Inference in Computational Semantics (ICoS-3)*, 2001.

Pascal. Pascal Recognising Textual Entailment Challenge, 2005. URL: http://www.pascal-network.org/Challenges/RTE/.