Textual Entailment Through Extended Lexical Overlap and Lexico-Semantic Matching

Rod Adams, Gabriel Nicolae, Cristina Nicolae and Sanda Harabagiu
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, Texas
{rod, gabriel, cristina, sanda}@hlt.utdallas.edu

Abstract

This paper presents two systems for textual entailment, both employing decision trees as a supervised learning algorithm. The first one is based primarily on the concept of lexical overlap, considering a bag of words similarity overlap measure to form a mapping of terms in the hypothesis to the source text. The second system is a lexico-semantic matching between the text and the hypothesis that attempts an alignment between chunks in the hypothesis and chunks in the text, and a representation of the text and hypothesis as two dependency graphs. Their performances are compared and their positive and negative aspects are analyzed.

1 Introduction

Textual entailment is the task of taking a pair of passages, referred to as the *text* and the *hypothesis*, and labeling whether or not the hypothesis (H) can be fully inferred from the text (T), as is illustrated in Pair 1. In Pair 1, the knowledge that an attorney representing someone's interests entails that they work for that person.

Pair 1 (RTE2 IE 58)

T: "A force majeure is an act of God," said attorney Phil Wittmann, who represents the New Orleans Saints and owner Tom Benson's local interests.
H: Phil Wittmann works for Tom Benson.

The Third PASCAL Recognizing Textual Entailment Challenge[1] follows the experience of the second challenge (Bar-Haim et al., 2006), whose main task was to automatically detect if a hypothesis H is entailed by a text T. To increase the "reality" of the task, the text-hypothesis examples were taken from outputs of actual systems that solved applications like Question Answering (QA), Information Retrieval (IR), Information Extraction (IE) and Summarization (SUM).

In the challenge, there are two corpora, each consisting of 800 annotated pairs of texts and hypotheses. Pairs are annotated as to whether there exists a positive entailment between them and from which application domain each example came from. Instances are distributed evenly among the four tasks in both corpora, as are the positive and negative examples. One corpus was designated for development and training, while the other was reserved for testing.

In the Second PASCAL RTE Challenge (Bar-Haim et al., 2006), one of the best performing submissions was (Adams, 2006), which focused on strict lexical methods so that the system could remain relatively simple and be easily applied to various entailment applications. However, this simple approach did not take into account details like the syntactic structure, the coreference or the semantic relations between words, all necessary for a deeper understanding of natural language text. Thus, a new system, based on the same decision tree learning algorithm, was designed in an attempt to gain performance by adding alignment and dependency relations information. The two systems will be compared and their advantages and disadvantages discussed.

[1] http://www.pascal-network.org/Challenges/RTE3/

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 119–124,
Prague, June 2007.

This paper is organized as follows: The first system is discussed in Section 2, followed by the second system in Section 3. The experimental results are presented in Section 4, and the paper concludes in Section 5.

2 Textual entailment through extended lexical overlap

The first system (Adams, 2006) follows a four step framework. The first step is a tokenization process that applies to the content words of the text and hypothesis. The second step is building a "token map" of how the individual tokens in the hypothesis are tied to those in the text, as explained in Section 2.1. Thirdly, several features, as described in Section 2.2, are extracted from the token map. Finally, the extracted features are fed into Weka's (Witten and Frank, 2005) J48 decision tree for training and evaluation.

2.1 The token map

Central to this system is the concept of the token map. This map is inspired by (Glickman et al., 2005)'s use of the most probable lexical entailment for each hypothesis pair, but has been modified in how each pair is evaluated, and that the mapping is stored for additional extraction of features. The complete mapping is a list of (H_i, T_j) mappings, where H_i represents the i^{th} token in the hypothesis, and T_j is similarly the j^{th} token in the text. Each mapping has an associated similarity score. There is one mapping per token in the hypothesis. Text tokens are allowed to appear in multiple mappings.

The mappings are created by considering each hypothesis token and comparing it to each token in the text and keeping the one with the highest similarity score.

Similarity scores A similarity score ranging from 0.0 to 1.0 is computed for any two tokens via a combination of two scores. This score can be thought of as the probability that the text token implies the hypothesis one, even though the methods used to produce it were not strictly probabilistic in nature.

The first score is derived from the cost of a WordNet (Fellbaum, 1998) path. The WordNet paths between two tokens are built with the method reported in (Hirst and St-Onge, 1998), and designated as $Sim_{WN}(H_i, T_j)$. Exact word matches are always given a score of 1.0, words that are morphologically related or that share a common sense are 0.9 and other paths give lower scores down to 0.0. This method of obtaining a path makes use of three groups of WordNet relations: *Up* (e.g. hypernym, member meronym), *Down* (e.g. hyponym, cause) and *Horizontal* (e.g. nominalization, derivation). The path can only follow certain combinations of these groupings, and assigns penalties for each link in the path, as well as for changing from one direction group to another.

The secondary scoring routine is the lexical entailment probability, $lep(u, v)$, from (Glickman et al., 2005). This probability is estimated by taking the page counts returned from the *AltaVista*[2] search engine for a combined u and v search term, and dividing by the count for just the v term. This can be precisely expressed as:

$$\mathrm{Sim}_{AV}(H_i, T_j) = \frac{\mathrm{AVCount}(H_i \,\&\, T_j)}{\mathrm{AVCount}(T_j)}$$

The two scores are combined such that the secondary score can take up whatever slack the dominant score leaves available. The exact combination is:

$$\begin{aligned}\mathrm{Sim}(H_i, T_j) &= \mathrm{Sim}_{WN}(H_h, T_t) \\ &\quad + \alpha \cdot (1 - \mathrm{Sim}_{WN}(H_h, T_t)) \cdot \mathrm{Sim}_{AV}(H_h, T_t)\end{aligned}$$

where α is a tuned constant ($\alpha \in [0, 1]$). Empirical analysis found the best results with very low values of α[3]. This particular combination was chosen over a strict linear combination, so as to more strongly relate to Sim_{WN} when it's values are high, but allow Sim_{AV} to play a larger role when Sim_{WN} is low.

2.2 Feature extraction

The following three features were constructed from the token map for use in the training of the decision tree, and producing entailment predictions.

Baseline score This score is the product of the similarities of the mapped pairs, and is an extension of (Glickman et al., 2005)'s notion of $P(H|T)$. This

[2] http://www.av.com

[3] The results reported here used $\alpha = 0.1$

is the base feature of entailment.

$$Score_{BASE} = \prod_{(H_i,T_j)\in Map} \text{Sim}(H_i, T_J)$$

One notable characteristic of this feature is that the overall score can be no higher than the lowest score of any single mapping. The failure to locate a strong similarity for even one token will produce a very low base score.

Unmapped negations A token is considered unmapped if it does not appear in any pair of the token map, or if the score associated with that mapping is zero. A token is considered a negation if it is in a set list of terms such as `no` or `not`. Both the text and the hypothesis are searched for unmapped negations, and total count of them is kept, with the objective of determining whether there is an odd or even number of them. A (possibly) modified, or flipped, score feature is generated:

$$n = \text{\# of negations found.}$$

$$Score_{NEG} = \begin{cases} Score_{BASE} & \text{if } n \text{ is even,} \\ 1 - Score_{BASE} & \text{if } n \text{ is odd.} \end{cases}$$

Task The task domain used for evaluating entailment (i.e. IE, IR, QA or SUM) was also used as a feature to allow different thresholds among the domains.

3 Textual entailment through lexico-semantic matching

This second system obtains the probability of entailment between a text and a hypothesis from a supervised learning algorithm that incorporates lexical and semantic information extracted from WordNet and PropBank. To generate learning examples, the system computes features that are based upon the alignment between chunks from the text and the hypothesis. In the preliminary stage, each instance pair of text and hypothesis is processed by a chunker. The resulting chunks can be simple tokens or compound words that exist in WordNet, e.g., *pick up*. They constitute the lexical units in the next stages of the algorithm.

identity	1.0	coreference	0.8
synonymy	0.8	antonymy	-0.8
hypernymy	0.5	hyponymy	-0.5
meronymy	0.4	holonymy	-0.4
entailment	0.6	entailed by	-0.6
cause	0.6	caused by	-0.6

Table 2: Alignment relations and their scores.

3.1 Alignment

Once all the chunks have been identified, the system searches for alignment candidates between the chunks of the hypothesis and those of the text. The search pairs all the chunks of the hypothesis, in turn, with all the text chunks, and for each pair it extracts all the relations between the two nodes. Stop words and auxiliary verbs are discarded, and only two chunks with the same part of speech are compared (a noun must be transformed into a verb to compare it with another verb). The alignments obtained in this manner constitute a one-to-many mapping between the chunks of the hypothesis and the chunks of the text.

The following relations are identified: (a) identity (between the original spellings, lowercase forms or stems), (b) coreference and (c) WordNet relations (synonymy, antonymy, hypernymy, meronymy, entailment and causation). Each of these relations is attached to a score between -1 and 1, which is hand-crafted by trial and error on the development set (Table 2).

The score is positive if the relation from the text word to the hypothesis word is compatible with an entailment, e.g., identity, coreference, synonymy, hypernymy, meronymy, entailment and causation, and negative in the opposite case, e.g., antonymy, hyponymy, holonymy, reverse entailment and reverse causation. This is a way of quantifying intuitions like: *"The cat ate the cake"* entails *"The animal ate the cake"*. To identify these relations, no word sense disambiguation is performed; instead, all senses from WordNet are considered. Negations present in text or hypothesis influence the sign of the score; for instance, if a negated noun is aligned with a positive noun through a negative link like antonymy, the two negations cancel each other and the score of the relation will be positive. The score of an alignment is the sum of the scores of all the

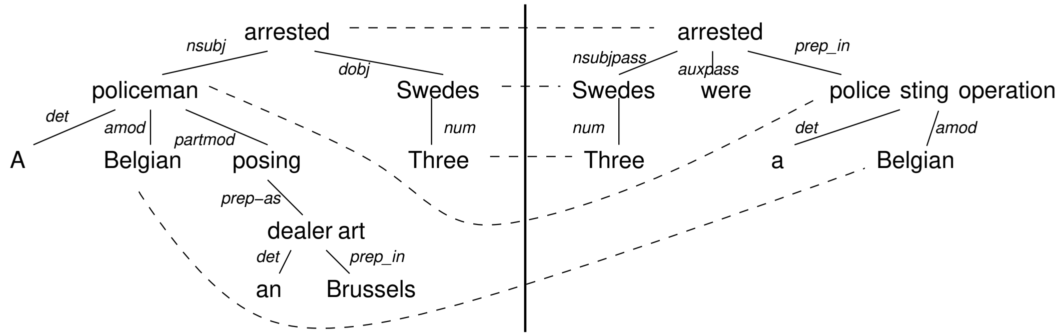

Figure 1: The dependency graphs and alignment candidates for Pair 2 (RTE3 SUM 633).

Category	***Feature name***	***Feature description***
alignment (score)	*totaligscore*	the total alignment score (sum of all scores)
	totminaligscore	the total alignment score when considering only the minimum scored relation for any two chunks aligned
	totmaxaligscore	the total alignment score when considering only the maximum scored relation for any two chunks aligned
alignment (count)	*allaligs*	the number of chunks aligned considering all alignments
	posaligs	the number of chunks aligned considering only positive alignments
	negaligs	the number of chunks aligned considering only negative alignments
	minposaligs	the number of alignments that have the minimum of their scores positive
	maxposaligs	the number of alignments that have the maximum of their scores positive
	minnegaligs	the number of alignments that have the minimum their scores negative
	maxnegaligs	the number of alignments that have the maximum of their scores negative
dependency	*edgelabels*	the pair of labels of non matching edges
	match	the number of relations that match when comparing the two edges
	nonmatch	the number of relations that don't match when comparing the two edges

Table 1: Features for lexico-semantic matching.

relations between the two words, and if the sum is positive, the alignment is considered positive.

3.2 Dependency graphs

The system then creates two dependency graphs, one for the text and one for the hypothesis. The dependency graphs are directed graphs with chunks as nodes, interconnected by edges according to the relations between them, which are represented as edge labels. The tool used is the dependency parser developed by (de Marneffe et al., 2006), which assigns some of 48 grammatical relations to each pair of words within a sentence. Further information is added from the predicate-argument structures in PropBank, e.g., a node can be the ARG0 of another node, which is a predicate.

Because the text can have more than one sentence, the dependency graphs for each of the sentences are combined into a larger one. This is done by collapsing together nodes (chunks) that are coreferent, identical or in an *nn* relation (as given by the parser). The relations between the original nodes and the rest of the nodes in the text (dependency links) and nodes in the hypothesis (alignment links) are all inherited by the new node. Again, each edge can have multiple relations as labels.

3.3 Features

With the alignment candidates and dependency graphs obtained in the previous steps, the system computes the values of the feature set. The features used are of two kinds (Table 1):

(a) The *alignment features* are based on the scores and counts of the candidate alignments. All the scores are represented as real numbers between -1 and 1, normalized by the number of concepts in the hypothesis.

(b) The *dependency features* consider each positively scored aligned pair with each of the other positively scored aligned pairs, and compare the set of

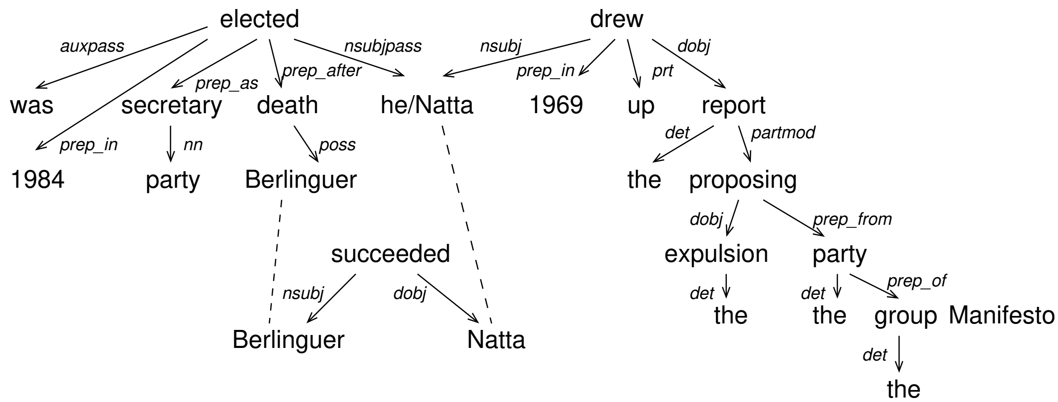

Figure 2: The dependency graphs and alignment candidates for Pair 3 (RTE3 IE 19).

relations between the two nodes in the text with the set of relations between the two nodes in the hypothesis. This comparison is performed on the dependency graphs, on the edges that immediately connect the two text chunks and the two hypothesis chunks, respectively. They have numerical values between 0 and 1, normalized by the square of the total number of aligned chunks.

3.4 Examples

Pair 2 (RTE3 SUM 633)

T: A Belgian policeman posing as an art dealer in Brussels arrested three Swedes.
H: Three Swedes were arrested in a Belgian police sting operation.

Figure 1 illustrates the dependency graphs and alignment candidates extracted for the instance in Pair 2. There is no merging of graphs necessary here, because the text is made up of a single sentence. The vertical line in the center divides the graph corresponding to the text from the one corresponding to the hypothesis. The dependency relations in the two graphs are represented as labels of full lines, while the alignment candidate pairs are joined by dotted lines. As can be observed, the alignment was done based on identity of spelling, e.g., *Swedes-Swedes*, and stem, e.g., *policeman-police*. For the sake of simplicity, the predicate-argument relations have not been included in the drawing. This is a case of a positive instance, and the dependency and alignment relations strongly support the entailment.

Pair 3 (RTE3 IE 19)

T: In 1969, he drew up the report proposing the expulsion from the party of the Manifesto group. In 1984, after Berlinguer's death, Natta was elected as party secretary.
H: Berlinguer succeeded Natta.

Figure 2 contains an example of a negative instance (Pair 3) that cannot be solved through the simple analysis of alignment and dependency relations. The graphs corresponding to the two sentences of the text have been merged into a single graph because of the coreference between the pronoun *he* in the first sentence and the proper name *Natta* in the second one. This merging has enriched the overall information about relations, but the algorithm does not take advantage of this. To correctly solve this problem of entailment, one needs additional information delivered by a temporal relations system. The chain of edges between *Berlinguer* and *Natta* in the text graph expresses the fact that the event of Natta's election happened after Berlinguer's death. Since the hypothesis states that Berlinguer succeeded Natta, the entailment is obviously false. The system presented in this section will almost certainly solve this kind of instance incorrectly.

4 Results

The experimental results are summarized in Tables 3 and 4. The first table presents the accuracy scores obtained by running the two systems through 10-fold crossvalidation on incremental RTE datasets. The first system, based on extended lexical overlap (ELO), almost consistently outperforms the second system, lexico-semantic matching (LSM),

Evaluation set	ELO	LSM	ELO+LSM	
	J48	J48	J48	JRip
RTE3Dev	66.38	63.63	65.50	67.50
+RTE2Dev	64.38	59.19	61.56	62.50
+RTE1Dev	62.11	56.67	60.36	59.62
+RTE2Test	61.04	57.77	61.51	61.20
+RTE1Test	60.07	56.57	59.04	60.42

Table 3: Accuracy for the two systems on various datasets.

Task	IE	IR	QA	SUM	All
Accuracy	53.50	73.50	80.00	61.00	67.00

Table 4: Accuracy by task for the Extended Lexical Overlap system tested on the RTE3Test corpus.

and the combination of the two. The only case when the combination gives the best score is on the RTE3 development set, using the rule-based classifier JRip. It can be observed from the table that the more data is added to the evaluation set, the poorer the results are. This can be explained by the fact that each RTE dataset covers a specific kind of instances. Because of this variety in the data, the results obtained on the whole collection of RTE datasets available are more representative than the results reported on each set, because they express the way the systems would perform in real-life natural language processing as opposed to an academic setup.

Since the ELO system was clearly the better of the two, it was the one submitted to the Third PASCAL Challenge evaluation. Table 4 contains the scores obtained by the system on the RTE3 testing set. The overall accuracy is 67%, which represents an increase from the score the system achieved at the Second PASCAL Challenge (62.8%). The task with the highest performance was Question Answering, while the task that ranked the lowest was Information Extraction. This is understandable, since IE involves a very deep understanding of the text, which the ELO system is not designed to do.

5 Conclusions

This paper has presented two different approaches of solving textual entailment: one based on extended lexical overlap and the other on lexico-semantic matching. The experiments have shown that the first approach, while simpler in concept, yields a greater performance when applied on the PASCAL RTE3 development set. At first glance, it seems puzzling that a simple approach has outperformed one that takes advantage of a deeper analysis of the text. However, ELO system treats the text naively, as a bag of words, and does not rely on any preprocessing application. The LSM system, while attempting an understanding of the text, uses three other systems that are not perfect: the coreference resolver, the dependency parser and the semantic parser. The performance of the LSM system is limited by the performance of the tools it uses. It will be of interest to evaluate this system again once they increase in accuracy.

References

Rod Adams. 2006. Textual entailment through extended lexical overlap. In *The Second PASCAL Recognising Textual Entailment Challenge (RTE-2)*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *PASCAL RTE Challenge*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web based probabilistic textual entailment. In *PASCAL RTE Challenge*.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. The MIT Press.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition.