Classification of semantic relations by humans and machines *

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

Abstract

This paper addresses the classification of semantic relations between pairs of sentences extracted from a Dutch parallel corpus at the word, phrase and sentence level. We first investigate the performance of human annotators on the task of manually aligning dependency analyses of the respective sentences and of assigning one of five semantic relations to the aligned phrases (equals, generalizes, specifies, restates and intersects). Results indicate that humans can perform this task well, with an F-score of $.98$ on alignment and an F-score of $.95$ on semantic relations (after correction). We then describe and evaluate a combined alignment and classification algorithm, which achieves an F-score on alignment of .85 (using EuroWordNet) and an F-score of .80 on semantic relation classification.

1 Introduction

An automatic method that can determine how two sentences relate to each other in terms of **semantic overlap** or **textual entailment** (e.g., (Dagan and Glickman, 2004)) would be a very useful thing to have for robust natural language applications. A summarizer, for instance, could use it to extract the most informative sentences, while a question-answering system – to give a second example – could use it to select potential answer string (Punyakanok et al., 2004), perhaps preferring more specific answers over more general ones. In general, it is very useful to know whether some sentence S is more specific (entails) or more general than (is entailed by) an alternative sentence S', or whether the two sentences express essentially the same information albeit in a different way (paraphrasing).

Research on automatic methods for recognizing semantic relations between sentences is still relatively new, and many basic issues need to be resolved. In this paper we address two such related issues: (1) to what extent can human annotators label semantic overlap relations between words, phrases and sentences, and (2) what is the added value of linguistically informed analyses.

It is generally assumed that pure string overlap is not sufficient for recognizing semantic relations; and that using some form of syntactic analysis may be beneficial (e.g., (Herrera et al., 2005), (Vanderwende et al., 2005)). Our working hypothesis is that semantic overlap at the word and phrase levels may provide a good basis for deciding the semantic relation between sentences. Recognising semantic relations between sentences then becomes a two-step procedure: first, the words and phrases in the respective sentences need to be aligned, after which the relations between the pairs of aligned words and phrases should be labeled in terms of semantic relations.

Various alignment algorithms have been developed for data-driven approaches to machine translation (e.g. (Och and Ney, 2000)). Initially work focused on word-based alignment, but more and more work is also addressing alignment at the higher levels (substrings, syntactic phrases or trees), e.g., (Meyers et al., 1996), (Gildea, 2003). For our purposes, an additional advantage of aligning syntactic structures is that it keeps the alignment feasible (as the number of arbitrary substrings that may be aligned grows exponentially to the number of words

* This work was carried out within the IMIX-IMOGEN (Interactive Multimodal Output Generation) project, sponsored by the Netherlands Organization of Scientific Research (NWO).

Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pages 1–6,
Ann Arbor, June 2005.

in the sentence). Here, following (Herrera et al., 2005) and (Barzilay, 2003), we will align sentences at the level of **dependency structures**. In addition, we will label the alignments in terms of five basic semantic relations to be defined below. We will perform this task both manually and automatically, so that we can address both of the issues raised above.

Section 2 describes a monolingual parallel corpus consisting of two Dutch translations, and formalizes the alignment-classification task to be performed. In section 3 we report the results on alignment, first describing interannotator agreement on this task and then the results on automatic alignment. In section 4, then, we address the semantic relation classification; again, first describing interannotator results, followed by results obtained using memory-based machine learning techniques. We end with a general discussion.

2 Corpus and Task definition

2.1 Corpus

We have developed a **parallel monolingual corpus** consisting of two different Dutch translations of the French book "Le petit prince" (*the little prince*) by Antoine de Saint-Exupéry (published 1943), one by Laetitia de Beaufort-van Hamel (1966) and one by Ernst Altena (2000). For our purposes, this proved to be a good way to quickly find a large enough set of related sentence pairs, which differ semantically in interesting and subtle ways. In this work, we used the first five chapters, with 290 sentences and 3600 words in the first translation, and 277 sentences and 3358 words in the second translation. The texts were automatically tokenized and split into sentences, after which errors were manually corrected. Corresponding sentences from both translations were manually aligned; in most cases this was a one-to-one mapping, but occasionally a single sentence in one translation mapped onto two or more sentences in the other: this occurred 23 times in all five chapters. Next, the **Alpino** parser for Dutch (e.g., (Bouma et al., 2001)) was used for part-of-speech tagging and lemmatizing all words, and for assigning a dependency analysis to all sentences. The POS labels indicate the major word class (e.g. *verb*, *noun*, *adj*, and *adv*). The dependency relations hold between tokens and are identical to those used in the Spoken Dutch Corpus. These include dependencies such as *head/subject*, *head/modifier* and *coordination/conjunction*. If a full parse could not be obtained, Alpino produced partial analyses collected under a single root node. Errors in lemmatization, POS tagging, and syntactic dependency parsing were not subject to manual correction.

2.2 Task definition

The task to be performed can be described informally as follows: given two dependency analyses, align those nodes that are semantically related. More precisely: For each node v in the dependency structure for a sentence S, we define $\textsc{str}(v)$ as the substring of all tokens under v (i.e., the composition of the tokens of all nodes reachable from v). An alignment between sentences S and S' pairs nodes from the dependency graphs for both sentences. Aligning node v from the dependency graph D of sentence S with node v' from the graph D' of S' indicates that there is a semantic relation between $\textsc{str}(v)$ and $\textsc{str}(v')$, that is, between the respective substrings associated with v and v'. We distinguish five potential, mutually exclusive, relations between nodes (with illustrative examples):

1. v **equals** v' iff $\textsc{str}(v)$ and $\textsc{str}(v')$ are literally identical (abstracting from case). Example: "a small and a large boa-constrictor" equals "a large and a small boa-constrictor";
2. v **restates** v' iff $\textsc{str}(v)$ is a paraphrase of $\textsc{str}(v')$ (same information content but different wording). Example: "a drawing of a boa-constrictor snake" restates "a drawing of a boa-constrictor";
3. v **specifies** v' iff $\textsc{str}(v)$ is more specific than $\textsc{str}(v')$. Example: "the planet B 612" specifies "the planet";
4. v **generalizes** v' iff $\textsc{str}(v')$ is more specific than $\textsc{str}(v)$. Example: "the planet" generalizes "the planet B 612";
5. v **intersects** v' iff $\textsc{str}(v)$ and $\textsc{str}(v')$ share some informational content, but also each express some piece of information not expressed in the other. Example: "Jupiter and Mars" intersects "Mars and Venus"

Figure 1 shows an example alignment with semantic relations between the dependency structures of

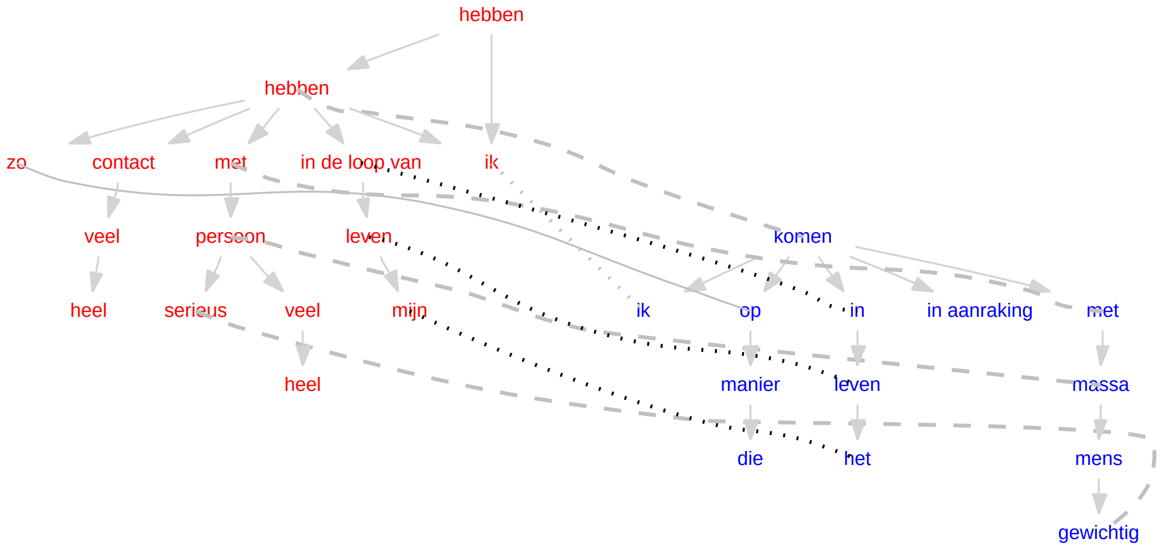

Figure 1: Dependency structures and alignment for the sentences *Zo heb ik in de loop van mijn leven heel veel contacten gehad met heel veel serieuze personen.* (lit. ‘Thus have I in the course of my life very many contacts had with very many serious persons’) and *Op die manier kwam ik in het leven met massa’s gewichtige mensen in aanraking.*. (lit. ‘In that way came I in the life with mass-of weighty/important people in touch’). The alignment relations are *equals* (dotted gray), *restates* (solid gray), *specifies* (dotted black), and *intersects* (dashed gray). For the sake of transparency, dependency relations have been omitted.

two sentences. Note that there is an intuitive relation with entailment here: both *equals* and *restates* can be understood as mutual entailment (i.e., if the root nodes of the analyses corresponding S and S' stand in an equal or restate relation, S entails S' and S' entails S), if S *specifies* S' then S also entails S' and if S *generalizes* S' then S is entailed by S'.

In remainder of this paper, we will distinguish two aspects of this task: **alignment** is the subtask of pairing related nodes – or more precise, pairing the token strings corresponding to these nodes; **classification of semantic relations** is the subtask of labeling these alignments in terms of the five types of semantic relations.

2.3 Annotation procedure

For creating manual alignments, we developed a special-purpose annotation tool which shows, side by side, two sentences, as well as their respective dependency graphs. When the user clicks on a node v in the graph, the corresponding string ($\text{STR}(v)$) is shown at the bottom. The tool enables the user to manually construct an alignment graph on the basis of the respective dependency graphs. This is done by focusing on a node in the structure for one sentence, and then selecting a corresponding node (if possible) in the other structure, after which the user can select the relevant alignment relation. The tool offers additional support for folding parts of the graphs, highlighting unaligned nodes and hiding dependency relation labels.

All text material was aligned by the two authors. They started with annotating the first ten sentences of chapter one together in order to get a feel for the task. They continued with the remaining sentences from chapter one individually (35 sentences and 521 in the first translation, and 35 sentences and 481 words in the second translation). Next, both annotators discussed annotation differences, which triggered some revisions in their respective annotation. They also agreed on a single consensus annotation. Interannotator agreement will be discussed in the next two sections. Finally, each author annotated two additional chapters, bringing the total to five.

3 Alignment

3.1 Interannotator agreement

Interannotator agreement was calculated in terms of precision, recall and F-score (with $\beta = 1$) on aligned

	(A_1, A_2)	$(A_{1'}, A_{2'})$	$(A_c, A_{1'})$	$(A_c, A_{2'})$
#real:	322	323	322	322
#pred:	312	321	323	321
#correct:	293	315	317	318
precision:	.94	.98	.98	.99
recall:	.91	.98	.98	.99
F-score:	.92	.98	.98	.99

Table 1: Interannotator agreement with respect to alignment between annotators 1 and 2 before (A_1, A_2) and after $(A_{1'}, A_{2'})$ revision , and between the consensus and annotator 1 $(A_c, A_{1'})$ and annotator 2 $(A_c, A_{2'})$ respectively.

node pairs as follows:

$$precision = \mid A_{real} \cap A_{pred} \mid / \mid A_{pred} \mid \tag{1}$$

$$recall = \mid A_{real} \cap A_{pred} \mid / \mid A_{real} \mid \tag{2}$$

$$F\text{-}score = \ (2 \times prec \times rec) \ / \ (prec + rec) \tag{3}$$

where A_{real} is the set of all real alignments (the reference or golden standard), A_{pred} is the set of all predicted alignments, and $A_{pred} \cap A_{real}$ is the set all correctly predicted alignments. For the purpose of calculating interannotator agreement, one of the annotations (A_1) was considered the 'real' alignment, the other (A_2) the 'predicted'. The results are summarized in Table 1 in column (A_1, A_2).[1]

As explained in section 2.3, both annotators revised their initial annotations. This improved their agreement, as shown in column $(A_{1'}, A_{2'})$. In addition, they agreed on a single consensus annotation (A_c). The last two columns of Table 1 show the results of evaluating each of the revised annotations against this consensus annotation. The F-score of $.98$ can therefore be regarded as the upper bound on the alignment task.

3.2 Automatic alignment

Our tree alignment algorithm is based on the dynamic programming algorithm in (Meyers et al., 1996), and similar to that used in (Barzilay, 2003). It calculates the match between each node in dependency tree D against each node in dependency tree D'. The score for each pair of nodes only depends on the similarity of the words associated with the nodes and, recursively, on the scores of the best matching pairs of their descendants. The node similarity function relies either on identity of the lemmas or on synonym, hyperonym, and hyponym relations between them, as retrieved from EuroWordNet.

Automatic alignment was evaluated with the consensus alignment of the first chapter as the gold standard. A baseline was constructed by aligning those nodes which stand in an *equals* relation to each other, i.e., a node v in D is aligned to a node v' in D' iff $\text{STR}(v) = \text{STR}(v')$. This baseline already achieves a relatively high score (an F-score of .56), which may be attributed to the nature of our material: the translated sentence pairs are relatively close to each other and may show a sizeable amount of literal string overlap. In order to test the contribution of synonym and hyperonym information for node matching, performance is measured with and without the use of EuroWordNet. The results for automatic alignment are shown in Table 2. In comparison with the baseline, the alignment algorithm without use of EuroWordnet loses a few points on precision, but improves a lot on recall (a 200% increase), which in turn leads to a substantial improvement on the overall F-score. The use of EurWordNet leads to a small increase (two points) on both precision and recall, and thus to small increase in F-score. However, in comparison with the gold standard human score for this task (.95), there is clearly room for further improvement.

4 Classification of semantic relations

4.1 Interannotator agreement

In addition to alignment, the annotation procedure for the first chapter of *The little prince* by two annotators (cf. section 2.3) also involved labeling of the semantic relation between aligned nodes. Interannotator agreement on this task is shown Table 3, before and after revision. The measures are *weighted* precision, recall and F-score. For instance, the precision is the weighted sum of the separate precision scores for each of the five relations. The table also shows the κ-score. The F-score of .97 can be regarded as the upper bound on the relation labeling task. We think these numbers indicate that the classification of semantic relations is a well defined task which can be accomplished with a high level of interannotator agreement.

[1] Note that since there are no classes, we can not calculate change agreement rethe $Kappa$ statistic.

$Alignment:$	$Prec:$	$Rec:$	*F-score:*
baseline	.87	.41	.56
algorithm without wordnet	.84	.82	.83
algorithm with wordnet	.86	.84	.85

Table 2: Precision, recall and F-score on automatic alignment

	(A_1, A_2)	$(A_{1'}, A_{2'})$	$(A_c, A_{1'})$	$(A_c, A_{2'})$
precision:	.86	.96	.98	.97
recall:	.86	.95	.97	.97
F-score:	.85	.95	.97	.97
κ:	.77	.92	.96	.96

Table 3: Interannotator agreement with respect to semantic relation labeling between annotators 1 and 2 before (A_1, A_2) and after $(A_{1'}, A_{2'})$ revision , and between the consensus and annotator 1 $(A_c, A_{1'})$ and annotator 2 $(A_c, A_{2'})$ respectively.

4.2 Automatic classification

For the purpose of *automatic* semantic relation labeling, we approach the task as a classification problem to be solved by machine learning. Alignments between node pairs are classified on the basis of the lexical-semantic relation between the nodes, their corresponding strings, and – recursively – on previous decisions about the semantic relations of daughter nodes. The input features used are:

- a boolean feature representing string identity between the strings corresponding to the nodes
- a boolean feature for each of the five semantic relations indicating whether the relation holds for at least one of the daughter nodes;
- a boolean feature indicating whether at least one of the daughter nodes is *not* aligned;
- a categorical feature representing the lexical semantic relation between the nodes (i.e. the lemmas and their part-of-speech) as found in EuroWordNet, which can be *synonym*, *hyperonym*, or *hyponym*.[2]

To allow for the use of previous decisions, the nodes of the dependency analyses are traversed in a bottom-up fashion. Whenever a node is aligned, the classifier assigns a semantic label to the alignment. Taking previous decisions into account may cause a proliferation of errors: wrong classification of daughter nodes may in turn cause wrong classification of the mother node. To investigate this risk, classification experiments were run both with and without (i.e. using the annotation) previous decisions.

	$Prec:$	$Rec:$	*F-score:*
equals	.93 ± .06	.95 ± .04	.94 ± .02
restates	.56 ± .08	.78 ± .04	.65 ± .05
specifies	$n.a.$	0	$n.a.$
generalizes	.19 ± .06	.37 ± .09	.24 ± .05
intersects	$n.a.$	0	$n.a.$
Combined:	.62 ± .01	.70 ± .02	.64 ± .02

Table 4: Average precision, recall and F-score (and SD) over all 5 folds on automatic classification of semantic relations

Since our amount of data is limited, we used a memory-based classifier, which – in contrast to most other machine learning algorithms – performs no abstraction, allowing it to deal with productive but low-frequency exceptions typically occurring in NLP tasks(Daelemans et al., 1999). All memory-based learning was performed with TiMBL, version 5.1 (Daelemans et al., 2004), with its default settings (overlap distance function, gain-ratio feature weighting, $k = 1$).

The five first chapters of *The little prince* were used to run a 5-fold cross-validated classification experiment. The first chapter is the consensus alignment and relation labeling, while the other four were done by one out of two annotators. The alignments to be classified are those from to the *human* alignment. The baseline of always guessing *equals* – the majority class – gives a precision of 0.26, a recall of 0.51, and an F-score of 0.36. Table 4 presents the results broken down to relation type. The combined F-score of 0.64 is almost twice the baseline score. As expected, the highest score goes to *equals*, followed by a reasonable score on *restates*. Performance on the other relation types is rather poor, with even no predictions of *specifies* and *intersects* at all.

Faking perfect previous decisions by using the annotation gives a considerable improvement, as shown in Table 5, especially on *specifies*, *generalizes* and *intersects*. This reveals that the proliferation of classification errors is indeed a problem that should be addressed.

[2] These three form the bulk of all relations in Dutch EuroWordnet. Since no word sense disambiguation was involved, we simply used all word senses.

	$Prec:$	$Rec:$	*F-score:*
equals	.99 ± .02	.97 ± .02	.98 ± .01
restates	.65 ± .04	.82 ± .04	.73 ± .03
specifies	.60 ± .12	.48 ± .10	.53 ± .09
generalizes	.50 ± .11	.52 ± .10	.50 ± .09
intersects	.69 ± .27	.35 ± .12	.46 ± .16
Combined:	.82 ± .02	.81 ± .02	.80 ± .02

Table 5: Average precision, recall and F-score (and SD) over all 5 folds on automatic classification of semantic relations without using previous decisions.

In sum, these results show that automatic classification of semantic relations is feasible and promising – especially when the proliferation of classification errors can be prevented – but still not nearly as good as human performance.

5 Discussion and Future work

This paper presented an approach to detecting semantic relations at the word, phrase and sentence level on the basis of dependency analyses. We investigated the performance of human annotators on the tasks of manually aligning dependency analyses and of labeling the semantic relations between aligned nodes. Results indicate that humans can perform this task well, with an F-score of $.98$ on alignment and an F-score of $.92$ on semantic relations (after revision). We also described and evaluated automatic methods addressing these tasks: a dynamic programming tree alignment algorithm which achieved an F-score on alignment of $.85$ (using lexical semantic information from EuroWordNet), and a memory-based semantic relation classifier which achieved F-scores of $.64$ and $.80$ with and without using real previous decisions respectively.

One of the issues that remains to be addressed in future work is the effect of parsing errors. Such errors were not corrected, but during manual alignment, we sometimes found that substrings could not be properly aligned because the parser had failed to identify them as syntactic constituents. As far as classification of semantic relations is concerned, the proliferation of classification errors is an issue that needs to be solved. Classification performance may be further improved with additional features (e.g. phrase length information), optimization, and more data. Also, we have not yet tried to combine automatic alignment and classification. Yet another point concerns the type of text material. The sentence pairs from our current corpus are relatively close, in the sense that both translations more or less convey the same information. Although this seems a good starting point to study alignment, we intend to continue with other types of text material in future work. For instance, in extending our work to the actual output of a QA system, we expect to encounter sentences with far less overlap.

References

R. Barzilay. 2003. *Information Fusion for Multidocument Summarization*. Ph.D. Thesis, Columbia University.

G. Bouma, G. van Noord, and R. Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In *Computational Linguistics in The Netherlands 2000*, pages 45–59.

W. Daelemans, A. Van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. van den Bosch. 2004. TiMBL: Tilburg memory based learner, version 5.1, reference guide. ILK Technical Report 04-02, Tilburg University.

I. Dagan and O. Glickman. 2004. Probabilistic textual entailment: Generic applied modelling of language variability. In *Learning Methods for Text Understanding and Mining*, Grenoble.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the ACL*, Sapporo, Japan.

J. Herrera, A. Pe nas, and F. Verdejo. 2005. Textual entailment recognition based on dependency analysis and wordnet. In *Proceedings of the 1st. PASCAL Recognision Textual Entailment Challenge Workshop*. Pattern Analysis, Statistical Modelling and Computational Learning, PASCAL.

A. Meyers, R. Yangarber, and R. 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.

F.J. Och and H. Ney. 2000. Statistical machine translation. In *EAMT Workshop*, pages 39–46, Ljubljana, Slovenia.

V. Punyakanok, D. Roth, and W. Yih. 2004. Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics*, 6(9).

L. Vanderwende, D. Coughlin, and W. Dolan. 2005. What syntax can contribute in entailment task. In *Proceedings of the 1st. PASCAL Recognision Textual Entailment Challenge Workshop*, Southampton, U.K.