Recognizing Textual Entailment Using Lexical Similarity

Valentin Jijkoun and Maarten de Rijke
Informatics Institute, University of Amsterdam
jijkoun,mdr@science.uva.nl

Abstract

We describe our participation in the PASCAL-2005 Recognizing Textual Entailment Challenge. Our method 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. Our best run shows 0.55 accuracy on the test data, although the difference between our two runs is not significant. We found remarkably different optimal threshold values for the development and test data. We argue that, in addition to accuracy, precision and recall are valuable measures to consider for textual entailment.

1 Introduction

Recognizing Textual Entailment Challenge, organized within the PASCAL network, 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.*

The organizers of the entailment challenge provided participants with development and test corpora, with 567 and 800 sentence pairs, respectively, manually annotated for logical entailment.

In this paper we describe a simple system based on lexical similarity, with two different word similarity measures. We also present our official results and a deeper analysis of the system's performance.

2 System Description

For every text/hypothesis pair (T, H), we consider each sentence a bag of words and calculate *directed sentence similarity score*. To check for entailment, we compare the score against a threshold. This method is implemented as shown in the pseudo-algorithm below.

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

Essentially, for every word in the hypothesis we find the most similar word in the text according to the measure wordsim(w_1, w_2). If such a similar word exists (*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. The confidence of a system's decision is determined by looking at the distance between the similarity value and the threshold. For example, for positive decisions (*sim* $\geq$ *threshold*):

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

The algorithm is parametrized with two functions:

- weight(w): importance of the word for the similarity identification;
- wordsim(w_1, w_2): similarity between two words, with range $[0, 1]$.

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. Recently, IDF was used for the light-weight entailment checking in (Monz and de Rijke, 2001). For our experiments we used *normalized inverse collection frequency* of words, calculated on a big collection of newspaper texts. For a word w:

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

and

$$\text{weight}(w) = 1 - \frac{\text{ICF}(w) - \text{ICF}_{\min}}{\text{ICF}_{\max} - \text{ICF}_{\min}}.$$

The minimum and maximum of the inverse frequencies ($\text{ICF}_{\min}$ and $\text{ICF}_{\max}$) are used to normalize weights between 0 and 1.

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 (Hirst and St-Onge, 1998). For both measures, words were first converted to lemmas.

3 Results

We submitted two runs that differ in the word similarity measures they use: sim-lin and sim-wn. The table below summarizes the results on the test and development corpora: accuracy (A), confidence-weighted score (CWS), and also precision (P) and recall (R) for the entailment identification.

Run	A	CWS	P	R
Test corpus:				
sim-lin	55.3	55.9	53.7	75.5
sim-wn	53.6	55.3	53.4	56.5
Development corpus:				
sim-lin	61.0	64.9	57.6	81.8
sim-wn	63.4	67.4	61.6	70.6

For our two official runs, sim-lin performed significantly better than random at the 0.01 level, and sim-wn better than random at the 0.05 level.

4 Discussion

The evaluation scores are better on the development corpus than on the test corpus. This is expected since the thresholds were selected on the development corpus. However, a more detailed analysis shows that the differences between the evaluations on the test and development data are not only due to the choice of thresholds. Figure 1 shows how the performance of the system changes when the thresholds are changed from 0.1 to 0.9. We give evaluation results for both our methods and also for a simple baseline that only considers lexical overlap, without WordNet and frequency information.

Surprisingly, the performance of the system on the test corpus (thick lines) is substantially worse than on the development corpus even if optimal similarity thresholds are taken. It is not clear whether this is due to the test corpus being more "difficult," or our system overfits the development corpus in ways other than threshold selection.

Another important observation is that the optimal threshold values differ substantially for different corpora: 0.20–0.4 for the test corpus and 0.6–0.7 for the development corpus. Moreover, whereas the difference between the two similarity measures seems substantial on the development corpus, they perform very similarly on the test corpus. For these reasons, we find it impossible to tell which of the measures is

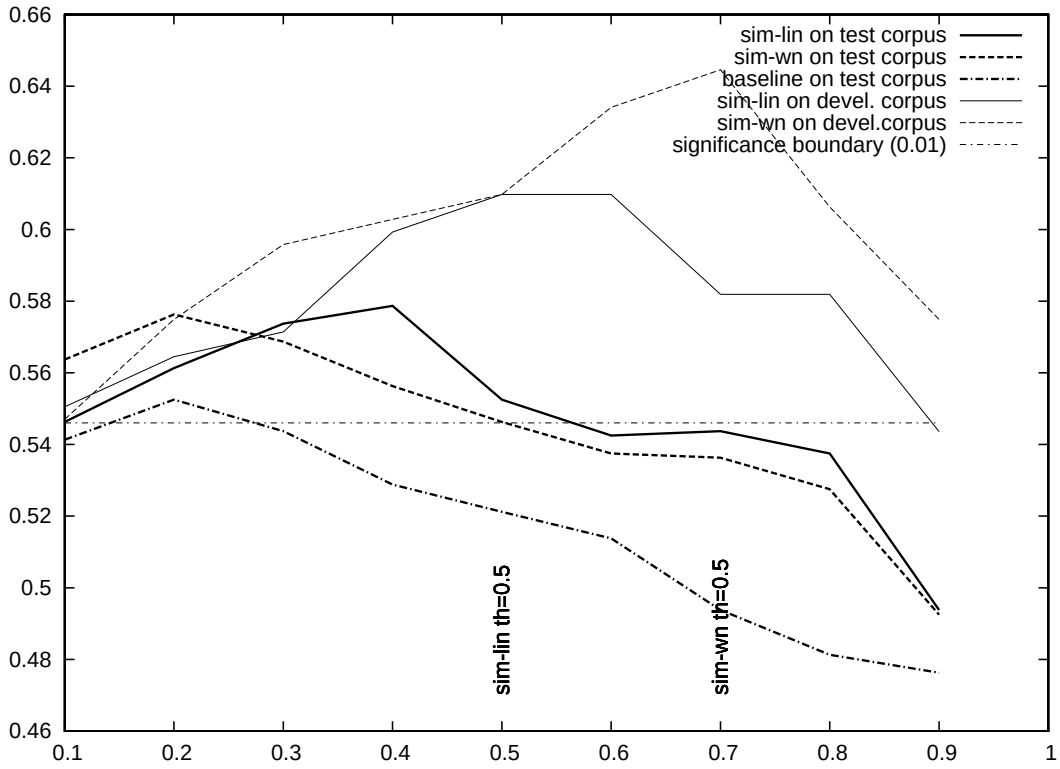

Figure 1: Performance of similarity measures with different thresholds. Thick lines show the performance on the test corpus. The thresholds optimal for the development corpus are clearly not optimal for the test corpus.

better for the task, and how to select thresholds in a robust way.

We also compared the performance of our entailment checking system on different subtasks, corresponding to different sources of the entailment pairs. The table below shows the accuracy, precision and recall for the sim-lin run for all subtasks.[1]

Subtask	A	P	R
CD	84.7	74.7	93.3
IE	55.0	95.0	52.8
MT	46.7	63.3	47.5
QA	42.3	53.9	43.8
RC	49.3	88.6	49.6
PP	42.0	80.0	45.5
IR	53.3	75.6	52.3
Overall	55.3	75.5	53.7

[1]Recall that the identifiers for the substasks have the following readings: comparable documents (CD), reading comprehension (RC), question answering (QA), information extraction (IE), machine translation (MT), and paraphrase acquisition (PP).

From the table it is clear that the overall accuracy of the system is relatively high only due to the resonable performance on the CD subtask. This particular subtask appears to be quite easy for our system, whereas on other tasks the performance is close to (or worse than) that of the random guessing. Manual examination of the entailment candidate pairs from the CD subtask shows that the pairs usually have many words in common:

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, Similarity: 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, Similarity: 0.44

In the second example the similarity is substantially lower since numbers (which occur relatively rarely in our newspaper collection, and thus get higher weight) are different. We have not checked whether a simple word overlap baseline would give a reasonable performance for the CD subtask.

Note that we give precision (P) and recall (R) scores as well as accuracy. We believe that P and R help us to better understand the behavior of our algorithms in ways that accuracy does not. For instance, for all subtasks, except CD, precision is substantially higher than recall. This can be explained by the fact that our lexical similarity resources are far from complete and we are not trying to detect various complex types of paraphrasing (e.g., syntactic). Our method seems very cautious: it prefers to reject the entailment if it cannot find simple lexical evidence to support it. Although, in principle, we can tune the precision/recall balance by varying the thresholds, the experimental results on which we report in this note show that the thresholds are very corpus-specific and thus can hardly be used for this tuning.

5 Conclusions

We described our participation in the PASCAL-2005 Recognizing Textual Entailment Challenge, with a simple sentence similarity-based system that uses two different word similarity measures. Although both our runs show significant improvement over random guessing, the improvement is based only on one subtask (CD). We found that the system cannot be further tuned without overfitting, which suggests that other, deeper text 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 365-20-005, 612.069.006, 220-80-001, 612.000.106, 612.000.207, 612.066.302, 264-70-050, and 017.001.190.

References

Graeme Hirst and David St-Onge. 1998. 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.

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

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