Explorations in Sentence Fusion

Marsi, E.C.; Krahmer, E.J.

Published in:
Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)

Publication date:
2005

Explorations in Sentence Fusion*

Erwin Marsi and Emiel Krahmer
Communication and Cognition
Faculty of Arts, Tilburg University
P.O.Box 90153, NL-5000 LE Tilburg, The Netherlands
{e.c.marsi, e.j.krahmer}@uvt.nl

Abstract

Sentence fusion is a text-to-text (revision-like) generation task which takes related sentences as input and merges these into a single output sentence. In this paper we describe our ongoing work on developing a sentence fusion module for Dutch. We propose a generalized version of alignment which not only indicates which words and phrases should be aligned but also labels these in terms of a small set of primitive semantic relations, indicating how words and phrases from the two input sentences relate to each other. It is shown that human labelers can perform this task with a high agreement (F-score of .95). We then describe and evaluate our adaptation of an existing automatic alignment algorithm, and use the resulting alignments, plus the semantic labels, in a generalized fusion and generation algorithm. A small-scale evaluation study reveals that most of the resulting sentences are adequate to good.

1 Introduction

Traditionally, Natural Language Generation (NLG) is defined as the automatic production of "meaningful texts in (...) human language from some underlying non-linguistic representation of information" [Reiter and Dale, 2000, xvii]. Recently, there is an increased interest in NLG applications that produce meaningful text *from meaningful text* rather than from abstract meaning representations. Such applications are sometimes referred to as **text-to-text generation** applications (e.g., [Chandrasekar and Bangalore, 1997], [Knight and Marcu, 2002], [Lapata, 2003]), and may be likened to earlier revision-based generation strategies, e.g. [Robin, 1994] [Callaway and Lester, 1997]. Text-to-text generation is often motivated from practical applications such as summarization, sentence simplification, and sentence compression. One reason for the interest in such generation systems is the possibility to automatically learn text-to-text generation strategies from corpora of parallel text.

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

In this paper, we take a closer look at **sentence fusion** [Barzilay, 2003][Barzilay *et al.*, 1999], one of the interesting variants in text-to-text generation. A sentence fusion module takes related sentences as input, and generates a single sentence summarizing the input sentences. The general strategy described in [Barzilay, 2003] is to first align the **dependency structures** of the two input sentences to find the common information in both sentences. On the basis of this alignment, the common information is framed into an fusion tree (i.e., capturing the shared information), which is subsequently realized in natural language by generating all traversals of the fusion tree and scoring their probability using an n-gram language model. Of the sentences thus generated the one with the lowest (length normalized) entropy is selected.

Barzilay and co-workers apply sentence fusion in the context of multi-document summarization, where the input sentences typically come from multiple documents describing the same event, but sentence fusion seems to be useful for other applications as well. In **question-answering**, for instance, sentence fusion could be used to generate more *complete* answers. Many current QA systems use various parallel answer-finding strategies, each of which may produce an N-best list of answers (e.g., [Maybury, 2004]) In response to a question like "What causes RSI?" one potential answer sentence could be:

> RSI can be caused by repeating the same sequence of movements many times an hour or day.

And another might be:

> RSI is generally caused by a mixture of poor ergonomics, stress and poor posture.

These two incomplete answers might be fused into a more complete answer such as:

> RSI can be caused by a mixture of poor ergonomics, stress, poor posture and by repeating the same sequence of movements many times an hour or day.

The same process of sentence fusion can of course be applied to the whole list of N-best answers in order to derive a more specific, or even the most specific, answer, akin to taking the union of a number of sets. Likewise, we can rely on sentence fusion to derive a more general answer, or even the most general one (cf. intersection), in the hope that this will filter out irrelevant parts of the answer.

Arguably, such applications call for a generalized version of sentence fusion, which may have consequences for the various components (alignment, fusion and generation) of the sentence fusion pipeline. At the **alignment** level, we would like to have a better understanding of how words and phrases in the input sentences relate to each other. Rather than a binary choice (align or not), one might want to distinguish more fine-grained relations such as overlap (if two phrases share some but not all of their content), paraphrases (if two phrases express the same information in different ways), entailments (if one phrase entails the other, but not vice versa), etc. Such an alignment strategy would be especially useful for applications such as question answering and information extraction, where it is often important to know whether two sentences are paraphrases or stand in an entailment relation [Dagan and Glickman, 2004]. In the **fusion** module, we are interested in the possibilities to generate various kinds of fusions depending on the relations between the respective sentences, e.g., selecting the more specific or the more general phrase depending on whether the fusion tree is an intersection or a union one. Finally, the **generation** may be more complicated in the generalized version, and it is an interesting question whether the use of language models is equally suitable for different kinds of fusion.

In this paper, we will explore some of these issues related to a generalized version of sentence fusion. We start with the basic question whether it is possible at all to reliably align sentences, including different potential relations between words and phrases (section 2). We then present our ongoing work on sentence fusion, describing the current status and performance of the alignment algorithm (section 3), as well as the fusion and generation components (section 4). We end with discussion and description of future plans in section 5.

2 Data collection and Annotation

2.1 General approach

Alignment has become standard practice in data-driven approaches to machine translation (e.g. [Och and Ney, 2000]). Initially work focused on word-based alignment, but more recent research also addresses alignment at the higher levels (substrings, syntactic phrases or trees), e.g.,[Gildea, 2003]. The latter approach seems most suitable for current purposes, where we want to express that a sequence of words in one sentence is related to a non-identical sequence of words in another sentence (a paraphrase, for instance). However, if we allow alignment of arbitrary substrings of two sentences, then the number of possible alignments grows exponentially to the number of tokens in the sentences, and the process of alignment – either manually or automatically – may become infeasible. An alternative, which seems to occupy the middle ground between word alignment on the one hand and alignment of arbitrary substrings on the other, is to align syntactic analyses. Here, following [Barzilay, 2003], we will align sentences at the level of **dependency structures**. Unlike to [Barzilay, 2003], we are interested in a number of different alignment relations between sentences, and pay special attention to the feasibility of this alignment task.

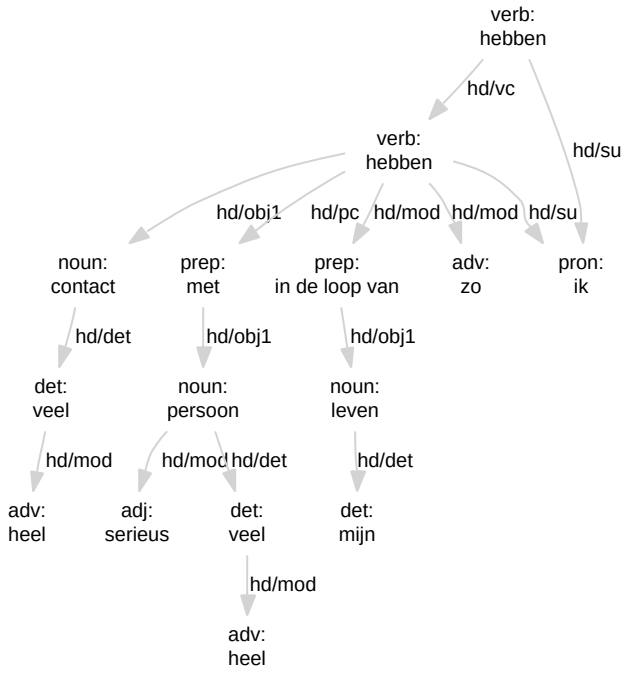

Figure 1: Example dependency structure for the sentence *Zo heb ik in the loop van mijn leven heel veel contacten gehad met heel veel serieuze personen.* (lit. 'Thus have I in the course of my life very many contacts had with very many serious persons').

2.2 Corpus

For evaluation and parameter estimation 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). 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 version mapped onto two sentences in the other: 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*, *pron*, and *adv*). The dependency relations hold between tokens and are the same as used in the Spoken Dutch Corpus (see e.g., [van der Wouden *et al.*, 2002]). These include dependencies such as *head/subject*, *head/modifier* and *coordination/conjunction*. See Figure 1 for an example. 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.

2.3 Task definition

A dependency analysis of a sentence S yields a labeled directed graph $D = \langle V, E \rangle$, where V (vertices) are the nodes, and E (edges) are the dependency relations. For each node v in the dependency structure for a sentence S, we define $\text{STR}(v)$ as the substring of all tokens under v (i.e., the composition of the tokens of all nodes reachable from v). For example, the string associated with node *persoon* in Figure 1 is *heel veel serieuze personen* ('very many serious persons').

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 relation between $\text{STR}(v)$ and $\text{STR}(v')$, i.e., 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 $\text{STR}(v)$ and $\text{STR}(v')$ are literally identical (abstracting from case and word order)
   Example: "a small and a large boa-constrictor" equals "a large and a small boa-constrictor";
2. v **restates** v' iff $\text{STR}(v)$ is a paraphrase of $\text{STR}(v')$ (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 $\text{STR}(v)$ is more specific than $\text{STR}(v')$,
   Example: "the planet B 612" specifies "the planet";
4. v **generalizes** v' iff $\text{STR}(v')$ is more specific than $\text{STR}(v)$,
   Example: "the planet" generalizes "the planet B 612";
5. v **intersects** v' iff $\text{STR}(v)$ and $\text{STR}(v')$ 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"

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'.

An alignment between S and S' can now formally be defined on the basis of the respective dependency graphs $D = \langle V, E \rangle$ and $D' = \langle V', E' \rangle$ as a graph $A = \langle V_A, E_A \rangle$, such that

$$E_A = \{\langle v, l, v' \rangle \mid v \in V \;\&\; v' \in V' \;\&\; l(\text{STR}(v), \text{STR}(v'))\},$$

where l is one of the five relations defined above. The nodes of A are those nodes from D en D' which are aligned, formally defined as

$$V_A = \{v \mid \exists v' \exists l \langle v, l, v' \rangle \in E_A\} \cup \{v' \mid \exists v \exists l \langle v, l, v' \rangle \in E_A\}$$

A complete example alignment can be found in the Appendix, Figure 3.

	(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.

2.4 Alignment tool

For creating manual alignments, we developed a special-purpose annotation tool called **Gadget** ('Graphical Aligner of Dependency Graphs and Equivalent Tokens'). It 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. See Figure 4 in the Appendix for a screen shot of Gadget.

2.5 Results

All text material was aligned by the two authors. They started doing 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. The total number of nodes in the two translations of the chapter was 445 and 399 respectively. Inter-annotator agreement was calculated for two aspects: alignment and relation labeling. With respect to alignment, we calculated the precision, recall and F-score (with $\beta = 1$) on aligned node pairs as follows:

$$precision(A_{real}, A_{pred}) = \frac{\mid A_{real} \cap A_{pred} \mid}{\mid A_{pred} \mid} \quad (1)$$

$$recall(A_{real}, A_{pred}) = \frac{\mid A_{real} \cap A_{pred} \mid}{\mid A_{real} \mid} \quad (2)$$

$$F\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

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 inter-annotator 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).

Next, both annotators discussed the differences in alignment, and corrected mistaken or forgotten alignments. This improved their agreement as shown in column $(A_{1'}, A_{2'})$. In

	(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 2: Inter-annotator agreement with respect to alignment **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.

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 .96 can therefore be regarded as the upper bound on the alignment task.

In a similar way, the agreement was calculated for the task of labeling the alignment relations. Results are shown in Table 2, where 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, which is another commonly used measure for inter-annotator agreement [Carletta, 1996]. Again, the F-score of .97 can be regarded as the upper bound on the relation labeling task.

We think these numbers indicate that the labeled alignment task is well defined and can be accomplished with a high level of inter-annotator agreement.

3 Automatic alignment

In this section, we describe the alignment algorithm that we use (section 3.1), and evaluate its performance (section 3.2).

3.1 Tree alignment algorithm

The tree alignment algorithm is based on [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. For an efficient implementation, dynamic programming is used to build up a score matrix, which guarantees that each score will be calculated only once.

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

$$S(v, v') = max \begin{cases} \textsc{TreeMatch}(v, v') \\ max_{i=1,\ldots,n}\ S(v_i, v') \\ max_{j=1,\ldots,m}\ S(v, v'_j) \end{cases} \quad (4)$$

where $v_1, \ldots, v_n$ denote the children of v and $v'_1, \ldots, v'_m$ denote the children of v'. The three terms correspond to the three ways that nodes can be aligned: (1) v can be directly aligned to v'; (2) any of the children of v can be aligned to v'; (3) v can be aligned to any of the children of v'. Notice that the last two options imply skipping one or more edges, and leaving one or more nodes unaligned.[1]

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

$$\textsc{TreeMatch}(v, v') = \textsc{NodeMatch}(v, v') + \max_{p \in \mathcal{P}(v,v')} \left[\sum_{(i,j) \in p} \left(\textsc{RelMatch}(\overrightarrow{v}_i, \overrightarrow{v}'_j) + S(v_i, v'_j) \right) \right]$$

Here $\overrightarrow{v}_i$ denotes the dependency relation from v to v_i. $\mathcal{P}(v, v')$ is the set of all possible pairings of the n children of v against the m children of v', which is the power set of $\{1, \ldots, n\} \times \{1, \ldots, m\}$. The summation in (5) ranges over all pairs, denoted by (i, j), which appear in a given pairing $p \in \mathcal{P}(v, v')$. Maximizing this summation thus amounts to finding the optimal alignment of children of v to children of v'.

$\textsc{NodeMatch}(v, v') \geq 0$ is a measure of how well the label of node v matches the label of v'.

$\textsc{RelMatch}(\overrightarrow{v}_i, \overrightarrow{v}'_j) \geq 0$ is a measure for how well the dependency relation between node v and its child v_i matches that of the dependency relation between node v' and its child v_j.

Since the dependency graphs delivered by the Alpino parser were usually not trees, they required some modification in order to be suitable input for the tree alignment algorithm. We first determined a root node, which is defined as a node from which all other nodes in the graph can be reached. In the rare case of multiple root nodes, an arbitrary one was chosen. Starting from this root node, any cyclic edges were temporarily removed during a depth-first traversal of the graph. The resulting directed acyclic graphs may still have some amount of structure sharing, but this poses no problem for the algorithm.

3.2 Evaluation of automatic alignment

We evaluated the automatic alignment of nodes, abstracting from relation labels, as we have no algorithm for automatic labeling of these relations yet. The baseline is achieved by aligning those nodes with stand in an *equals* relation to each other, i.e., a node v in D is aligned to a node v' in D' iff $\textsc{str}(v) = \textsc{str}(v')$. This alignment can be constructed relatively easy.

The alignment algorithm is tested with the following NODEMATCH function:

$$\textsc{NodeMatch}(v, v') = \begin{cases} 10 & \text{if } \textsc{str}(v) = \textsc{str}(v') \\ 5 & \text{if } \textsc{label}(v) = \textsc{label}(v') \\ 2 & \text{if } \textsc{label}(v) \text{ is a synonym} \\ & \text{hyperonym or hyponym} \\ & \text{of } \textsc{label}(v') \\ 0 & \text{otherwise} \end{cases}$$

[1] In the original formulation of the algorithm by [Meyers *et al.*, 1996], there is a penalty for skipping edges.

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

Table 3: Precision, recall and F-score on automatic alignment

It reserves the highest value for a literal string match, a somewhat lower value for matching lemmas, and an even lower value in case of a synonym, hyperonym or hyponym relation. The latter relations are retrieved from the Dutch part of EuroWordnet [Vossen, 1998]. For the RELMATCH function, we simply used a value of 1 for identical dependency relations, and 0 otherwise. These values were found to be adequate in a number of test runs on two other, manually aligned chapters (these chapters were not used for the actual evaluation). In the future we intend to experiment with automatic optimizations.

We measured the alignment accuracy defined as the percentage of correctly aligned node pairs, where the consensus alignment of the first chapter served as the golden standard. The results are summarized in Table 3. 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 show that the algorithm improves substantially on the baseline. The 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. The alignment algorithm (without use of EuroWordnet) loses a few points on precision, but improves a lot on recall (a 200% increase with respect to the baseline), which in turn leads to a substantial improvement on the overall F-score. The use of Eurowordnet leads to a small increase (two points) on both precision and recall (and thus to small increase on F-score). Yet, in comparison with the gold standard human score for this task (.95), there is clearly room for further improvement.

4 Merging and generation

The remaining two steps in the sentence fusion process are merging and generation. In general, **merging** amounts to deciding which information from either sentence should be preserved, whereas **generation** involves producing a grammatically correct surface representation. In order to get an idea about the baseline performance, we explored a simple, somewhat naive string-based approach. Below, the pseudocode is shown for merging two dependency trees in order to get restatements. Given a labeled alignment A between dependency graphs D and D', if there is a **restates** relation between node v from D and node v' from D', we add the string realization of v' as an alternative to those of v.

RESTATE(A)

1 **for** each edge $\langle v, l, v' \rangle \in E_A$
2 **do if** $l = \mathbf{restates}$
3 **then** STR$(v) \leftarrow$ STR$(v) \vee$ STR(v')

The same procedure is followed in order to get specifications:

SPECIFY(A)

1 **for** each edge $\langle v, l, v' \rangle \in E_A$
2 **do if** $l = \mathbf{generalizes}$
3 **then** STR$(v) \leftarrow$ STR$(v) \vee$ STR(v')

The generalization procedure adds the option to omit the realization of a modifier that is *not* aligned:

GENERALIZE(D, A)

1 **for** each edge $\langle v, l, v' \rangle \in E_A$
2 **do if** $l = \mathbf{specifies}$
3 **then** STR$(v) \leftarrow$ STR$(v) \vee$ STR(v')
4 **for** each edge $\langle v, l, v' \rangle \in E_D$
5 **do if** $l \in$ MOD-DEP-RELS and $v \notin E_A$
6 **then** STR$(v) \leftarrow$ STR$(v) \vee$ NIL

where MOD-DEP-REL is the set of dependency relations between a node and a modifier (e.g. *head/mod* and *head/predm*).

Each procedure is repeated twice, once adding substrings from D into D' and once the other way around. Next, we traverse the dependency trees and generate all string realizations, extending the list of variants for each node that has multiple realizations. Finally, we filter out multiple copies of the same string, as well as strings that are identical to the input sentences.

This procedure for merging and generation was applied to the 35 sentence pairs from the consensus alignment of chapter one of "Le Petit Prince". Overall this gave rise to 194 restatement, 62 specifications and 177 generalizations, with some sentence pairs leading to many variants and others to none at all. Some output showed only minor variations, for instance, substitution of a synonym. However, others revealed surprisingly adequate generalizations or specifications. Examples of good and bad output are given in Figure 2.

As expected, many of the resulting variants are ungrammatical, because constraints on word order, agreement or subcategorisation are violated. Following work on statistical surface generation [Langkilde and Knight, 1998] and other work on sentence fusion [Barzilay, 2003], we tried to filter ungrammatical variants with an n-gram language model. The Cambridge-CMU Statistical Modeling Toolkit v2 was used to train a 3-gram model on over 250M words from the Twente Newscorpus , using back-off and Good-Turing smoothing. Variants were ranked in order of increasing entropy. We found, however, that the ranking was often inadequate, showing ungrammatical variants at the top and grammatical variants in the lower regions.

To gain some insight into the general performance of the merging and generation strategy, we performed a small evaluation test in which the two authors independently judged all generated variants in terms of three categories:

1. **Perfect**: no problems in either semantics or syntax;
2. **Acceptable**: understandable, but with some minor flaws in semantics or grammar;
3. **Nonsense**: serious problems in semantics or grammar

Table 4 shows the number of sentences in each of the three categories per judge, broken down in restatements, generalization and specifications. The κ-score on this classification

Input1: Zo heb ik in de loop van mijn leven heel veel contacten gehad met heel veel serieuze personen
Thus have I in the course of my life very many contacts had with very many serious persons
Input2: Op die manier kwam ik in het leven met massa's gewichtige mensen in aanraking
In that way came I in the life with masses-of weighty/important people in touch
Restate: op die manier heb ik in de loop van mijn leven heel veel contacten gehad met heel veel serieuze personen
in that way have I in the course of my life very many contacts had with very many serious persons
Specific: op die manier kwam ik in de loop van mijn leven met massa's gewichtige mensen in aanraking
in that way have I in the course of my life with masses-of weighty/important people in touch
General: zo heb ik in het leven veel contacten gehad met veel serieuze personen
thus have I in the life many contacts had with many serious persons

Input1: En zo heb ik op mijn zesde jaar een prachtige loopbaan als kunstschilder laten varen .
And so have I at my sixth year a wonderful career as art-painter let sail
Input2: Zo kwam het , dat ik op zesjarige leeftijd een schitterende schildersloopbaan liet varen .
Thus came it , that I at six-year age a bright painter-career let sail
Specific: en zo heb ik op mijn zesde jaar als kunstschilder laten een schitterende schildersloopbaan varen
and so have I at my sixth year as art-painter let a bright painter-career sail
General: zo kwam het dat ik op leeftijd een prachtige loopbaan liet varen
so came it that I at age a wonderful career let sail

Figure 2: Examples of good (top) and bad (bottom) sentence fusion output

	Restate:		Specific:		General:	
	J1	J2	J1	J2	J1	J2
Perfect:	109	104	28	22	89	86
Acceptable:	44	58	15	16	34	24
Nonsense:	41	32	19	24	54	67
Total:	194		62		177	

Table 4: Results of the evaluation of the sentence fusion output as the number of sentences in each of the three categories *perfect*, *acceptable* and *nonsense* per judge (J1 and J2), broken down in restatements, generalizations and specifications.

task is .75, indicating a moderate to good agreement between the judges. Roughly half of the generated restatements and generalization are perfect, while this is not the case for specifications. We have no plausible explanation for this yet.

We think we can conclude from this evaluation that sentence fusion is a viable and interesting approach for producing restatements, generalization and specifications. However, there is certainly further work to do; the procedure for merging dependency graphs should be extended, and the realization model clearly requires more linguistic sophistication in particular to deal with word order, agreement and subcategorisation constraints.

5 Discussion and Future work

In this paper we have described our ongoing work on sentence fusion for Dutch. Starting point was the sentence fusion model proposed by [Barzilay *et al.*, 1999; Barzilay, 2003] in which dependency analyses of pairs of sentences are first aligned, after which the aligned parts (representing the common information) are fused. The resulting fused dependency tree is subsequently transfered into natural language. Our new contributions are primarily in two areas. First, we carried out an explicit evaluation of the alignment – both human and automatic alignment – whereas [Barzilay, 2003] only evaluates the output of the complete sentence fusion process. We found that annotators can reliably align phrases and assign relation labels to them, and that good results can be achieved with automatic alignment, certainly above an informed baseline, albeit still below human performance. Second, Barzilay and co-workers developed their sentence fusion model in the context of multi-document summarization, but arguably the approach could also be applicable for applications such as question answering or information extraction. This seems to call for a more refined version of sentence fusion, which has consequences for alignment, merging and realization. We have therefore introduced five different types of semantic relations between strings, namely equals, restates, specifies, generalizes and intersects. This increases the expressiveness of the representation, and supports generating restatements, generalizations and specifications. Finally, we described and evaluated our first results on sentence realization based on these refined alignments, with promising results.

Similar work is described in [Pang *et al.*, 2003], who describe a syntax-based algorithm that builds word lattices from parallel translations which can be used to generate new paraphrases. Their alignment algorithm is less refined, and there is only type of alignment and hence output (only restatements), but their mapping of aligned trees to a word lattice (or FSA) seems worthwhile to explore in combination with the approach we have proposed here.

One of the issues that remains to be addressed in future work is the effect of parsing errors. Such errors were not manually corrected, but during manual alignment, however, we sometimes found that substrings could not be properly aligned because the parser failed to identify them as syntactic constituents. The repercussions of this for the generation should be investigated by comparing the results obtained here with alignments on perfect parses. Furthermore, our work on

automatic alignment so far only concerned the alignment of nodes, not the determination of the relation type. We intend to address this task with machine learning, initially relying on shallow features such as the length of the respective token strings and the amount of overlap. It is also clear that more work is needed on merging and surface realization. One possible direction here is to exploit the relatively rich linguistic representation of the input sentences (POS tags, lemmas and dependency structures), for instance, along the lines of [Bangalore and Rambow, 2000]. Yet another issue concerns the type of text material. The sentence pairs from our current corpus are relatively close, in the sense that there is usually a 1-to-1 mapping between sentences, and 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. Of particular interest to us is also whether sentence fusion can be shown to improve the quality of QA system output.

References

[Bangalore and Rambow, 2000] Srinivas Bangalore and Owen Rambow. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 17th conference on Computational linguistics*, pages 42–48, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[Barzilay *et al.*, 1999] R. Barzilay, K. McKeown, and M. Elhaded. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, Maryland, 1999.

[Barzilay, 2003] R. Barzilay. *Information Fusion for Multi-document Summarization*. Ph.D. Thesis, Columbia University, 2003.

[Bouma *et al.*, 2001] Gosse Bouma, Gertjan van Noord, and Robert Malouf. Alpino: Wide-coverage computational analysis of dutch. In *Computational Linguistics in The Netherlands 2000*. 2001.

[Callaway and Lester, 1997] C. Callaway and J. Lester. Dynamically improving explanations: A revision-based approach to explanation generation. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 1997)*, pages 952–958, Nagoya, Japan, 1997.

[Carletta, 1996] Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, 22(2):249–254, 1996.

[Chandrasekar and Bangalore, 1997] R. Chandrasekar and S. Bangalore. Automatic induction of rules for text simplification. *Knowledge-based Systems*, 10(3):183–190, 1997.

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

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

[Imamura, 2001] K. Imamura. Hierarchical phrase alignment harmonized with parsing. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS 2001)*, pages 377–384, Tokyo, Japan, 2001.

[Knight and Marcu, 2002] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[Langkilde and Knight, 1998] Irene Langkilde and Kevin Knight. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th conference on Association for Computational Linguistics*, pages 704–710, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[Lapata, 2003] M. Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 545–552, Sapporo, 2003.

[Maybury, 2004] M. Maybury. *New Directions in Question Answering*. AAAI Press, 2004.

[Meyers *et al.*, 1996] A. Meyers, R. Yangarber, and R. Grisham. Alignment of shared forests for bilingual corpora. In *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, pages 460–465, Copenhagen, Denmark, 1996.

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

[Pang *et al.*, 2003] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *HLT-NAACL*, 2003.

[Reiter and Dale, 2000] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, 2000.

[Robin, 1994] J. Robin. *Revision-based generation of Natural Language Summaries Providing Historical Background*. Ph.D. Thesis, Columbia University, 1994.

[van der Wouden *et al.*, 2002] T. van der Wouden, H. Hoekstra, M. Moortgat, B. Renmans, and I. Schuurman. Syntactic analysis in the spoken dutch corpus. In *Proceedings of the third International Conference on Language Resources and Evaluation*, pages 768–773, Las Palmas, Canary Islands, Spain, 2002.

[Vossen, 1998] Piek Vossen, editor. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

6 Appendix

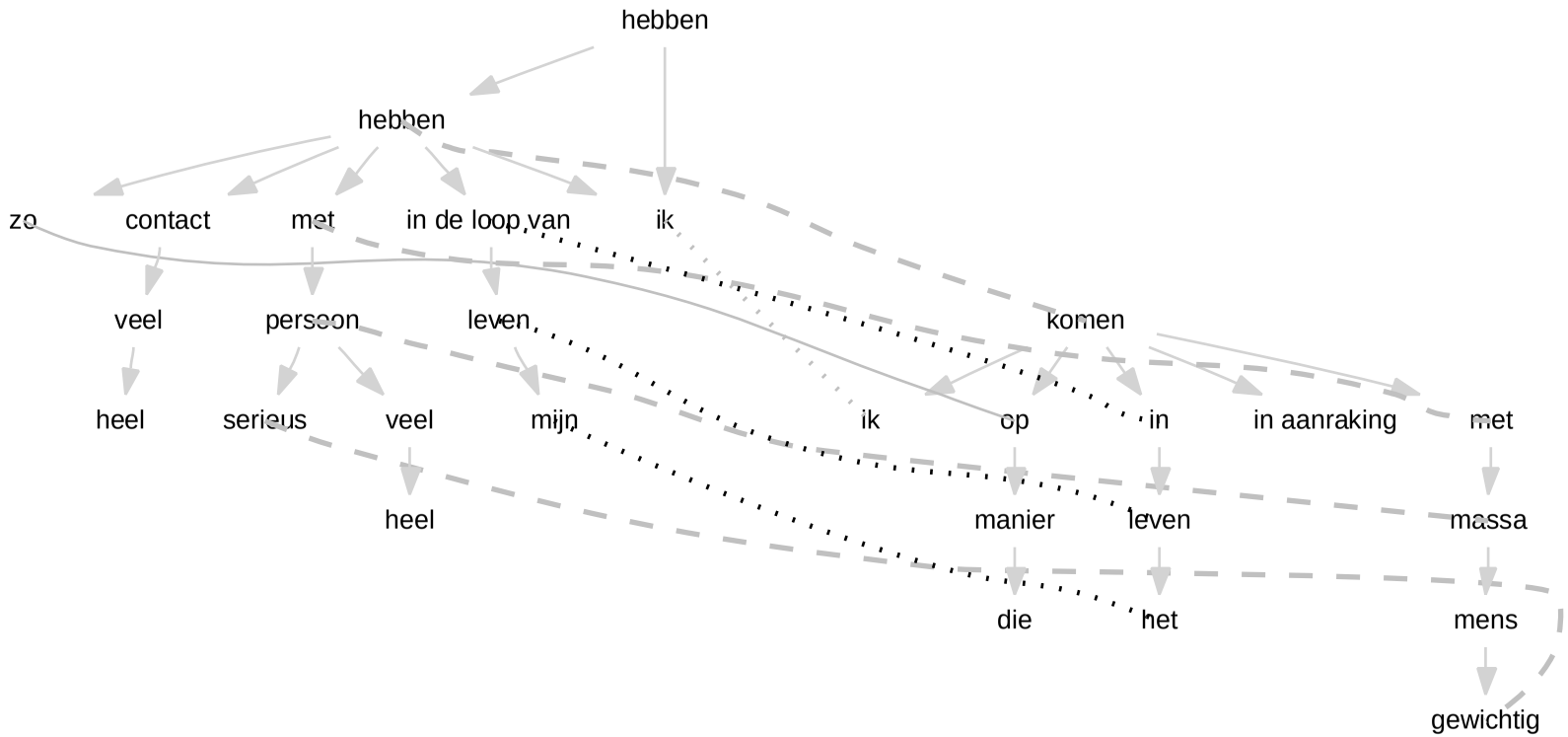

Figure 3: Dependency structures and alignment for the sentences *Zo heb ik in de loop van mijn leven heel veel contacten gehad met heel veel serieuze personen.* (lit. 'Thus have I in the course of my life very many contacts had with very many serious persons') and *Op die manier kwam ik in het leven 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.

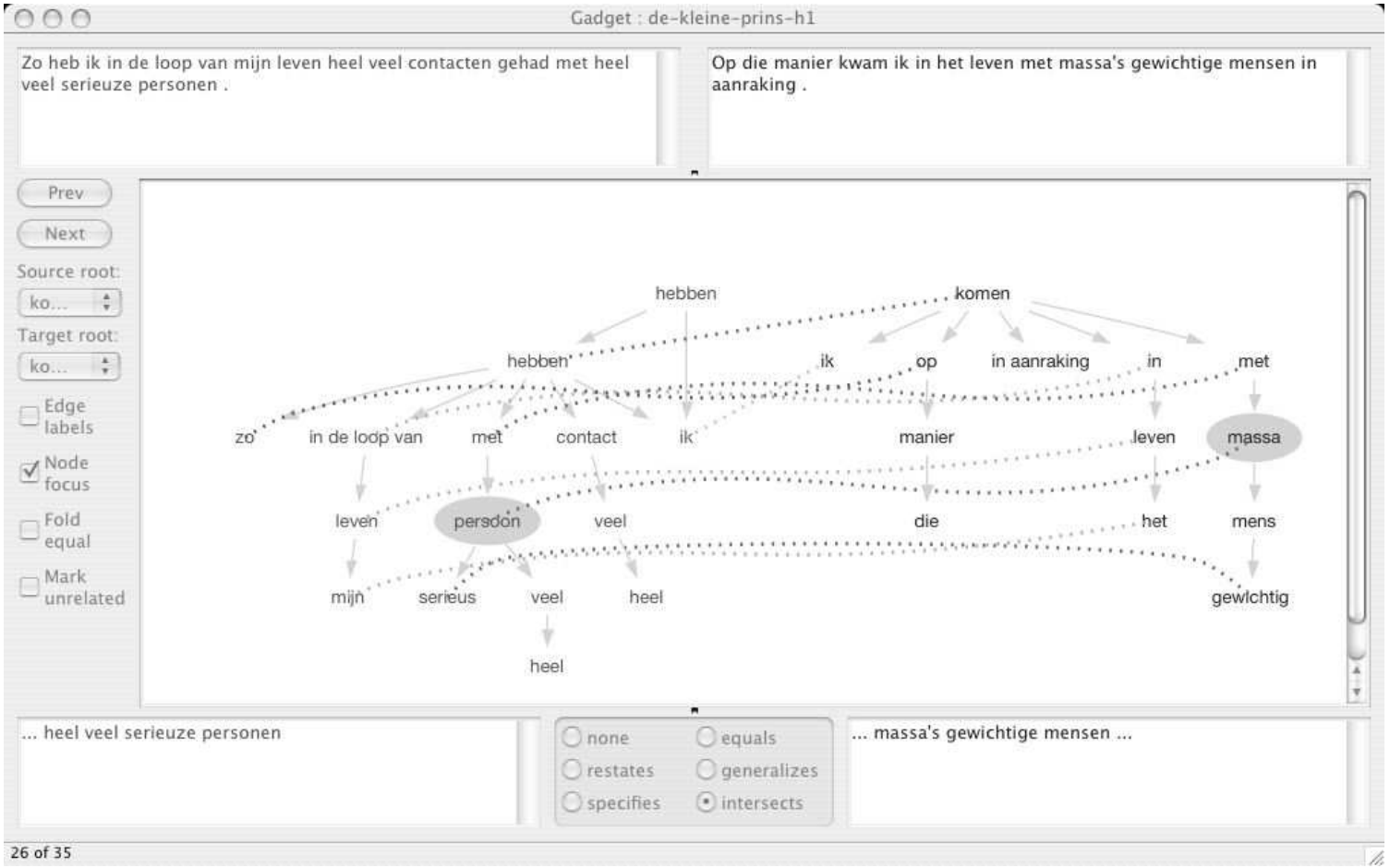

Figure 4: Screen shot of Gadget, the tool used for aligning dependency structures of sentences.