Normalized alignment of dependency trees for detecting textual entailment

Erwin Marsi, Emiel Krahmer
Communication & Cognition
Tilburg University
The Netherlands
e.c.marsi@uvt.nl
e.j.krahmer@uvt.nl

Wauter Bosma, Mariët Theune
Human Media Interaction
University of Twente
The Netherlands
w.e.bosma@ewi.utwente.nl
m.theune@ewi.utwente.nl

Abstract

In this paper, we investigate the usefulness of normalized alignment of dependency trees for entailment prediction. Overall, our approach yields an accuracy of 60% on the RTE2 test set, which is a significant improvement over the baseline. Results vary substantially across the different subsets, with a peak performance on the summarization data. We conclude that normalized alignment is useful for detecting textual entailments, but a robust approach will probably need to include additional sources of information.

1 Introduction

The well-known fact that similar information can be expressed in many different ways is a challenge for robust NLP applications. It is generally assumed that the performance of such applications could improve when they would have a better understanding of how different expressions relate to each other, for instance in terms of paraphrases (same semantic content, different wording) or entailments (one expression more specific than the other). An automatic summarisation tool, for instance, could use semantic overlap to extract more informative sentences, while a QA system, to give another example, could use it to select answer strings, perhaps preferring more specific answers over more general ones. In a similar vein, Information Extraction (IE) and Information Retrieval (IR) applications might be able to improve recall by taking overlap information into account. In fact, detecting semantic overlap may well be regarded as a generic NLP task on a par with tasks such as word sense disambiguation and named entity recognition.

Recognizing textual entailments (RTE) is a specific instance of detecting semantic overlap, and is currently an active area of research (Dagan et al., 2005). The RTE task is commonly defined as follows: given a text T (usually consisting of one or two sentences, the premise) determine whether a sentence H (the hypothesis) is entailed by T (the premise). Table 1 shows three randomly selected examples from the RTE2 development set.[1]

Various approaches have been proposed to tackle this problem. One approach is to translate both T and H into a logical form, and then use a general purpose theorem prover to check for entailment (Bos and Markert, 2005). To do proper theorem proving, this method requires that formalized background knowledge is taken into account. A different approach is to try to compute the amount of "similarity" between T and H, under the assumption that a higher similarity increases the likelihood of an entailment relation. It is generally assumed that pure string overlap is not sufficient for detecting entailment, and that using some amount of linguistic information may be beneficial (Herrera et al., 2005; Vanderwende et al., 2005). Then the question becomes how to align the linguistic analyses of T and H.

Alignment has been studied extensively in data-

[1] Available from http://www.pascal-network.org/Challenges/RTE2/.

Table 1: Three Text-Hypothesis pairs taken from the IR part of the RTE2 development set. IDs 462 and 759 are examples of textual entailments, while ID 472 is not.

ID	$\models$	Text and Hypothesis Sentences
462	Y	T = The development of agriculture by early humans, roughly 10,000 years ago, was also harmful to many natural ecosystems as they were systematically destroyed and replaced with artificial versions. H = Humans existed 10,000 years ago.
472	N	T = Compuware claims that Allan Tortorice and Jim Hildner were among several former employees who revealed trade secrets after they moved to IBM. H = Trade secrets were stolen.
759	Y	T = Tropical Storm Debby is blamed for several deaths across the Caribbean. H = A tropical storm has caused loss of life.

driven machine translation (Och and Ney, 2000), initially at the word level, but current work increasingly focusses on alignment at higher levels (substrings, syntactic parser or trees) as well, e.g., Meyers et al. (1996). Here, following Herrera et al. (2005) we align sentences at the level of dependency structures. Dependency trees seem particularly useful for the purpose of RTE (and more useful than, for instance, phrase structure), because they reflect the semantic content of a sentence more directly, abstract over word order, and are relatively compact; but see Gildea (2004) for a dissenting view. Recognizing textual entailments then becomes a three step procedure: first, for both T and H a dependency analysis is obtained, then the respective analyses are aligned, after which it is decided whether T entails H or not.

As our starting point we take the alignment algorithm described in Marsi and Krahmer (2005), which itself is based on an alignment algorithm of Meyers et al. (1996) developed specifically for machine translation (MT). In MT, it is assumed that source and target sentence are closely related; that is, both express the same information albeit in a different language. This "relatedness assumption" is not generally valid for potential entailments, where the text usually contains additional information which is not related to the hypothesis (cf. Table 1). In the case of an entailment, all information from the hypothesis must normally have an aligned counterpart in the text (but not vice versa). Besides making the alignment asymmetric, special attention will be given to normalization, both where the depth and width of the dependency trees as where the structure of the trees are concerned. Since we are primarily interested in the possibilities and limitations of normalized alignment techniques for RTE, we abstained from including other sources in the classification and focussed entirely on tuning the alignment algorithm.

2 Method

2.1 Preprocessing

Starting from the text-hypothesis pairs in the RTE XML format, we first preprocess the data in a number of steps. As the text part may consist of more than one sentence, we first perform sentence splitting using Mxterminator (Reynar and Ratnaparkhi, 1997), a maximum entropy-based end of sentence classifier trained on the Penn Treebank data. Next, all sentences are tokenized with the script originally used for tokenizing the Penn Treebank, with some tweaks to correctly tokenize large numbers. Next, we perform part-of-speech tagging and lemmatization. For POS tagging, we use the memory-based tagger (Daelemans et al., 2003) trained on the Penn Treebank data, using the Penn Treebank tagset. For lemmatization, we employ the memory-based lemmatizer (van den Bosch and Daelemans, 1999) trained on the CELEX lexicon for English.

This is followed by the syntactic analysis, for which we relied on the MaltParser system, a data-driven dependency parser which can be used to induce a parsing model from treebank data and to parse new data using the induced model (Nivre and Scholz, 2004). We use MaltParser as trained on the Penn Treebank (which explains why we use Penn Treebank tokenization and POS tags). We employ the arc-eager version of the MaltParser, which delivers projective dependency trees.

Finally, the dependency structures are syntactically normalized to facilitate alignment between T and H. These normalization rules rely on lemmas, POS tags and dependency relations. This step involves the following three syntactic transformations. (1) *Auxiliary reduction*: To simplify the dependency trees, auxiliaries of progressive and perfective tense are removed, and their children are attached to the remaining content verb. The same goes for modal verbs, and for *do* in the do-support function. (2) *Passive to active form*: The passive form auxiliary is removed, the original subject becomes object, and (where possible) a by-phrase becomes the subject. This facilitates alignment between passive and active sentences. (3) *Copula reduction*: Copular verbs are removed by attaching the predicate (i.e., the subtree with dependency relation PRED) as a daughter to the subject (i.e., the subtree with dependency relation SUB). The motivation for this rule is to enhance the alignment of appositive constructions ("The U.S. president, George Bush") with their sentential counterparts ("The U.S. president is George Bush").

The linguistically enriched text-hypothesis pairs are stored in XML format to serve as input for tree alignment.

2.2 Tree alignment algorithm

The tree alignment algorithm of Marsi and Krahmer (2005), which was adapted from Meyers et al. (1996), calculates the match between each node in a dependency tree D against each node in another dependency tree D'. The matching score for each pair of nodes depends not only on the similarity of the nodes, but also recursively on the scores of the best matching pairs of their descendants. For an efficient implementation, dynamic programming is used to build up a score matrix, which guarantees that each score will be calculated only once.

More precisely: given two dependency trees D and D', for T and H respectively, the algorithm builds up a score function $S(v, v')$ for matching each node v in D against each node v' in D', which is stored in a matrix M. The value of $S(v, v')$ is the score for the best match between the two subtrees rooted at v in D and at v' in D'. When a value for $S(v, v')$ is required, and is not yet in the matrix M, it is recursively computed by the following formula:

$$S(v, v') = max \begin{pmatrix} \textsc{TreeMatch}(v, v') \\ max_i\ S(v_i, v') \\ max_j\ S(v, v'_j) - SP \end{pmatrix}$$

where v_i denotes the i-th child of v and v_j denotes the j-th child of v'. The three terms correspond to the three ways that nodes can be aligned:

1. root node v can be directly aligned to root node v' (see below);
2. any of the children of v can be aligned to v';
3. v can be aligned to any of the children of v'.

The last two options imply skipping one or more edges, and leaving one or more nodes unaligned. In the original formulation of the algorithm (Meyers et al., 1996), there is a penalty for skipping edges. We modified the algorithm such that only skipping nodes in the hypothesis' dependency tree is penalized by Skip Penalty ($0 \leq SP \leq 1$), whereas skipping nodes in the text's dependency tree is not.

The function $\textsc{TreeMatch}(v, v')$ is a measure of how well the subtrees rooted at v and v' match:

$$\begin{aligned} &\textsc{TreeMatch}(v, v') = \\ &\quad PW \cdot \textsc{ParentMatch}(v, v') + \\ &\quad (1 - PW) \cdot \textsc{ChildMatch}(v, v') \end{aligned}$$

Here we introduced a weighting factor Parent Weight ($0 \leq PW \leq 1$) which determines the contribution of the match between the parent nodes relative to the contribution of the match between child nodes. The PARENTMATCH function is defined as:

$$\textsc{ParentMatch}(v, v') = \begin{cases} 1 & \text{if } word(v) = word(v') \\ 1 & \text{if } lemma(v) = lemma(v') \\ 1 & \text{if } synonym(v, v') \\ 1 & \text{if } hypernym(v, v') \\ sim(v, v') & \text{if } sim(v, v') > 0.1 \\ 0 & otherwise \end{cases}$$

This basically states that two words are similar if their lowercase word forms or lemmas are identical, but also if the word in the text is a synonymn

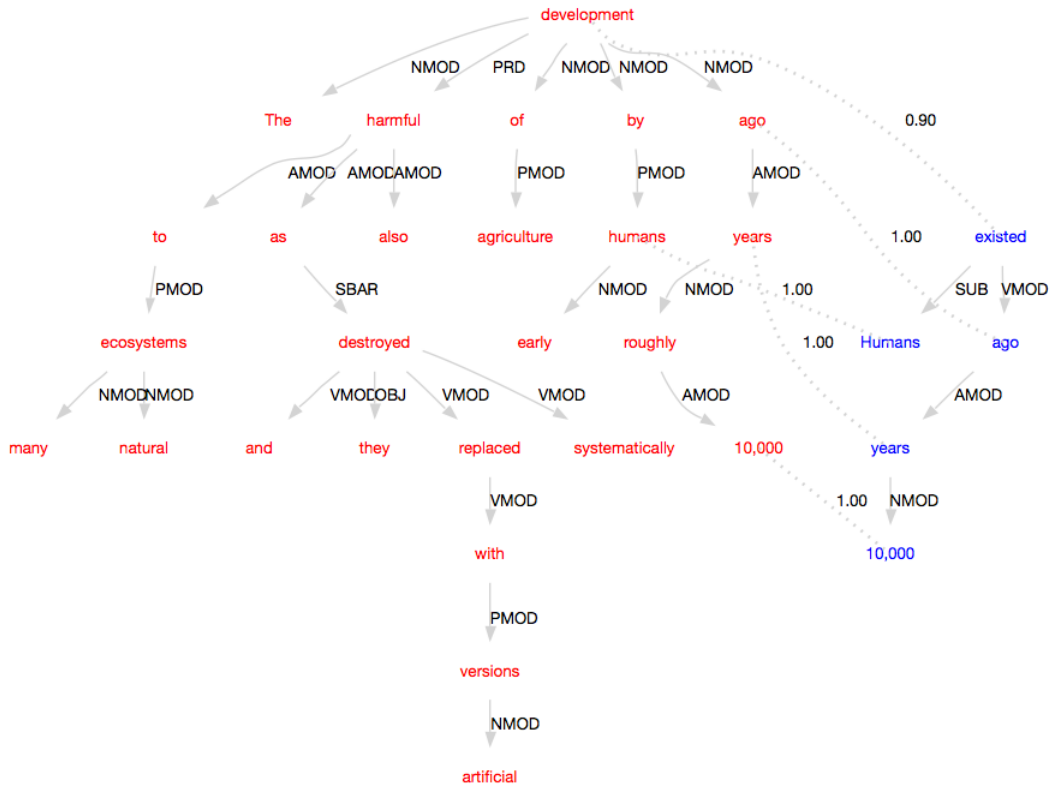

Figure 1: Normalized alignment (dotted lines) of the dependency trees (straight lines) for T (left) and H (right) of ID 462, see Table 1. Since the alignment of the respective root nodes is above the optimal threshold for IR (0.6), this example is classified as an entailment.

of the word in the hypothesis according to WordNet, or when the hypernym closure of the text word contains the hypothesis word. Finally, a pair of words is to some degree similar when the pair is found in Dekang Lin's dependency-based thesaurus with a proximity score higher than 0.1.[2]

The second component of TREEMATCH is the CHILDMATCH function, which represents how well the children of node v and v' can be aligned.

$$\text{CHILDMATCH}(v, v') = \max_{p \in \mathcal{P}(v,v')} \left[\sum_{(i,j) \in p} \frac{|v'_j|}{|v'|} \cdot S(v_i, v'_j) \right]$$

[2]Available from `http://www.cs.ualberta.ca/~lindek/downloads.htm`

Here $\mathcal{P}(v, v')$ is the set of all possible pairings of the n children of v against the m children of v', which amounts to the power set of $\{1, \ldots, n\} \times \{1, \ldots, m\}$. Notice that this implies the match is unordered, so we intentionally abstract from the surface word order. The summation ranges over all pairs, denoted by (i, j), which appear in a given pairing $p \in \mathcal{P}(v, v')$. Maximizing this summation thus amounts to finding the optimal alignment of children of v to children of v'.

The expression $|v'_j|/|v'|$ represent the number of tokens dominated by the j-th child node of node v' in the hypothesis divided by the total number of tokens dominated by node v'. This weighting factor is another extension to the original algorithm. It not

Table 2: Parameter settings for Skip Penalty (SP), Parent Weight (PW) and Treshold (TH) per task.

Task	SP	PW	TH
IE	0.6	0.2	0.6
IR	0.8	0.1	0.6
QA	0.9	0.2	0.6
SUM	0.9	0.1	0.4

only guarantees that the value of CHILDMATCH is normalized (i.e., always between zero and one), but also has the effect of giving a higher weight to complex child nodes. Without this factor, an aligned terminal node would have the same contribution as a complex non-terminal child node in which just one node is aligned.

Finally, there are a number of specific issues which we mention briefly. During the alignment, all nodes representing punctuation are discounted. The skip penalty (SP) and parent weight (PW) parameters have task-specific settings (depicted in Table 2). Somewhat to our surprise, we found that it is not beneficial to take dependency relation labels into account during the matching of child nodes. If the text consists of multiple sentences, we try to align the hypothesis to each of these sentences, and pick the alignment with the highest score. Similarly, if the dependency analysis of a sentence fails and as a result is not fully connected (i.e., consists of multiple trees), we try to align the hypothesis to each of these, and again pick the alignment with the highest score. When looking for synonymns, hypernyms and similar words in the PARENTMATCH function, we also look up phrasal verbs, where the most likely verbal particle is derived from the dependency analysis and POS tag.

2.3 Entailment prediction

In order to predict whether an entailment relation holds between the text and hypothesis, we simply look at whether the top node of the hypothesis dependency tree is aligned, and whether the alignment strength exceeds a certain treshold value. The treshold (TH) is set differently depending on the task, as shown in Table 2. These settings were obtained by manual optimization on the training set.

Table 3: Percent accuracy on RTE2 development and test sets, where Dev_o uses optimized settings and Dev_s uses the submitted settings (same as for the test set).

Task	Dev_o	Dev_s	Test
IE	56.0	53.0	52.0
IR	61.0	58.0	58.5
QA	60.0	57.5	62.5
SUM	72.0	72.0	69.0
Overall	62.25	60.1	60.5

3 Results

Table 3 presents the results on the RTE2 development and test sets for each of the four subtasks. The normalized alignment approach yields an overall 60% accuracy on the test set, with the results on the SUM subset clearly best (with an accuracy of nearly 70%) and those on the IE subset clearly worst (barely above chance level). Overall, the alignment algorithm thus significantly outperforms the "always predict entailment" baseline (50%). Moreover, the scores on the test set differ only marginally from those on the training set (in fact, they are slightly better), which suggests that the approach is not overfitted and that the obtained performance level is fairly robust. Tuning the three parameters for each subset individually is also beneficial, which indicates that the nature of the alignment is different across the various subsets.

4 Discussion

The pattern of results seems comparable to the results reported on the RTE1 test set (Dagan et al., 2005), in the sense that one subset appears to be easier than the others (CD in RTE1 and SUM in RTE2). Presumably, these subsets are easier, because they rely least on the presence of background knowledge. In addition, the intuition that a good alignment between T and H is indicative of entailment seems inherently more plausible for the SUM subset than for the other three.

Even though our approach to alignment clearly performs better than the baseline strategy, there is plenty of room for performance gains, which can be obtained in various ways. First, the dependency parser could be improved. In the current set up it erred occasionally; a small number of texts (91) resulted in broken parses (no unique root node), which obviously makes aligning problematic. Moreover, the analysis of numbers and dates was not always adequate, and various additional syntactic transformations suggest themselves but are not yet implemented.

Still there are clear limits to the applicability of alignment for entailment detection. A manual analysis of the test set revealed that in the vast majority of the cases where T entails H, it was possible to align the top node of H with some node in T . But in many cases, the alignment can only be established on the basis of background knowledge. The problem can be illustrated with example ID 759 shown in Figure 1. Arguably, "several deaths" in T can be aligned with "loss of life" in H, and probably "blamed" can be aligned with "caused" (but this is trickier[3]). It can be argued that for these cases, besides alignment, the classification of semantic relations between phrases might be beneficial as well (Marsi and Krahmer, 2005), and we hope to experiment with this in future work.

We believe that the best and most robust results on the RTE task will be obtained by combining different information sources, see e.g., Bos and Markert (2005) or Raina et al. (2005). The current paper argues that normalized alignment could be one of these information sources.

Acknowledgements

This work was carried out within the IMIX-IMOGEN (Interactive Multimodal Output Generation) project, sponsored by the Netherlands Organization of Scientific Research (NWO), and during the preparatory stages of the Stevin-NWO DAESO (Detecting and Exploiting Semantic Overlap) project.

[3]The underlying assumption would be that if A is blamed for B, then A has caused B. Notice that this is not a logical necessity, while loss of life is a necessary consequence of death.

References

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of EMNLP*.

Walter Daelemans, Jakub Zavrel, Antal van den Bosch, and Ko Van der Sloot. 2003. MBT: Memory based tagger, version 2.0, reference guide. ILK Technical Report 03-13, Tilburg University.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the 1st PASCAL Challenges Workshop on Recognising Textual Entailment*.

Daniel Gildea. 2004. Dependencies vs. constituents for tree-based alignment. In *Proceedings of EMNLP*, Barcelona.

Jesus Herrera, Anselmo Penas, and Felisa Verdejo. 2005. Textual entailment recognition based on dependency analysis and WordNet. In *Proceedings of the 1st PASCAL Challenges Workshop on Recognising Textual Entailment*.

Erwin Marsi and Emiel Krahmer. 2005. Classification of semantic relations by humans and machines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 1–6, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Adam Meyers, Roman Yangarber, and Ralph Grisham. 1996. Alignment of shared forests for bilingual corpora. In *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, pages 460–465, Copenhagen, Denmark.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING 2004*, pages 23–27, Geneva, Switzerland.

Franz Josef Och and Hermann Ney. 2000. Statistical machine translation. In *EAMT Workshop*, pages 39–46, Ljubljana, Slovenia.

Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova, Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, and Andrew Y. Ng. 2005. Robust textual inference using diverse knowledge sources. In *Proceedings of the 1st PASCAL Challenges Workshop on Recognising Textual Entailment*.

Jeffrey Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C.

Antal van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 285–292, San Francisco, CA. Morgan Kaufmann.

Lucy Vanderwende, Deborah Coughlin, and William Dolan. 2005. What syntax can contribute in entailment task. In *Proceedings of the 1st PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, U.K.