Chapter 8
Automatic Tree Matching for Analysing Semantic Similarity in Comparable Text

Erwin Marsi and Emiel Krahmer

8.1 Introduction

Natural languages allow us to express essentially the same underlying meaning in a virtually unlimited number of alternative surface forms. In other words, there are often many similar ways to say the same thing. This characteristic poses a problem for natural language processing applications. Automatic summarisers, for example, typically rank sentences according to their informativity and then extract the top n sentences, depending on the required compression ratio. 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 summarisation, where sentences extracted from related documents are very likely to express similar information in different ways [21]. Provided 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. Yet

E. Marsi (✉)
Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
e-mail: emarsi@idi.ntnu.no

E. Krahmer
Department of Communication and Information Sciences, Tilburg University (UvT), Tilburg, The Netherlands
e-mail: e.j.krahmer@uvt.nl

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_8,

another application is in the area of automatic evaluation of machine translation output [20]. The general problem is that even though system output does not superficially match any of the human-produced gold standard translations, it may still be a good translation provided that it expresses the same semantic content. Measuring the semantic similarity between system output and reference translations may therefore be a better alternative to the more superficial evaluation measures currently in use.

In addition to merely *detecting* semantic similarity, we can ask to what extent two expressions share meaning. For instance, the meaning of a sentence can be fully contained in that of another, it may 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 summarisation, 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 [1, 16].

In this paper we present a 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, alignments are labeled according to the type of similarity relation that holds between the aligned phrases, which supports further processing. For instance, Marsi and Krahmer [8, 16] describe how to generate different types of sentence fusions on the basis of this relation labelling.

This chapter is structured in the following way. The next section defines the task of matching syntactic trees and labelling alignments in a more formal way. This is followed by an overview of the DAESO corpus, a large parallel monolingual treebank for Dutch, which forms the basis for developing and testing our approach. Section 8.4 outlines an algorithm for simultaneous node alignment and relation labelling. The results of some evaluation experiments are reported in Sect. 8.5. We finish with a discussion of related work and a conclusion.

8.2 Analysing Semantic Similarity

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

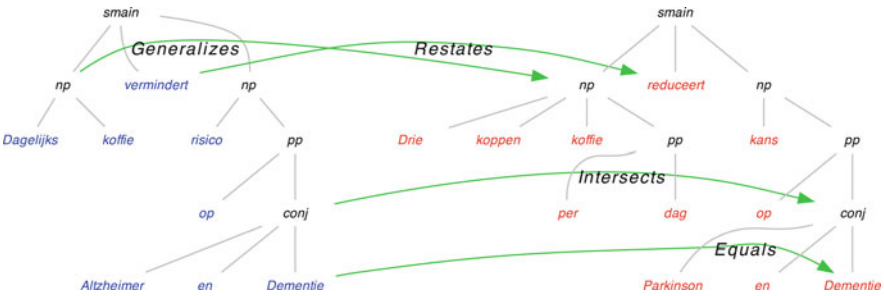

Fig. 8.1 Example of two aligned and labeled syntactic trees. For expository reasons the alignment is not exhaustive

linguistic processing. We therefore argue that the middle ground between these two extremes currently offers the best solution: analysing semantic similarity by means of syntactic tree alignment.

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 subsequence 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 Alzheimer en Dementie.
Daily coffee diminishes risk of Alzheimer and Dementia.
b. Drie koppen koffie per dag reduceert kans op Parkinson en Dementie.
Three cups coffee a day reduces chance of Parkinson and Dementia.

The corresponding syntax trees and their (partial) alignment is shown in Fig. 8.1. 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 **generalises** v' iff $\text{STR}(v)$ is more general than $\text{STR}(v')$ – example: *daily coffee* generalises *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 *daily coffee*;

5. v **intersects** v' iff STR(v) and 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 [4]. Note also that relations are prioritised: *equals* takes precedence over *restates*, etc. Furthermore, *equals*, *restates* and *intersects* are symmetrical, whereas *generalises* is the inverse of *specifies*. Finally, nodes containing unique information, such as *Alzheimer* and *Parkinson*, remain unaligned.

8.3 DAESO Corpus

The DAESO[1] corpus is a parallel monolingual treebank for Dutch that contains parallel and comparable Dutch text from several text domains:

- Alternative Dutch translations of a number of foreign language books
- Auto-cue (text that is automatically presented to a news reader) and subtitle text from news broadcasts by Dutch and Belgium public television channels
- Similar headlines from online news obtained from the Dutch version of Google News
- Similar answers from a Question-Answer corpus in the medical domain
- Press releases about the same news event from two major Dutch press agencies

All text was preprocessed in a number of steps. First, text was obtained by extraction from electronic documents or by OCR and converted to XML. All text material was subsequently processed with a tokeniser for Dutch [22]. OCR and tokenisation errors were in part manually corrected. Next, the Alpino parser for Dutch [2] was used to parse sentences. It provides a relatively theory-neutral syntactic analysis originally developed for the Spoken Dutch Corpus [25]. It is a blend of phrase structure analysis and dependency analysis, with a backbone of phrasal constituents and arcs labeled with syntactic function/dependency labels. Due to time and cost constraints, parsing errors were not subject to manual correction.

The next stage involved aligning similar sentences (regardless of their syntactic structure). This involved automatic alignment using heuristic methods, followed by manual correction using a newly developed alignment annotation tool, called *Hitaext*, for visualising and editing alignments between textual segments.[2] Annotator guidelines specified that aligned sentences must minimally share a "proposition", i.e. a predication over some entity. Just sharing a single entity (typically an noun)

[1] The acronym of the *Detecting And Exploiting Semantic Overlap* research project which gave rise to the corpus; see also http://daeso.uvt.nl

[2] http://daeso.uvt.nl/hitaext

or single predicate (typically a verb or adjective) is insufficient. This prevents alignment of trees which share virtually no content later on.

The final stage consisted of analysing the semantic similarity of aligned sentences along the lines described in the previous section. This included manual alignment of syntactic nodes, as well as labelling these alignments with one of five semantic relations. This work was carried out by six specially trained annotators. For creating and labelling alignments, a special-purpose graphical annotation tool called *Algraeph* was developed.[3]

The resulting corpus comprises over 2.1 M tokens, 678 K of which is manually annotated and 1,511 K is automatically processed. It is freely available for research purposes.[4] It is unique in its size and detailed annotations, and holds great potential for a wide range of research areas.

8.4 Memory-Based Graph Matcher

In order to automatically perform the alignment and labelling tasks described in Sect. 8.2, we cast these tasks simultaneously as a combination of exhaustive pairwise classification using a supervised machine learning algorithm, followed by global optimisation of the alignments using a combinatorial optimisation 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 [3], 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 normalised entropy of the class labels in the set of nearest neighbours (H) defined as

[3] http://daeso.uvt.nl/algraeph

[4] www.tst-centrale.org

$$H = -\frac{\sum_{c \in C} p(c) \log_2 p(c)}{\log_2 |C|} \tag{8.1}$$

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 0 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 results in one-to-many alignment of nodes. In order to reduce this to just one-to-one alignments, we search for a node matching which minimises the sum of costs over all alignments. This is a well-known problem in combinatorial optimisation 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* [9]. The output of the algorithm is the labeled tree matching obtained by removing all node alignments labeled with the special *none* relation.

8.5 Experiments

8.5.1 Experimental Setup

These experiments focus on analysing semantic similarity between sentences rather than merely detecting similarity (as a binary classification task). Hence it is assumed that there is at least some semantic overlap between comparable sentences and the task is a detailed analysis of this similarity in terms of a labeled alignment of syntactic constituents.

8.5.1.1 Data Sets

For developing and testing our alignment algorithm, we used half of the manually aligned press releases from the DAESO corpus. This data was divided into a development and held-out test set. The left half of Table 8.1 summarises the respective sizes of development and 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. The right half of Table 8.1 gives the distribution of semantic relations in the development and test sets. It can be observed that the distribution is fairly skewed with *equals* being the majority class.

Development was carried out using ten-fold cross validation on the development data and consequently reported scores on the development data are average scores

Table 8.1 Properties of development and test data sets

Data	Graph pairs	Node pairs	Tokens	Aligned nodes (%)	Equals (%)	Restates (%)	Specifies (%)	Generalises (%)	Intersects (%)
Develop	2,664	22,741	45,149	47.20	56.61	6.57	7.52	6.38	22.91
Test	547	4,894	10,005	47.05	58.40	7.11	7.40	6.38	20.72

over ten folds. Only two parameters were optimised on the development set. First, the amount of downsampling of the *none* class was fixed at 20 %; this will be motivated in Sect. 8.5.3. Second, the parameter k of the memory-based classifier – the number of nearest neighbours taken into account during classification – was evaluated in the range from 1 to 15. It was found that $k = 5$ provided the best trade-off between performance and speed. These optimised settings were then applied when testing on the held-out test data.

8.5.1.2 Features

All features used during classification are described in Table 8.2. 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 [2]. 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 [27], an improved and extended version of 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, so a certain amount of noise is present in these features. In addition, a background corpus of over 500M words of (mainly) news text provides the word counts required to calculate the Lin similarity measure [11]. 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.

We have not yet performed any systematic feature selection experiments. However, we did experiment with a substantial number of other features and combinations. The current feature set resulted from manual tuning on the development set. When removing any of these features, we observed decreased performance.

8.5.1.3 Evaluation Measures

A tree alignment A is a set of node alignments $< v, v' >$ where v and v' are source and target nodes respectively. As sets can be compared using the well-known

Table 8.2 Features[a] used during classification step

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
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	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
cornet-generalises	float	Lin similarity score if source word is a hypernym of target word
cornet-intersects	float	Lin similarity score if source word share a common hypernym
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
same-parent-cat	bool	test if parents of source and target have same syntactic category
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 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 nodes are identical
source/target-phrase-len	int	length of source/target phrase in words
phrase-len-diff	int	phrase length difference in words

[a]slashes indicate multiple versions of the same feature, e.g. *source/target-pos* represents the two features *source-pos* and *target-pos*

precision and *recall* measures [26], the same measures can be applied to alignments. Given that A_{true} is a true tree alignment and A_{pred} is a predicted tree alignment, precision and recall are defined as follows:

$$precision = \frac{|A_{true} \cap A_{pred}|}{|A_{pred}|} \tag{8.2}$$

$$recall = \frac{|A_{true} \cap A_{pred}|}{|A_{true}|} \tag{8.3}$$

Precision and recall are combined in the F_1 score, which is defined as the harmonic mean between the two, giving equal weight to both terms, i.e. $F_1 score = (2 * precision * recall)/(precision + recall)$

The same measures can be used for comparing *labeled* tree alignments in a straight forward way. Recall that a labeled tree alignment is a set of labeled node alignments $< v, v', r >$ where v is a source node, v' a target node and r is a label from the set of semantic similarity relations. Let A^{rel} be the subset of all alignments in A with label *rel*, i.e. $A^{rel} = \{< v_s, v_t, r > \in A : r = rel\}$. This allows us to calculate, for example, precision on relation *equals* as follows.

$$precision^{EQ} = \frac{|A^{EQ}_{true} \cap A^{EQ}_{pred}|}{|A^{EQ}_{pred}|} \tag{8.4}$$

We thus calculate precision as in the unlabelled case, but ignore all alignments – whether true or predicted – labeled with a different relation. Recall and F score on a particular relation can be calculated in a similar fashion.

8.5.2 Results on Tree Alignment

Table 8.3 presents the results on tree alignment consisting of baseline, human and MBGM scores.

8.5.2.1 Baseline Scores

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 *specifies*, *generalises* or *intersects* relations, as that would require a more involved, knowledge-based approach, relying on resources such as a wordnet.

8.5.2.2 Human Scores

A subset of the test data, consisting of 10 similar press releases comprising a total of 48 sentence pairs, was independently annotated by 6 annotators to determine

Table 8.3 Scores (in percentages) on tree alignment and semantic relation labelling

			Labelling:						
	Alignment:		Eq:	Re:	Spec:	Gen:	Int:	Macro:	Micro:
	Prec:	82.50	83.76	46.72	0.00	0.00	0.00	26.10	82.18
Develop baseline:	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
	Prec:	86.40	95.08	45.22	41.45	44.95	64.17	58.18	78.66
Develop MBGM:	Rec:	86.06	96.16	35.86	31.16	39.06	72.21	54.89	78.35
	F:	86.23	95.62	40.00	35.58	41.80	67.95	56.19	78.51
	Prec:	84.23	85.68	42.24	0.00	0.00	0.00	25.58	84.14
Test baseline:	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
	Prec:	86.87	95.96	51.79	40.43	38.36	60.87	57.48	78.10
Test MBGM:	Rec:	86.46	96.27	40.56	32.20	34.23	70.35	54.72	77.88
	F:	86.66	96.11	45.49	35.85	36.18	65.27	55.78	77.99
Human:	F:	88.31	95.83	71.38	60.21	66.71	62.67	71.36	81.92

inter-annotator agreement on the alignment and labelling tasks. Given the six annotations $A_1, \ldots, A_6$, we repeatedly took one as the A_{true} against which the five other annotations were evaluated as A_{pred}. We then computed the average scores over these $6 * 5 = 30$ scores.[5] This resulted in an F-score of 88.31 % on alignment only. For relation labelling, the scores differed per relation, as is to be expected: the average F-score for *equals* was 95.83 % alignment,[6] and for the other relations average F-scores between 62 and 72 % were obtained.

8.5.2.3 System Scores

The first thing to observe is that the MBGM scores on the development and tests sets are very similar throughout, suggesting that generalisation across the news domain is fairly good. We will therefore focus on the test scores, comparing them statistically with the baseline scores and informally with the human scores.

With an alignment F-score on the test set of 86.66 %, MBGM scores over 19 % higher than the baseline system, which is significant ($t(18) = 25.68$, $p < 0.0001$). This gain is mainly due to a much better recall score. This F-score is also less than

[5] As a result of this procedure, precision, recall and F score end up being equal.

[6] At first sight, it may seem that labelling *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.

2 % lower than the average alignment F-score obtained by our human annotators, albeit on a subset of test data.

In a similar vein, the performance of MBGM on relation labelling is considerably better than that of the baseline system, significantly outperforming the baseline for each semantic relation ($t(18) > 12.6636, p < 0.0001$), trivially so for the *specifies*, *generalises specifies* and *intersects* relations, which the baseline system never predicts.

The macro scores are plain averages over the five scores on each relation, whereas the micro scores are weighted averages. As *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*. MBGM scores a macro F-score of 55.78 % (an improvement of over 33 % over the baseline) and a micro average of 77.99 % (over 10 % above the baseline). It is interesting to observe that MBGM obtains *higher* F-scores on *equals* and *intersects* (the two most frequent relations) than the human annotators. As a result of this, the micro F-score of the automatic tree alignment is merely 4 % lower than the human reference counterpart. However, MBGM's macro F-score (55.78) is still well below the human score (71.36).

8.5.3 Effects of Downsampling

As described in Sect. 8.4, MBGM performs tree alignment by initially considering every possible alignment from source nodes to target nodes. For each possible pairing of a source node n_s in tree T_s and a target node n_t in tree T_t, an instance is created consisting of feature values extracted from the input trees. A memory-based classifier is then used to predict a class label for each instance, either one of the semantic similarity relations or the special class *none*, which is interpreted as *no alignment*. The vast majority of the training instances is of class *none*, because a node is aligned to at most one node in the other tree and unaligned to all other nodes in the same tree. The class distribution in the development data is: *equals* 0.81 %, *restates* 0.08 %, *specifies* 0.07 %, *generalises* 0.10 %, *intersects* 0.31 %, *none* 98.63 %. The problem is that most classifiers have difficulties with handling heavily skewed class distributions, usually causing them to always predict the majority class. We address this by downsampling the *none* class (in the training data) so that less frequent classes become more likely to be predicted.

The effects of downsampling are shown in Fig. 8.2 where precision, recall and F-score are plotted as a function of the percentage of original *none* instances in the training data. The training and test material correspond to a 90/10 % split of the development data. Timbl was used with its default settings, except for $k = 5$. The first plot shows scores on alignment regardless of relation labelling. The general trend is that downsampling increases the recall at the cost of precision until a cross-over point at around 20 %. This effect is mainly due to the fact that downsampling increases the number of predictions other than *none*.

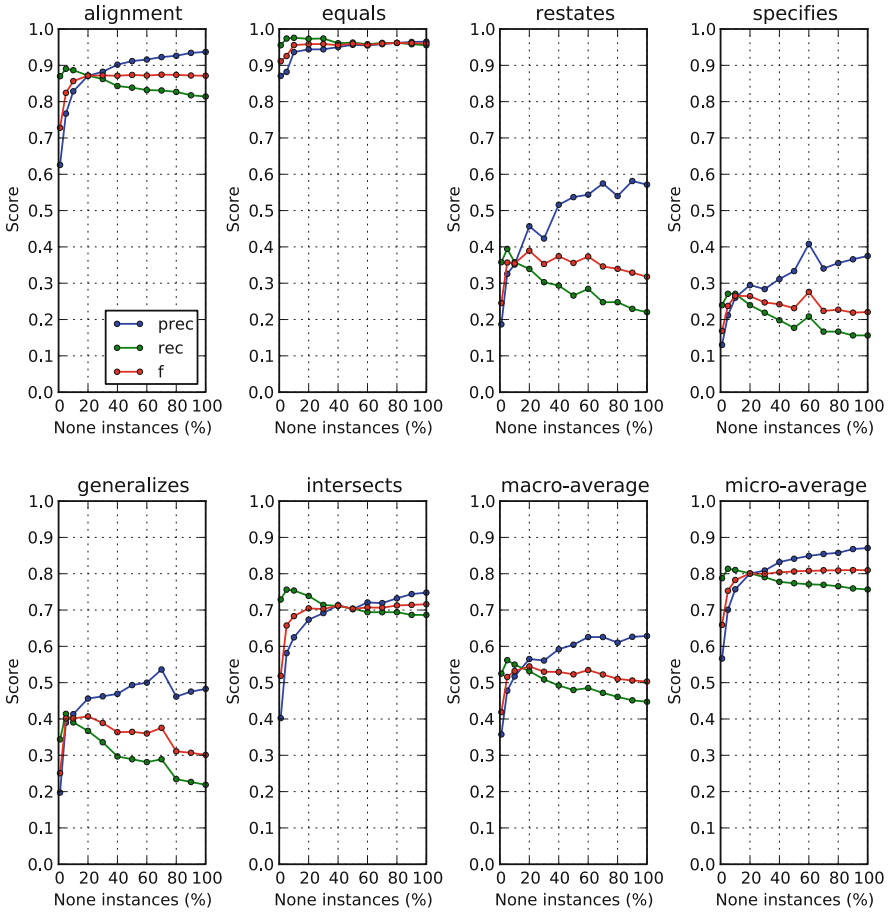

Fig. 8.2 Effects of downsampling *none* instances with regard to precision, recall and F-score, first for alignment only (i.e. ignoring relation label), next per alignment relation and finally as macro/micro average over all relations

The next five plots show the effect of downsampling per alignment relation. The cross-over point is higher for *equals* and *intersects*, at about 40 %. As these are still relatively frequent relations, their F-score is not negatively affected by all the *none* instances. However, for the least frequent relations – *restates*, *specifies*, *generalises* – it can be observed that the F-score is going down when using more than 20 % of the *none* instances. A pattern that is reflected in the macro-average plot (i.e. plain average score over all five relations), while the micro-average plot (i.e. weighted average) is more similar to those for *equals* and *intersects*, as it is dominated by these two most frequent relations.

Even though the alignment only and micro-average F-scores are marginally best without any downsampling, we choose to report results with downsampling *none* to

20 %, because this yields the optimal macro-average F-score. Arguably the optimal downsampling percentage may be specific to the data set and may change with, for example, more training data or another value of the k parameter in nearest neighbour classification.

8.5.4 Effects of Training Data Size

To study the effects of more training data on the scores, experiments were run gradually increasing the amount of training data from 1 up to 100 %. The experimental setting was the same as described in the previous section, including a constant downsampling to 20 % of the *none* class. The resulting learning curves are shown in Fig. 8.3. The learning curve for alignment only suggests that the learner is saturated at about 50 % of the training data, after which precision and recall are virtually identical and the F-score improves only very slowly. With regard to the alignment relations, *equals* and *intersects* show similar behaviour, with arguably no gain in performance after using more than half of the training data. Being dominated by these two relations, the same goes for the micro average scores. For *restates* and *generalises*, however, we find that scores are getting better, and further improvement may therefore be expected with even more training data. The only outlier is *specifies*, with scores that appear to go down somewhat when more training data is consumed. Until further study, we consider this an artefact of the test data. The general trend that the learner is not yet saturated with training samples for the less frequent relations is also reflected in the still improving macro-average scores.

8.6 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., [5, 6, 10, 24]. Most related to our approach is the work on discriminative tree alignment by Tiedemann and Kotzé [23]. 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 [12] for further arguments and empirical evidence that MT alignment algorithms are not suitable for aligning parallel monolingual text.

Recognising textual entailments (RTE) could arguably be seen as a specific instance of detecting semantic similarity [4]. The RTE task is commonly defined as: given a text T (usually consisting of one or two sentences) determine whether a sentence H (the hypothesis) is entailed by T. Various researchers have attempted to use alignments between T and H to predict textual entailments [7, 18]. However, these RTE systems have a directional bias (i.e., they assume the text is longer

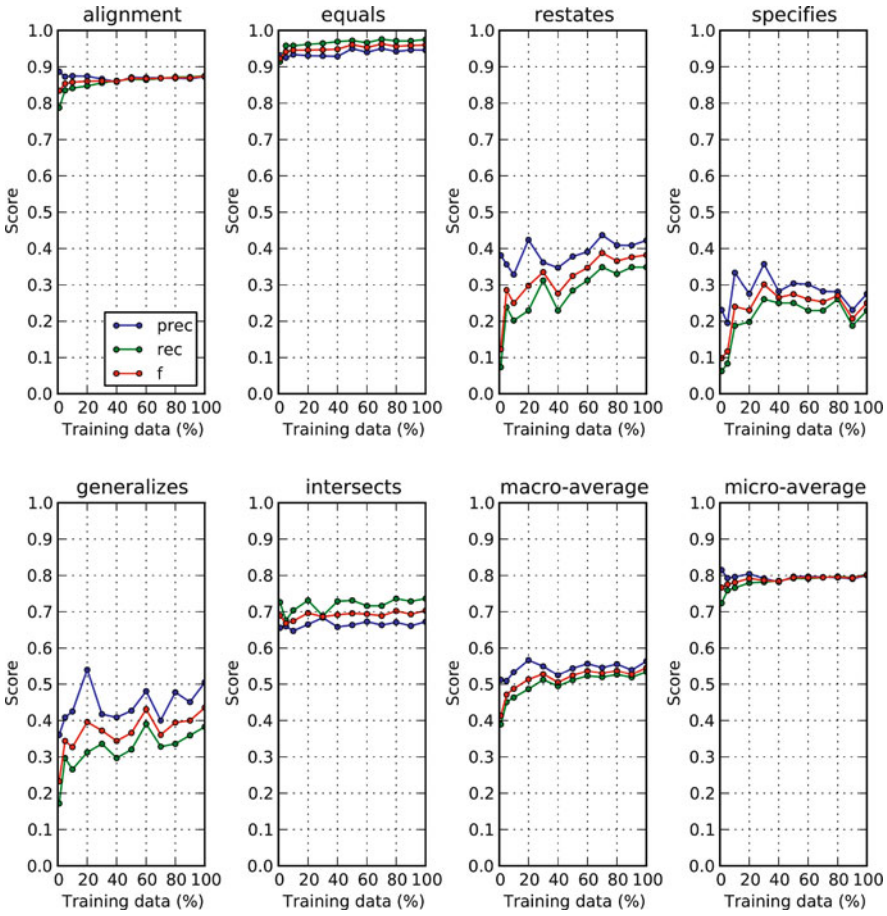

Fig. 8.3 Effects of training data size on precision, recall and F-scores, first for alignment only (i.e. ignoring relation label), next per alignment relation and finally as macro/micro average over all relations

than the the hypothesis), and apart from an entailment judgement do not provide an analysis of semantic similarity. Our *specifies* relation may be interpreted as entailment and vice versa, our *generalises* relation as reversed entailment. Likewise, *restates* may be regarded as mutual entailment. The *intersects* relation, however, cannot be stated in terms of entailment, which makes our relations somewhat more expressive. For instance, it can express the partial similarity in meaning between "*John likes milk*" and "*John likes movies*". In a similar way, contradictory statements such as "*John likes milk*" versus "*John hates milk*" can not be distinguished from completely unrelated statements such as "*John likes milk*" and "*Ice is cold*" in terms of entailment. In contrast, *intersects* is capable of capturing the partial similarity between contradictory statements.

Marneffe et al. [14] align semantic graphs for textual inference in machine reading, both manually and automatically. Although they do use typed dependency graphs, the alignment is only at the token level, and no explicit phrase alignment is carried out. As part of manual annotation, alignments are labeled with relations akin to ours (e.g. 'directional' versus 'bi-directional'), but their automatic alignment does not include labelling. MacCartney, Galley, and Manning [12] 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 [13] for modelling semantic containment and exclusion in natural language inference. Marsi and Krahmer [15] 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 is that it cannot cope with crossing alignments, which occur frequently in the manually aligned DAESO corpus. We are not aware of other work that combines alignment with semantic relation labelling, or algorithms which perform both tasks simultaneously.

8.7 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 reviewed the DAESO corpus, a parallel monolingual treebank for Dutch consisting of over two million tokens and covering both parallel and comparable text genres. It provides detailed analyses of semantically similar sentences in the form of syntactic node alignments and alignment relation labelling. We have subsequently presented a Memory-based Graph Matcher (MBGM) that performs both of these tasks simultaneously as a combination of exhaustive pairwise classification using a memory-based learning algorithm, and global optimisation of alignments using a combinatorial optimisation 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 the DAESO corpus show that MBGM consistently and significantly outperforms the baseline, both for alignment and labelling.

In future research we will test MBGM on other data, as the DAESO corpus contains other segments with various degrees of semantic overlap. We also intend to explore additional features which facilitate learning of lexical and syntactic paraphrasing patterns, for example, vector space models for word similarity. In addition, a comparison with other alignment systems, such as Giza++ [19], would provide a stronger baseline.

Acknowledgements This work was conducted within the DAESO project with participation from Tilburg University (Emiel Krahmer and Erwin Marsi), Antwerp University (Walter Daelemans and Iris Hendrickx), University of Amsterdam (Maarten de Rijke and Edgar Meij) and Textkernel (Jakub Zavrel and Martijn Spitters). We would also like to thank our three anonymous reviewers for their constructive criticism. Some parts of this work have been published before, most notably in [17].

References

1. Barzilay, R., McKeown, K.R.: Sentence fusion for multidocument news summarization. Comput. Linguist **31**(3), 297–328 (2005)
2. Bouma, G., van Noord, G., Malouf, R.: Alpino: Wide-coverage computational analysis of Dutch. In: Daelemans, W., Sima'an, K., Veenstra, J., Zavre, J. (eds.) Computational Linguistics in the Netherlands 2000, pp. 45–59. Rodopi, Amsterdam/New York (2001)
3. Daelemans, W., Zavrel, J., Van der Sloot, K., Van den Bosch, A.: TiMBL: Tilburg Memory Based Learner, version 6.2, reference manual. Tech. Rep. ILK 09-01, Induction of Linguistic Knowledge, Tilburg University (2009)
4. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK (2005)
5. Gildea, D.: Loosely tree-based alignment for machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Sapporo, Japan, pp. 80–87 (2003)
6. Groves, D., Hearne, M., Way, A.: Robust sub-sentential alignment of phrase-structure trees. In: Proceedings of the 20th International Conference on Computational Linguistics (CoLing '04), Geneva, Switzerland, pp. 1072–1078 (2004)
7. Herrera, J., nas, A.P., Verdejo, F.: Textual entailment recognition based on dependency analysis and wordnet. In: Proceedings of the 1st PASCAL Recognision Textual Entailment Challenge Workshop, Southampton, UK (2005)
8. Krahmer, E., Marsi, E., van Pelt, P.: Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In: Moore, J., Teufel, S., Allan, J., Furui, S. (eds.) Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Columbus, Ohio, USA, pp. 193–196 (2008)
9. Kuhn, H.W.: The Hungarian Method for the assignment problem. Nav. Res. Logist. Q. **2**, 83–97 (1955)
10. Lavie, A., Parlikar, A., Ambati, V.: 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, Columbus, Ohio, USA, pp. 87–95 (2008)
11. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, Madison, Wisconsin, USA, pp. 296–304 (1998)
12. MacCartney, B., Galley, M., Manning, C.D.: A phrase-based alignment model for natural language inference. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, pp. 802–811 (2008)

13. MacCartney, B., Manning, C.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, Manchester, UK, pp. 521–528 (2008)
14. de Marneffe, M., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., Manning, C.: Aligning semantic graphs for textual inference and machine reading. In: Proceedings of the AAAI Spring Symposium, Stanford, USA (2007)
15. Marsi, E., Krahmer, E.: Classification of semantic relations by humans and machines. In: Proceedings of the ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, Michigan, pp. 1–6 (2005)
16. Marsi, E., Krahmer, E.: Explorations in sentence fusion. In: Proceedings of the 10th European Workshop on Natural Language Generation, Aberdeen, GB (2005)
17. Marsi, E., Krahmer, E.: Automatic analysis of semantic similarity in comparable text through syntactic tree matching. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pp. 752–760. Coling 2010 Organizing Committee, Beijing, China (2010)
18. Marsi, E., Krahmer, E., Bosma, W., Theune, M.: Normalized alignment of dependency trees for detecting textual entailment. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy, pp. 56–61 (2006)
19. Och, F.J., Ney, H.: Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, pp. 440–447. Association for Computational Linguistics, Stroudsburg, PA, USA (2000)
20. Padó, S., Cer, D., Galley, M., Jurafsky, D., Manning, C.: Measuring machine translation quality as semantic equivalence: a metric based on entailment features. Mach. Transl. **23**, 181–193 (2009)
21. Radev, D., McKeown, K.: Generating natural language summaries from multiple on-line sources. Comput. Linguist. **24**(3), 469–500 (1998)
22. Reynaert, M.: Sentence-splitting and tokenization in d-coi. Tech. Rep. 07-07, ILK Research Group (2007)
23. Tiedemann, J., Kotzé, G.: Building a large machine-aligned parallel treebank. In: Eighth International Workshop on Treebanks and Linguistic Theories, Milan, Italy, p. 197 (2009)
24. Tinsley, J., Zhechev, V., Hearne, M., Way, A.: Robust language-pair independent sub-tree alignment. Mach. Transl. Summit **XI**, 467–474 (2007)
25. van der Wouden, T., Hoekstra, H., Moortgat, M., Renmans, B., Schuurman, I.: Syntactic analysis in the spoken dutch corpus. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain, pp. 768–773 (2002)
26. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworth, London/Boston (1979)
27. Vossen, P., Maks, I., Segers, R., van der Vliet, H.: Integrating lexical units, synsets and ontology in the Cornetto Database. In: Proceedings of the LREC 2008, Marrakech, Morocco (2008)