Automatic analysis of semantic similarity in comparable text through syntactic tree matching

Erwin Marsi
TiCC, Tilburg University
e.c.marsi@uvt.nl

Emiel Krahmer
TiCC, Tilburg University
e.j .krahmer@uvt.nl

Abstract

We propose to analyse semantic similarity in comparable text by matching syntactic trees and labeling the alignments according to one of five semantic similarity relations. We present a Memory-based Graph Matcher (MBGM) that performs both tasks simultaneously as a combination of exhaustive pairwise classification using a memory-based learner, followed by global optimization of the alignments using a combinatorial optimization algorithm. The method is evaluated on a monolingual treebank consisting of comparable Dutch news texts. Results show that it performs substantially above the baseline and close to the human reference.

1 Introduction

Natural languages allow us to express essentially the same underlying meaning as many alternative surface forms. In other words, there are often many similar ways to say the same thing. This characteristic poses a problem for many natural language processing applications. Automatic summarizers, for example, typically rank sentences according to their informativity and then extract the top n sentences, depending on the required compression rate. Although the sentences are essentially treated as independent of each other, they typically are not. Extracted sentences may have substantial semantic overlap, resulting in unintended redundancy in the summaries. This is particularly problematic in the case of multi-document summarization, where sentences extracted from related documents are very likely to express similar information in different ways (Radev and McKeown, 1998). Therefore, if semantic similarity between sentences could be detected automatically, this would certainly help to avoid redundancy in summaries.

Similar arguments can be made for many other NLP applications. Automatic duplicate and plagiarism detection beyond obvious string overlap requires recognition of semantic similarity. Automatic question-answering systems may benefit from clustering semantically similar candidate answers. Intelligent document merging software, which supports a minimal but lossless merge of several revisions of the same text, must handle cases of paraphrasing, restructuring, compression, etc. Recognizing textual entailments (Dagan et al., 2005) could arguably be seen as a specific instance of detecting semantic similarity.

In addition to merely *detecting* semantic similarity, we can ask to what extent two expressions share meaning. For instance, the meaning of one sentence can be fully contained in that of another, the meaning of one sentence can overlap only partly with that of another, etc. This requires an *analysis* of the semantic similarity between a pair of expressions. Like detection, automatic analysis of semantic similarity can play an important role in NLP applications. To return to the case of multi-document summarization, analysing the semantic similarity between sentences extracted from different documents provides the basis for *sentence fusion*, a process where a new sentence is generated that conveys all common information from both sentences without introducing redundancy (Barzilay and McKeown, 2005; Marsi and Krahmer, 2005b).

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 752–760,
Beijing, August 2010

Analysis of semantic similarity can be approached from different angles. A basic approach is to use string similarity measures such as the Levenshtein distance or the Jaccard similarity coefficient. Although cheap and fast, this fails to account for less obvious cases such as synonyms or syntactic paraphrasing. At the other extreme, we can perform a deep semantic analysis of two expressions and rely on formal reasoning to derive a logical relation between them. This approach suffers from issues with coverage and robustness commonly associated with deep linguistic processing. We therefore think that the middle ground between these two extremes offers the best option. In this paper we present a new method for analysing semantic similarity in comparable text. It relies on a combination of morphological and syntactic analysis, lexical resources such as word nets, and machine learning from examples. We propose to analyse semantic similarity between sentences by aligning their syntax trees, where each node is matched to the most similar node in the other tree (if any). In addition, we label these alignments according to the type of similarity relation that holds between the aligned phrases. The labeling supports further processing. For instance, Marsi & Krahmer (2005b; 2008) describe how to generate different types of sentence fusions on the basis of this relation labeling.

In the next Section we provide a more formal definition of the task of matching syntactic trees and labeling alignments, followed by a discusion of related work in Section 3. Section 4 describes a parallel, monolingual treebank used for developing and testing our approach. In Section 5 we propose a new algorithm for simultaneous node alignment and relation labeling. The results of several evaluation experiments are presented in Section 6. We finish with a conclusion.

2 Problem statement

Aligning a pair of similar syntactic trees is the process of pairing those nodes that are most similar. More formally: let v be a node in the syntactic tree T of sentence S and v' a node in the syntactic tree T' of sentence S'. A *labeled node alignment* is a tuple $< v, v', r >$ where r is a label from a set of relations. A *labeled tree alignment* is a set of labeled node alignments. A *labeled tree matching* is a tree alignment in which each node is aligned to at most one other node.

For each node v, its terminal *yield* $\text{STR}(v)$ is defined as the sequence of all terminal nodes reachable from v (i.e., a substring of sentence S). Aligning node v to v' with label r indicates that relation r holds between their yields $\text{STR}(v)$ and $\text{STR}(v')$. We label alignments according to a small set of *semantic similarity relations*. As an example, consider the following Dutch sentences:

(1) a. Dagelijks koffie vermindert risico op
Daily coffee diminishes risk on
Alzheimer en Dementie.
Alzheimer and Dementia.

b. Drie koppen koffie per dag reduceert
Three cups coffee a day reduces
kans op Parkinson en Dementie.
chance on Parkinson and Dementia.

The corresponding syntax trees and their (partial) alignment is shown in Figure 1. We distinguish the following five mutually exclusive similarity relations:

1. v **equals** v' iff lower-cased $\text{STR}(v)$ and lower-cased $\text{STR}(v')$ are identical – example: *Dementia* equals *Dementia*;
2. v **restates** v' iff $\text{STR}(v)$ is a proper paraphrase of $\text{STR}(v')$ – example: *diminishes* restates *reduces*;
3. v **generalizes** v' iff $\text{STR}(v)$ is more general than $\text{STR}(v')$ – example: *daily coffee* generalizes *three cups of coffee a day*;
4. v **specifies** v' iff $\text{STR}(v)$ is more specific than $\text{STR}(v')$ – example: *three cups of coffee a day* specifies *dailly coffee*;
5. v **intersects** v' iff $\text{STR}(v)$ and $\text{STR}(v')$ share meaning, but each also contains unique information not expressed in the other – example: *Alzheimer and Dementia* intersects *Parkinson and Dementia*.

Our interpretation of these relations is one of common sense rather than strict logic, akin to the definition of entailment employed in the RTE challenge (Dagan et al., 2005). Note also that relations are prioritized: *equals* takes precedence

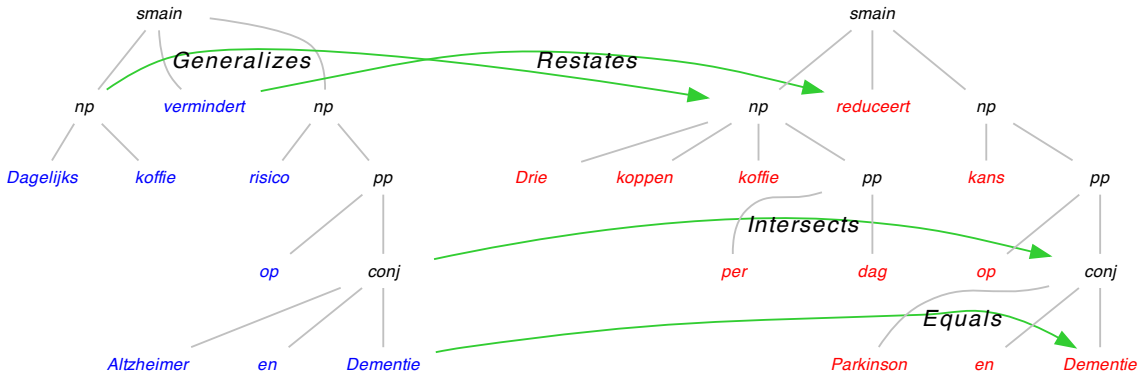

Figure 1: Example of two aligned and labeled syntactic trees. For expository reasons the alignment is not exhaustive.

over *restates*, etc. Furthermore, *equals*, *restates* and *intersects* are symmetrical, whereas *generalizes* is the inverse of *specifies*. Finally, nodes containing unique information, such as *Alzheimer* and *Parkinson*, remain unaligned.

3 Related work

Many syntax-based approaches to machine translation rely on bilingual treebanks to extract transfer rules or train statistical translation models. In order to build bilingual treebanks a number of methods for automatic tree alignment have been developed, e.g., (Gildea, 2003; Groves et al., 2004; Tinsley et al., 2007; Lavie et al., 2008). Most related to our approach is the work on discriminative tree alignment by Tiedemann & Kotzé (2009). However, these algorithms assume that source and target sentences express the same information (i.e. *parallel* text) and cannot cope with comparable text where parts may remain unaligned. See (MacCartney et al., 2008) for further arguments and empirical evidence that MT alignment algorithms are not suitable for aligning parallel monolingual text.

MacCartney, Galley, and Manning (2008) describe a system for monolingual phrase alignment based on supervised learning which also exploits external resources for knowledge of semantic relatedness. In contrast to our work, they do not use syntactic trees or similarity relation labels. Partly similar semantic relations are used in (MacCartney and Manning, 2008) for modeling semantic containment and exclusion in natural language inference. Marsi & Krahmer (2005a) is closely related to our work, but follows a more complicated method: first a dynamic programming-based tree alignment algorithm is applied, followed by a classification of similarity relations using a supervised-classifier. Other differences are that their data set is much smaller and consists of parallel rather than comparable text. A major drawback of this algorithmic approach it that it cannot cope with crossing alignments. We are not aware of other work that combines alignment with semantic relation labeling, or algorithms which perform both tasks simultaneously.

4 Data collection

For developing our alignment algorithm we use the DAESO corpus[1]. This is a Dutch parallel monolingual treebank of 1 million words, half of which were manually annotated. The corpus consists of pairs of sentences with different levels of semantic overlap, ranging from high (different Dutch translations of books from Darwin, Montaigne and Saint-Exupéry) to low (different press releases from the two main news agencies in The Netherlands, ANP and NOVUM). For this paper, we concentrate on the latter part of the DAESO corpus, where the proportion of Equals and Restates is relatively low. This corpus segment consists of 8,248 pairs of sentences, containing 162,361 tokens (ignoring punctuation). All sentences were tokenized and tagged, and subsequently parsed by the Alpino dependency parser for Dutch (Bouma et al., 2001). Two annota-

[1] http://daeso.uvt.nl

		Alignment:	Labeling:						
			Eq:	Re:	Spec:	Gen:	Int:	Macro:	Micro:
Words:	F:	95.38	95.48	58.50	65.81	65.00	25.85	62.11	88.72
	SD:	2.16	2.69	7.63	13.05	11.25	18.74		
Full trees:	F:	88.31	95.83	71.38	60.21	66.71	62.67	71.36	81.92
	SD:	1.15	2.27	3.77	7.63	8.17	6.14		

Table 1: Average F-scores (in percentages, with Standard Deviations) for the six human annotators on alignment and semantic relation labeling, for words and for full syntactic trees.

tors determined which sentences in the comparable news reports contained semantic overlap. Six other annotators produced manual alignments of words and phrases in matched sentence pairs, which resulted in 86,227 aligned pairs of nodes.

A small sample of 10 similar press releases comprising a total of 48 sentence pairs was independently annotated by all six annotators to determine inter-annotator agreement. We used precision, recall and F-score on alignment. To calculate these scores for relation labeling, we simply restrict the set of alignments to those labeled with a particular relation, ignoring all others. Likewise, we restrict these sets to terminal node alignments in order to get scores on word alignment.

Given the six annotations $A_1, \ldots, A_6$, we repeatedly took one as the $True$ annotation against which the five other annotations were evaluated. We then computed the average scores over these $6 * 5 = 30$ scores (note that with this procedure, precision, recall and F score end up being equal). Table 1 summarizes the results, both for word alignments and for full syntactic tree alignment. It can be seen that for alignment of words an average F-score of over 95 % was obtained, while alignment for full syntactic trees results in an F-score of 88%. For relation labeling, the scores differed per relation, as is to be expected: the average F-score for Equals was over 95% for both word and full tree alignment[2], and for the other relations average F-scores between 0.6 and 0.7 were obtained. The exception to note is Intersects on word level, which only occurred a few times according to a few of the annotators. The macro and micro (weighted) F-score averages on labeled alignment are 62.11% and 88.72% for words, and 71.36% and 81.92% for full syntactic trees.

[2]At first sight, it may seem that labeling Equals is a trivial and deterministic task, for which the F-score should always be close to 100%. However, the same word may occur multiple times in the source or target sentences, which introduces ambiguity. This frequently occurs with function words such as determiners and prepositions. Moreover, choosing among several equivalent Equals alignments may sometimes involve a somewhat arbitrary decision. This situation arises, for instance, when a proper noun is mentioned just once in the source sentence but twice in the target sentence.

5 Memory-based Graph Matcher

In order to automatically perform the alignment and labeling tasks described in Section 2, we cast these tasks simultaneously as a combination of exhaustive pairwise classification using a supervised machine learning algorithm, followed by global optimization of the alignments using a combinatorial optimization algorithm. Input to the tree matching algorithm is a pair of syntactic trees consisting of a source tree T_s and a target tree T_t.

Step 1: Feature extraction For each possible pairing of a source node n_s in tree T_s and a target node n_t in tree T_t, create an instance consisting of feature values extracted from the input trees. Features can represent properties of individual nodes, e.g. the category of the source node is NP, or relations between nodes, e.g. source and target node share the same part-of-speech.

Step 2: Classification A generic supervised classifier is used to predict a class label for each instance. The class is either one of the semantic similarity relations or the special class *none*, which is interpreted as *no alignment*. Our implementation employs the memory-based learner TiMBL (Daelemans et al., 2009), a freely available, efficient and enhanced implementation of k-nearest neighbour classification. The classifier is trained on instances derived according to Step 1 from a parallel treebank of aligned and labeled syntactic trees.

Step 3: Weighting Associate a cost with each prediction so that high costs indicate low confidence in the predicted class and vice versa. We use the normalized entropy of the class labels in the set of nearest neighbours (H) defined as

$$H = -\frac{\sum_{c \in C} p(c)\ log_2\, p(c)}{log_2 |C|} \quad (1)$$

where C is the set of class labels encountered in the set of nearest neighbours (i.e., a subset of the five relations plus *none*), and $p(c)$ is the probability of class c, which is simply the proportion of instances with class label c in the set of nearest neighbours. Intuitively this means that the cost is zero if all nearest neighbours are of the same class, whereas the cost goes to 1 if the nearest neighbours are equally distributed over all possible classes.

Step 4: Matching The classification step will usually give rise to one-to-many alignment of nodes. In order to reduce this to just one-to-one alignments, we search for a node matching which minimizes the sum of costs over all alignments. This is a well-known problem in combinatorial optimization known as the *Assignment Problem*. The equivalent in graph-theoretical terms is a *minimum weighted bipartite graph matching*. This problem can be solved in polynomial time ($O(n^3)$) using e.g., the *Hungarian algorithm* (Kuhn, 1955). The output of the algorithm is the labeled tree matching obtained by removing all node alignments labeled with the special *none* relation.

6 Experiments

6.1 Experimental setup

Word alignment and full tree alignments are conceptually different tasks, which require partly different features and may have different practical applications. These are therefore addressed in separate experiments.

Table 2 summarizes the respective sizes of development and the held-out test set in terms of number of aligned graph pairs, number of aligned node pairs and number of tokens. The percentage of aligned nodes over all graphs is calculated relative to the number of nodes over all graphs. Since alignments involving non-terminal nodes are ignored in the task of word alignment, the number of aligned node pairs and the percentage of aligned nodes is lower in the word develop and word test sets. Table 3 gives the distribution of semantic relations in the development and test set, for word and tree alignment. It can be observed that the distribution if fairly skewed with Equals being the majority class, even more so for word alignments. Another thing to notice is that Intersects are much more frequent at the level of non-terminal alignments.

Data	Graph pairs	Node pairs	Tokens	Aligned nodes (%)
word develop	2 664	13 027	45 149	15.71
word test	547	2 858	10 005	14.96
tree develop	2 664	22 741	45 149	47.20
tree test	547	4 894	10 005	47.05

Table 2: Properties of develop and test data sets

Data	Eq	Re	Spec	Gen	Int
word develop	84.92	6.15	2.10	1.77	5.07
word test	85.62	6.09	2.17	1.99	4.13
tree develop	56.61	6.57	7.52	6.38	22.91
tree test	58.40	7.11	7.40	6.38	20.72

Table 3: Distribution of semantic similarity relations for word alignment and for full tree alignments in both develop and test data sets

Development was carried out using 10-fold cross validation on the development data and consequently reported scores on the development data are averages over 10 folds. Only two parameters were coarsely optimized on the development set. First, the amount of downsampling of the *none* class varied between 0.1 or 0.5. Second, the parameter k of the memory-based classifier – the number of nearest neighbours taken into account during classification – ranged from 1 to 15. Optimal settings were finally applied when testing on the held-out data.

A simple greedy alignment procedure served as baseline. For word alignment, identical words are aligned as Equals and identical roots as Restates. For full tree alignment, this is extended to the level of phrases so that phrases with identical words are aligned as Equals and phrases with identical roots as Restates. The baseline does not predict Spec-

ifies, Generalizes or Intersects relations, as that would require a more involved, knowledge-based approach.

All features used are described in Table 4. The word-based features rely on pure string processing and require no linguistic preprocessing. The morphology-based features exploit the limited amount of morphological analysis provided by the Alpino parser (Bouma et al., 2001). For instance, it provides word roots and decomposes compound words. Likewise the part-of-speech-based features use the coarse-grained part-of-speech tags assigned by the Alpino parser. The lexical-semantic features rely on the Cornetto database (Vossen et al., 2008), a recent extension to the Dutch WordNet, to look-up synonym and hypernym relations among source and target lemmas. Unfortunately there is no word sense disambiguation module to identify the correct senses. In addition, a background corpus of over 500M words of (mainly) news text provides the word counts required to calculate the Lin similarity measure (Lin, 1998). The syntax-based features use the syntactic structure, which is a mix of phrase-based and dependency-based analysis. The phrasal features express similarity between the terminal yields of source and target nodes. With the exception of *same-parent-lc-phrase*, these features are only used for full tree alignment, not for word alignment.

6.2 Results on word alignment

We evaluate our alignment model in two steps: first focussing on word alignment and then on full tree alignment. Table 5 summarizes the results for MBGM on word alignment (50% downsampling and $k = 3$), which we compare statistically to the baseline performance, and informally with the human scores reported in Table 1 in Section 4 (note that the human scores are only for a subset of the data used for automatic evaluation).

The first thing to observe is that the MBGM scores on the development and tests sets are very similar throughout. For predicting word alignments, the MBGM system performs significantly better than the baseline system ($t(18) = 17.72, p < .0001$). On the test set, MBGM obtains an F-score of nearly 89%, which is almost exactly halfway between the scores of the baseline system and the human scores. In a similar vein, the performance of the MBGM system on relation labeling is considerably better than that of the baseline system. For all semantic relations, MBGM performs significantly better than the baseline ($t(18) > 9.4138, p < .0001$ for each relation, trivially so for the Specifies, Generalizes and Intersects relations, which the baseline system never predicts).

The macro scores are plain averages over the 5 scores on each relation, whereas the micro scores are weighted averages. As the Equals is the majority class and at the same time easiest to predict, the micro scores are higher. The macro scores, however, better reflect performance on the real challenge, that is, correctly predicting the relations other than Equals. The MBGM macro average is 27.37% higher than the baseline (but still some 10% below the human top line), while the micro average is 5.83% higher and only 0.75% below the human top line. Macro scores on the test set are overall lower than those on the develop set, presumably because of tuning on the development data.

6.3 Results on tree alignment

Table 6 contains the results of full tree alignment (50% downsampling and $k = 5$); here both terminal and non-terminal nodes are aligned and classified in one pass. Again scores on the development and test set are very similar, the latter being slightly better. For full tree alignment, MBGM once again performs significantly better than the baseline, $t(18) = 25.68, p < .0001$. With an F-score on the test set of 86.65, MBGM scores almost 20 percent higher than the baseline system. This F-score is less than 2% lower than the average F-score obtained by our human annotators on full tree alignment, albeit not on exactly the same sample. The picture that emerges for semantic relation labeling is closely related to the one we saw for word alignments. MBGM significantly outperforms the baseline, for each semantic relation ($t(18) > 12.6636, p < .0001$). MBGM scores a macro average F-score of 52.24% (an increase of 30.05% over the baseline) and a micro average of 80.03% (12.68% above the base score). It is inter-

Feature	Type	Description
		Word
word-subsumption	string	indicate if source word equals, has as prefix, is a prefix of, has a suffix, is a suffix of, has as infix or is an infix of target word
shared-pre-/in-/suffix-len	int	length of shared prefix/infix/suffix in characters
source/target-stop-word	bool	test if source/target word is in a stop word list of frequent function words
source/target-word-len	int	length of source/target word in characters
word-len-diff	int	word length difference in characters
source/target-word-uniq	bool	test if source/target word is unique in source/target sentence
same-words-lhs/rhs	int	no. of identical preceding/following words in source and target word contexts
		Morphology
root-subsumption	string	indicate if source root equals, has as prefix, is a prefix of, has a suffix, is a suffix of, has as infix or is an infix of target root
roots-share-pre-/in-/suffix	bool	source and target root share a prefix/infix/suffix
		Part-of-speech
source/target-pos	string	source/target part-of-speech
same-pos	bool	test if source and target have same part-of-speech
source/target-content-word	bool	test if source/target word is a content word
both-content-word	bool	test if both source and target word are content words
		Lexical-semantic using Cornetto
cornet-restates	float	1.0 if source and target words are synonyms and 0.5 if they are near-synonyms, zero otherwise
cornet-specifies	float	Lin similarity score if source word is a hyponym of target word, zero otherwise
cornet-generalizes	float	Lin similarity score if source word is a hypernym of target word, zero otherwise
cornet-intersects	float	Lin similarity score if source word share a common hypernym, zero otherwise
		Syntax
source/target-cat	string	source/target syntactic category
same-cat	bool	test if source and target have same syntactic category
source/target-parent-cat	string	source/target syntactic category of parent node
source/target-deprel	string	source/target dependency relation
same-deprel	bool	test if source and target have same dependency relation
same-dephead-root	bool	test if the dependency heads of the source and target have same root
		Phrasal
word-prec/rec	float	precision/recall on the yields of source and target nodes
same-lc-phrase	bool	test if lower-cased yields of source and target nodes are identical
same-parent-lc-phrase	bool	test if lower-cased yields of parents of source and target nodes are identical
source/target-phrase-len	int	length of source/target phrase in words
phrase-len-diff	int	phrase length difference in words

Table 4: Features (where slashes indicate multiple versions of the same feature, e.g. *source/target-pos* represents the two features *source-pos* and *target-pos*)

esting to observe that MBGM obtains *higher* F-scores on Equals and on Intersects (the two most frequent relations) than the human annotators obtained. As a result of this, the micro F-score of the automatic full tree alignment is less than 2% lower than the human reference score.

Tree alignment can also be implemented as a two-step procedure, where in the first step alignments and semantic relation classifications at the word level are produced, while in the second step these are used to predict alignments and semantic relations for non-terminals. We experimented with such a two-step procedure as well, in one version using the actual word alignments and in the other the predicted word alignments. The scores of the two-step prediction are only marginally different from those of one step prediction, both for alignment and for relation classification, giving improvements in the order of about 1% for both subtasks. As is to be expected, the scores with true word alignments are much better than those with predicted word alignments. They are interesting though, because they suggest that a fairly good full tree alignment can be automatically ob-

		Alignment:	Labeling:						
			Eq:	Re:	Spec:	Gen:	Int:	Macro:	Micro:
Develop baseline:	Prec:	80.59	81.84	46.26	0.00	0.00	0.00	25.61	80.22
	Rec:	81.58	93.10	34.71	0.00	0.00	0.00	25.56	82.20
	F:	81.08	87.11	39.66	0.00	0.00	0.00	25.35	80.70
Develop MBGM:	Prec:	91.72	94.54	61.26	74.60	67.82	45.80	68.80	90.82
	Rec:	87.82	95.91	46.19	40.87	43.22	27.27	50.61	86.96
	F:	89.73	95.02	52.67	52.81	52.80	34.19	57.50	88.85
Test baseline:	Prec:	82.45	83.83	43.12	0.00	0.00	0.00	25.39	82.17
	Rec:	82.19	93.87	27.01	0.00	0.00	0.00	24.18	82.02
	F:	82.32	88.57	33.22	0.00	0.00	0.00	24.36	82.14
Test MBGM:	Prec:	90.92	94.20	53.33	59.87	54.21	42.47	60.84	89.90
	Rec:	87.09	95.41	40.21	32.75	43.28	20.31	46.39	86.11
	F:	88.96	94.80	45.85	42.34	48.17	27.48	51.73	87.97

Table 5: Scores (in percentages) on word alignment and semantic relation labeling

		Alignment:	Labeling:						
			Eq:	Re:	Spec:	Gen:	Int:	Macro:	Micro:
Develop baseline:	Prec:	82.50	83.76	46.72	0.00	0.00	0.00	26.10	82.18
	Rec:	54.54	93.66	20.01	0.00	0.00	0.00	22.74	54.34
	F:	65.67	88.43	28.02	0.00	0.00	0.00	23.29	65.42
Develop MBGM:	Prec:	92.23	96.15	55.90	54.40	56.15	70.33	66.59	84.99
	Rec:	81.04	94.03	26.64	21.71	29.34	70.27	48.40	74.68
	F:	86.27	95.08	36.08	31.03	38.54	70.30	54.21	79.50
Test baseline:	Prec:	84.23	85.68	42.24	0.00	0.00	0.00	25.58	84.14
	Rec:	56.21	94.44	14.08	0.00	0.00	0.00	21.70	56.15
	F:	67.43	89.85	21.12	0.00	0.00	0.00	22.19	67.35
Test MBGM:	Prec:	92.27	96.67	60.25	46.92	56.85	68.64	65.87	85.23
	Rec:	81.67	94.54	27.87	19.55	30.94	71.01	48.87	75.44
	F:	86.65	95.60	38.11	27.60	40.07	69.80	54.24	80.03

Table 6: Scores (in percentages) on full tree alignment and semantic relation labeling

tained given a manually checked word alignment.

7 Conclusions

We have proposed to analyse semantic similarity between comparable sentences by aligning their syntax trees, matching each node to the most similar node in the other tree (if any). In addition, alignments are labeled with a semantic similarity relation. We have presented a Memory-based Graph Matcher (MBGM) that performs both tasks simultaneously as a combination of exhaustive pairwise classification using a memory-based learning algorithm, and global optimization of alignments using a combinatorial optimization algorithm. It relies on a combination of morphological/syntactic analysis, lexical resources such as word nets, and machine learning using a parallel monolingual treebank. Results on aligning comparable news texts from a monolingual parallel treebank for Dutch show that MBGM consistently and significantly outperforms the baseline, both for alignment and labeling. This holds both for word alignment and tree alignment.

In future research we will test MBGM on other data, as the DAESO corpus contains sub-corpora with various degrees of semantic overlap. In addition, we intend to explore alternative features from word space models. Finally, we plan to evaluate MBGM in the context of NLP applications such as multi-document summarization. This includes work on how to define similarity at the sentence level in terms of the proportion of aligned constituents. Both MBGM and the annotated data set will be publicly released.[2]

Acknowledgments

This work was conducted within the DAESO project funded by the Stevin program (De Nederlandse Taalunie).

References

Barzilay, Regina and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Bouma, Gosse, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In Daelemans, Walter, Khalil Sima'an, Jorn Veenstra, and Jakub Zavre, editors, *Computational Linguistics in the Netherlands 2000.*, pages 45–59. Rodopi, Amsterdam, New York.

Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2009. TiMBL: Tilburg Memory Based Learner, version 6.2, reference manual. Technical Report ILK 09-01, Induction of Linguistic Knowledge, Tilburg University.

Dagan, I., O. Glickman, and B. Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, U.K.

Gildea, Daniel. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 80–87, Sapporo, Japan.

Groves, D., M. Hearne, and A. Way. 2004. Robust subsentential alignment of phrase-structure trees. In *Proceedings of the 20th International Conference on Computational Linguistics (CoLing '04)*, pages 1072–1078.

Krahmer, Emiel, Erwin Marsi, and Paul van Pelt. 2008. Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In Moore, J., S. Teufel, J. Allan, and S. Furui, editors, *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 193–196, Columbus, Ohio, USA.

Kuhn, Harold W. 1955. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Lavie, A., A. Parlikar, and V. Ambati. 2008. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, pages 87–95.

Lin, D. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304.

MacCartney, B. and C.D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528.

MacCartney, Bill, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii, October.

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

Marsi, Erwin and Emiel Krahmer. 2005b. Explorations in sentence fusion. In *Proceedings of the 10th European Workshop on Natural Language Generation*, Aberdeen, GB.

Radev, D.R. and K.R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

Tiedemann, J. and G. Kotzé. 2009. Building a Large Machine-Aligned Parallel Treebank. In *Eighth International Workshop on Treebanks and Linguistic Theories*, page 197.

Tinsley, J., V. Zhechev, M. Hearne, and A. Way. 2007. Robust language-pair independent sub-tree alignment. *Machine Translation Summit XI*, pages 467–474.

Vossen, P., I. Maks, R. Segers, and H. van der Vliet. 2008. Integrating lexical units, synsets and ontology in the Cornetto Database. In *Proceedings of LREC 2008*, Marrakech, Morocco.