Preferences versus adaption during referring expression generation

Goudbeek, M.B.; Krahmer, E.J.

Published in:
Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

On the Limits of Sentence Compression by Deletion

Erwin Marsi[1], Emiel Krahmer[1], Iris Hendrickx[2], and Walter Daelemans[2]

[1] Tilburg University
Tilburg, The Netherlands
{emarsi,ekrahmer}@uvt.nl
http://daeso.uvt.nl
[2] Antwerp University
Antwerpen, Belgium
{iris.hendrickx,walter.daelemans}@ua.ac.be

Abstract. Data-driven approaches to sentence compression define the task as dropping any subset of words from the input sentence while retaining important information and grammaticality. We show that only 16% of the observed compressed sentences in the domain of subtitling can be accounted for in this way. We argue that this is partly due to the lack of appropriate evaluation material and estimate that a deletion model is in fact compatible with approximately 55% of the observed data. We analyse the remaining cases in which deletion only failed to provide the required level of compression. We conclude that in those cases word order changes and paraphrasing are crucial. We therefore argue for more elaborate sentence compression models which include paraphrasing and word reordering. We report preliminary results of applying a recently proposed more powerful compression model in the context of subtitling for Dutch.

1 Introduction

The task of *sentence compression* (or *sentence reduction*) can be defined as summarizing a single sentence by removing information from it [17]. The compressed sentence should retain the most important information and remain grammatical. One of the applications is in automatic summarization in order to compress sentences extracted for the summary [17,20]. Other applications include automatic subtitling [9,27,28] and displaying text on devices with very small screens [8].

A more restricted version of the task defines sentence compression as dropping any subset of words from the input sentence while retaining important information and grammaticality [18]. This formulation of the task provided the basis for the noisy-channel and decision-tree based algorithms presented in [18], and for virtually all follow-up work on data-driven sentence compression [4,5,12,19,24,26,27,30] It makes two important assumptions: (1) only word deletions are allowed – no substitutions or insertions – and therefore no paraphrases; (2) the word order is fixed. In other words, the compressed sentence must be a

E. Krahmer, M. Theune (Eds.): Empirical Methods in NLG, LNAI 5790, pp. 45–66, 2010.

subsequence of the source sentence. We will call this *the subsequence constraint*, and refer to the corresponding compression models as *word deletion models*. Another implicit assumption in most work is that the scope of sentence compression is limited to isolated sentences and that the textual context is irrelevant.

Under this definition, sentence compression is reduced to a word deletion task. Although one may argue that even this counts as a form of text-to-text generation, and consequently an NLG task, the generation component is virtually non-existent. One can thus seriously doubt whether it really is an NLG task.

Things would become more interesting from an NLG perspective if we could show that sentence compression necessarily involves transformations beyond mere deletion of words, and that this requires linguistic knowledge and resources typical to NLG. The aim of this chapter is therefore to challenge the deletion model and the underlying subsequence constraint. To use an analogy, our aim is to show that sentence compression is less like carving something out of wood - where material can only be removed - and more like molding something out of clay - where the material can be thoroughly reshaped. In support of this claim we provide evidence that the coverage of deletion models is in fact rather limited and that word reordering and paraphrasing play an important role.

The remainder of this chapter is structured as follows. In Section 2, we introduce our text material which comes from the domain of subtitling in Dutch. We explain why not all material is equally well suited for studying sentence compression and motivate why we disregard certain parts of the data. We also describe the manual alignment procedure and the derivation of edit operations from it. In Section 3, an analysis of the number of deletions, insertions, substitutions, and reorderings in our data is presented. We determine how many of the compressed sentences actually satisfy the subsequence constraint, and how many of them could in principle be accounted for. That is, we consider alternatives with the same compression ratio which do not violate the subsequence constraint. Next is an analysis of the remaining problematic cases in which violation of the subsequence constraint is crucial to accomplish the observed compression ratio. We single out (1) word reordering after deletion and (2) paraphrasing as important factors. Given the importance of paraphrases, Section 4 discusses the perspectives for automatic extraction of paraphrase pairs from large text corpora, and tries to estimate how much text is required to obtain a reasonable coverage. Section 5 reports on a pilot experiment in which we apply a recently proposed and more expressive model for sentence compression [7] to the same Dutch data set. We identify a number of problems with the model, both when applied to Dutch and in general. We finish with a summary and discussion in Section 6.

2 Material

We study sentence compression in the context of subtitling. The basic problem of subtitling is that on average reading takes more time than listening, so subtitles can not be a verbatim transcription of the speech without increasingly lagging behind. Subtitles can be presented at a rate of 690 to 780 characters per minute,

Table 1. Degree of sentence alignment: shows the distribution of the number of other sentences (ranging from zero to four) that a given sentence is aligned to, for both autocue and subtitle sentences

Degree:	Autocue:	(%)	Subtitle:	(%)
0	3607	(20.74)	12542	(46.75)
1	12382	(71.19)	13340	(49.72)
2	1313	(7.55)	901	(3.36)
3	83	(0.48)	41	(0.15)
4	8	(0.05)	6	(0.02)

while the average speech rate is considerably higher [28]. Subtitles are therefore often a compressed representation of the original spoken text.

Our text material stems from the *NOS Journaal*, the daily news broadcast of the Dutch public television. It is parallel text with on source side the *autocue* sentences (aut), i.e. the text the news reader is reading, and on the target side the corresponding *subtitle* sentences (sub). It was originally collected and processed in two earlier research projects – ATRANOS and MUSA – on automatic subtitling [9,27,28]. All text was automatically tokenized and aligned at the sentence level, after which alignments were manually checked.

The same material was further annotated in a project called DAESO[1] (Detecting And Exploiting Semantic Overlap), in which the general goal is automatic detection of semantic overlap. All aligned sentences were first syntactically parsed using the Alpino parser for Dutch [3], after which their parse trees were manually aligned in more detail. Pairs of similar syntactic nodes – either words or phrases – were aligned and labeled according to a set of five semantic similarity relations [22,23]. For current purposes, only the alignment at the word level is used, ignoring phrasal alignments and relation labels.

Not all material in this corpus is equally well suited for studying sentence compression as defined in the introduction. As we will discuss in more detail below, this prompted us to disregard certain parts of the data.

Sentence deletion, splitting and merging. For a start, autocue and subtitle sentences are often not in a one-to-one alignment relation. Table 1 specifies the alignment degree (i.e. the number of other sentences that a sentence is aligned to) for autocue and subtitle sentences. The first thing to notice is that there is a large number of unaligned subtitles. These correspond to non-anchor text from, e.g., interviews or reporters abroad. More interesting is that about one in five autocue sentences is completely dropped. A small number of about 4 to 8 percent of the sentence pairs are not one-to-one aligned. A long autocue sentence may be split into several simpler subtitle sentences, each containing only a part of the semantic content of the autocue sentence. Conversely, one or more - usually short - autocue sentences may be merged into a single subtitle sentence.

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

These decisions of sentence deletion, splitting and merging are worthy research topics in the context of automatic subtitling, but they should not be confused with sentence compression, the scope of which is by definition limited to single sentences. Accordingly we disregarded all sentence pairs in which autocue and subtitle are not in a one-to-one relation with each other. This reduced the data set from 17393 to 12382 sentence pairs.

Word compression. A significant part of the reduction in subtitle characters is actually not obtained by deleting words but by lexical substitution of a shorter token. Examples of this include substitution by digits ("7" for "seven"), abbreviations or acronyms ("US" for "United States"), symbols (euro symbol for "Euro"), or reductions of compound words ("elections" for "state-elections"). We will call this *word compression*. Although an important part of subtitling, we prefer to abstract from word compression and focus here on sentence compression proper. Removing all sentence pairs containing a word compression has the disadvantage of further reducing the data set. Instead we choose to measure *compression ratio* (CR) in terms of tokens[2] rather than characters.

$$CR = \frac{\#tok_{sub}}{\#tok_{aut}} \tag{1}$$

This means that the majority of the word compressions do not affect the sentence CR.

Variability in compression ratio. The CR of subtitles is not constant, but varies depending (mainly) on the amount of provided autocue material in a given time frame. The histogram in Figure 1 shows the distribution of the CR (measured in tokens) over all sentence pairs (i.e. one-to-one aligned sentences). In fact, autocue sentences are most likely not to be compressed at all (thus belonging to the largest bin, from 1.00 to 1.09 in the histogram).[3] In order to obtain a proper set of compression examples, we retained only those sentence pairs with a compression ratio less than one.

Parsing failures. As mentioned earlier detailed alignment of autocue and subtitle sentences was carried out on their syntactic trees. However, for various reasons a small number of sentences (0.2%) failed to pass the Alpino parser and received no parse tree. As a consequence, their trees could not be aligned and there is no alignment at the word level available either. Variability in CR and parsing failures are together responsible for a further reduction down to 5233 sentence pairs, the final size of our data set, with an overall CR of 0.69. Other properties of this data set are summarized in Table 2.[4]

[2] Throughout this study we ignore punctuation and letter case.

[3] Some instances even show a CR larger than one, because occasionally there is sufficient time/space to provide a clarification, disambiguation, update, or stylistic enhancement.

[4] Notice that *Sum* is not meaningful for CR.

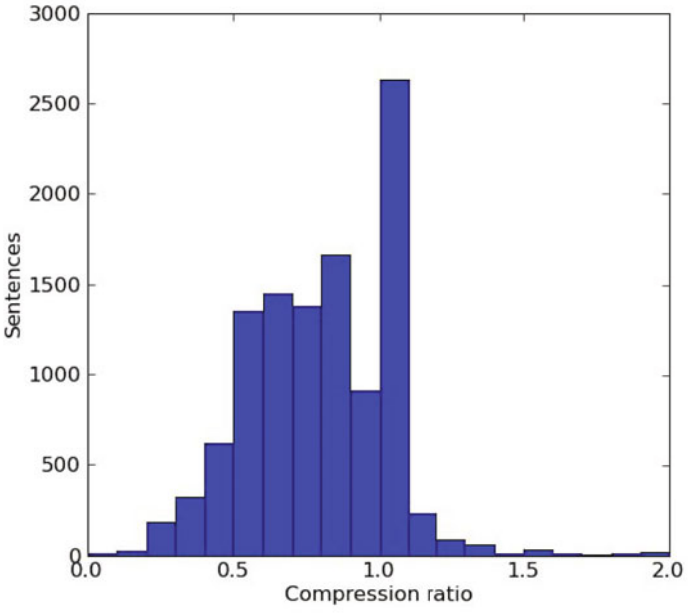

Fig. 1. Histogram of compression ratio: shows the distribution of the compression ratio (cf. equation 1) over all one-to-one aligned sentence pairs

Word deletions, insertions and substitutions Having a manual alignment of similar words in both sentences allows us to simply deduce word deletions, substitutions and insertions, as well as word order changes, in the following way:

- if an autocue word is not aligned to a subtitle word, then it is was deleted
- if a subtitle word is not aligned to an autocue word, then it was inserted
- if different autocue and subtitle words are aligned, then the former was substituted by the latter
- if alignments cross each other, then the word order was changed

The remaining option is that the aligned words are identical (ignoring differences in case).

Table 2. Properties of the final data set of 5233 pairs of autocue-subtitle sentences: minimum value, maximum value, total sum, mean and standard deviation for number of tokens per autocue/subtitle sentence and Compression Ratio

	Min:	Max:	Sum:	Mean:	SD:
aut-tokens	2	43	80651	15.41	5.48
sub-tokens	1	29	53691	10.26	3.72
CR	0.07	0.96		0.69	0.17

Without the word alignment, we would have to resort to automatically deriving the edit distance, i.e. the sum of the minimal number of insertions, deletions and substitutions required to transform one sentence into the other. However, this would result in different and often counter-intuitive sequences of edit operations. Our approach clearly distinguishes word order changes from the edit operations; the conventional edit distance, by contrast, can only account for changes in word order by sequences of the edit operations. Another problem with conventional edit distance is that substitution can also be accomplished by deletion and subsequent insertion, so we would either have to resort to assigning appropriate costs to the different operations or to abandon substitution altogether.

3 Analysis

3.1 Edit Operations

The observed deletions, insertions, substitutions and edit distances are shown in Table 3. For example, the minimum number of deletions observed in a single sentence pair (Min) is 1[5], whereas the maximum number of deletions observed in single sentence pair (Max) is 34. The total number of deletions over all sentence pairs is 34728, which amounts to a mean of 6.64 deletions per sentence pair and a standard deviation (SD) of 4.57. The rows for substitutions and insertions should be interpreted in a similar way. The edit distance is defined as the sum of all deletions, insertions and substitutions. On average, there are about 9 edit operations per sentence pair. As expected, deletion is the most frequent operation, with on average seven deletions per sentence. Insertion and substitutions are far less frequent. Note also that – even though the task is compression – insertions are somewhat more frequent than substitutions. Word order changes – which are not shown in the table – occur in 1688 cases, or 32.26% of all sentence pairs.

Another point of view is to look at the number of sentence pairs containing a certain edit operation. Here we find 5233 pairs (100.00%) with deletion, 2738 (52.32%) with substitution, 3263 (62.35%) with insertion, and 1688 (32.26%) with reordering.

Recall from the introduction that a subtitle is a *subsequence* of the autocue if there are no insertions, no substitutions, and no word order changes. In contrast, if any of these do occur, the subtitle is not a subsequence. The average CR for subsequences is 0.68 ($SD = 0.20$) versus 0.69 ($SD = 0.17$) for non-subsequences. A detailed inspection of the relation between the subsequence/non-subsequence ratio and CR revealed no clear correlation, so we did not find indications that non-subsequences occur more frequently at higher compression ratios.

3.2 Percentage of Subsequences

It turns out that only 843 (16.11%) subtitles are a subsequence, which is rather low. At first sight, this appears to be bad news for any deletion model, as it seems

[5] Every sentence pair must have at least one deletion, because by definition the CR must be less than one for all pairs in the data set.

Table 3. Observed word deletions, insertions, substitutions and edit distances

	Min:	Max:	Sum:	Mean:	SD:
del	1	34	34728	6.64	4.57
sub	0	6	4116	0.79	0.94
ins	0	17	7768	1.48	1.78
dist	1	46	46612	8.91	5.78

to imply that the model cannot account for close to 84% of the observed data. However, the important thing to keep in mind is that compression of a given sentence is a problem for which there are usually multiple solutions [2]. This is exactly what makes it so hard to perform automatic evaluation of NLG systems. There may very well exist semantically equivalent alternatives with the same CR which do satisfy the subsequence constraint. For this reason, a substantial part of the observed non-subsequences may have subsequence counterparts which can be accounted for by a deletion model. The question is: how many?

In order to address this question, we took a random sample of 200 non-subsequence sentence pairs. In each case we tried to come up with an alternative subsequence subtitle with the same meaning and the same CR (or when opportune, even a lower CR), but without compromising grammaticality. The task was carried out by one of the authors and subsequently checked by another author (both native speakers of Dutch), resulting in only a few minor improvements. Table 4 shows the distribution of the difference in tokens between the original non-subsequence subtitle and the manually-constructed equivalent subsequence subtitle. It demonstrates that 95 out of 200 (47%)

Table 4. Distribution of difference in tokens between original non-subsequence subtitle and equivalent subsequence subtitle

token-diff:	count:	(%:)
-2	4	2.00
-1	18	9.00
0	73	36.50
1	42	21.00
2	32	16.00
3	11	5.50
4	9	4.50
5	5	2.50
7	2	1.00
8	2	1.00
9	1	0.50
11	1	0.50

subsequence subtitles have the same (or even fewer) tokens, and thus the same (or an even lower) compression ratio. This suggests that the subsequence constraint is not as problematic as it seemed and that the coverage of a deletion model is in fact far better than it appeared to be. Recall that 16% of the original subtitles were already subsequences, so our analysis suggests that a deletion model can provide adequate output for 55% (16% plus 47% of 84%) of the data.

3.3 Problematic Non-subsequences

Another result of this exercise in rewriting subtitles is that it allows us to identify those cases in which the attempt to create a proper subsequence fails. In (1), we show one representative example of a problematic subtitle, for which the best equivalent subsequence we could obtain still has nine more tokens than the original non-subsequence.

(1) **Aut** *de bron was een geriatrische patient die zonder het zelf te*
the source was a geriatric patient who without it self to
merken uitzonderlijk veel larven bij zich bleek te dragen
notice exceptionally many larvae with him appeared to carry
en een grote verspreiding veroorzaakte
and a large spreading caused
"the source was a geriatric patient who unknowingly carried exceptionally many larvae and caused a wide spreading"

Sub *een geriatrische patient met larven heeft de verspreiding*
a geriatric patient with larvae has the spreading
veroorzaakt
caused

Seq *de bron was een geriatrische patient die veel larven bij*
the source was a geriatric patient who many larvae with
zich bleek te dragen en een verspreiding veroorzaakte
him appeared to carry and a spreading caused

These problematic non-subsequences reveal where insertion, substitution and/or word reordering are essential to obtain a subtitle with a sufficient CR (i.e. the CR observed in the real subtitles). At least three different types of phenomena were observed.

Word order. In some cases deletion of a constituent necessitates a change in word order to obtain a grammatical sentence. In example (2), the autocue sentence has the PP modifier *in verband met de lawineramp in galür* in its topic position (first sentence position).

(2) **Aut** *in verband met de lawineramp in galür hebben de*
in relation to the avalanche-disaster in Galtür have the
politieke partijen in tirol gezamenlijk besloten de
political parties in Tirol together decided the
verkiezingscampagne voor het regionale parlement op te
election-campaign for the regional parliament up to
schorten
postpone
"Due to the avalanche disaster in Galür, political parties in Tirol have decided to postpone the elections for the regional parliament."

Sub *de politieke partijen in tirol hebben besloten de verkiezingen*
the political parties in Tirol have decided the elections
op te schorten
up to postpone

Deleting this modifier, as is done in the subtitle, results in a sentence that starts with the verb *hebben*, which is interpreted as a yes-no question. For a declarative interpretation, we have to move the subject *de politieke partijen* to the first position, as in the subtitle. Incidentally, this indicates that it is instructive to apply sentence compression models to multiple languages, as a word order problem like this never arises in English.

Similar problems arise whenever an embedded clause is promoted to a main clause, which requires a change in the position of the finite verb in Dutch. In total, a word order problem occurred in 24 out 200 sentences.

Referring expressions. Referring expressions are on many occasions replaced by shorter ones – usually a little less precise. For example, *de belgische overheid* 'the Belgian authorities' is replaced by *belgie* 'Belgium'. Extreme cases of this occur where a long NP such as *deze tweede impeachment-procedure in de amerikaanse geschiedenis* 'this second impeachment-procedure in the American history' is replaced by an anaphor such as *het* 'it'.

Since a referring expression or anaphor must be appropriate in the given context, substitutions such as these transcend the domain of a single sentence and require taking the preceding textual context into account. This is especially clear in examples such as (3) in which 'many of them' is replaced by the 'refugees'.

(3) **Aut** *velen van hen worden door de serviërs in volgeladen treinen*
many of them are by the Serbs in crammed trains
gedeporteerd
deported
"Many of them are deported by Serbs in overcrowded trains."

Sub *vluchtelingen worden per trein gedeporteerd*
refugees are by train deported

It is questionable whether these types of substitutions belong to the task of sentence compression. We prefer to regard rewriting of referring expressions as

one of the additional tasks in automatic subtitling, apart from compression. As expected the challenge of generating appropriate referring expressions is also relevant for automatic subtitling.

Paraphrasing. Apart from the reduced referring expressions, there are nominal paraphrases reducing noun phrases such as *medewerkers van banken* 'employees of banks' to compound words such as *bankmedewerkers* 'bank-employees'. Likewise, there are adverbial paraphrases such as *sinds een paar jaar* 'since a few years' to *tegenwoordig* 'nowadays', and *van de afgelopen tijd* 'of the past time' to *recent* 'recent'. However, the majority of the paraphrasing concerns verbs as in the three examples below.

(4) **Aut** *X zijn doorgegaan met hun stakingen*
X are continued with their strikes
"X continued their strikes"
Sub *X staakten*
X striked

(5) **Aut** *X neemt het initiatief tot oprichting van Y*
X takes the initiative to founding of Y
Sub *X zet Y op*
X sets Y up

(6) **Aut** *X om zijn uitlevering vroeg maar Y die weigerde*
X for his extradition asked but Y that refused
Sub *Y hem niet wilde uitleveren aan X*
Y him not wanted extradite to Y
"Y refused to extradite him to Y"

Even though not all paraphrases are actually shorter, it seems that at least some of them boost compression beyond what can be accomplished with only word deletion. In the next section, we look at the possibilities of automatic extraction of such paraphrases.

3.4 Semantic Relations between Aligned Phrases

The aligned phrases in our corpus were also manually labeled according to a set of five different semantic similarity relations. By way of example, we use the following pair of Dutch sentences:

(7) a. *Dagelijks koffie vermindert risico op Alzheimer en Dementie.*
Daily coffee diminishes risk on Alzheimer and Dementia
b. *Drie koppen koffie per dag reduceert kans op Parkinson en Dementie.*
Three cups coffee a day reduces chance on Parkinson and Dementia

The corresponding syntax trees and their (partial) alignment are shown in Figure 2. It should be noted that for expository reasons the alignment shown in

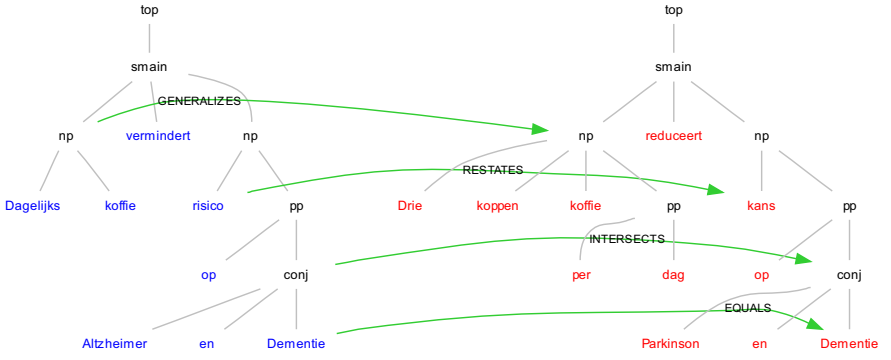

Fig. 2. Example of two (partially) aligned syntactic trees

the figure is not exhaustive. We distinguish the following five mutually exclusive similarity relations:

1. v **equals** v' iff $\textsc{str}(v)$ and $\textsc{str}(v')$ are literally identical (abstracting from case). Example: "Dementie" equals "Dementie";
2. v **restates** v' iff $\textsc{str}(v)$ is a paraphrase of $\textsc{str}(v')$ (same information content but different wording). Example: "risico" restates "kans";
3. v **generalizes** v' iff $\textsc{str}(v)$ is more general than $\textsc{str}(v')$. Example: "dagelijks koffie" generalizes "drie koppen koffie per dag";
4. v **specifies** v' iff $\textsc{str}(v)$ is more specific than $\textsc{str}(v')$. Example: "drie koppen koffie per dag" specifies "dagelijks koffie";
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: "Alzheimer en Dementie" intersects "Parkinson en Dementie"

The distribution of the semantic relations in our autocue-subtitle corpus is shown in Table 5. The bulk of the alignments concerns Equals (67%). As is to be expected, the next most frequent class is Specifies (14%), where the information in the autocue is more specific than that in the compressed subtitle, followed by Restates (11%), where information is paraphrased. Only a small percentage

Table 5. Distribution of semantic relations between aligned phrases

	#Alignments:	%Alignment:
Equals	91609	67.46
Restates	15583	11.48
Generalizes	3506	2.58
Specifies	19171	14.12
Intersects	5929	4.37

of Generalizes (3%) and Intersects (4%) relations are present. These numbers confirm our intuition that paraphrasing and generalization are the most frequent operations in sentence compression.

4 Perspectives for Automatic Paraphrase Extraction

There is a growing amount of work on automatic extraction of paraphrases from text corpora [1,10,15,21]. One general prerequisite for learning a particular paraphrase pattern is that it must occur in the text corpus with a sufficiently high frequency, otherwise the chances of learning the pattern are proportionally small. In this section, we investigate to what extent the paraphrases encountered in our random sample of 200 pairs (cf. Section 3.2) can be retrieved from a reasonably large text corpus.

In a first step, we manually extracted all 106 paraphrase patterns observed in out data set. We filtered these patterns and excluded anaphoric expressions, general verb alternation patterns such as active/passive and continuous/non-continuous, as well as verbal patterns involving more than two arguments. After this filtering step, 59 pairs of paraphrases remained, including the examples shown in the preceding section.

The aim was to estimate how big our corpus has to be to cover the majority of these paraphrase pairs. We started with counting for each of the paraphrase pairs in our sample how often they occur in a corpus of Dutch news texts, the Twente

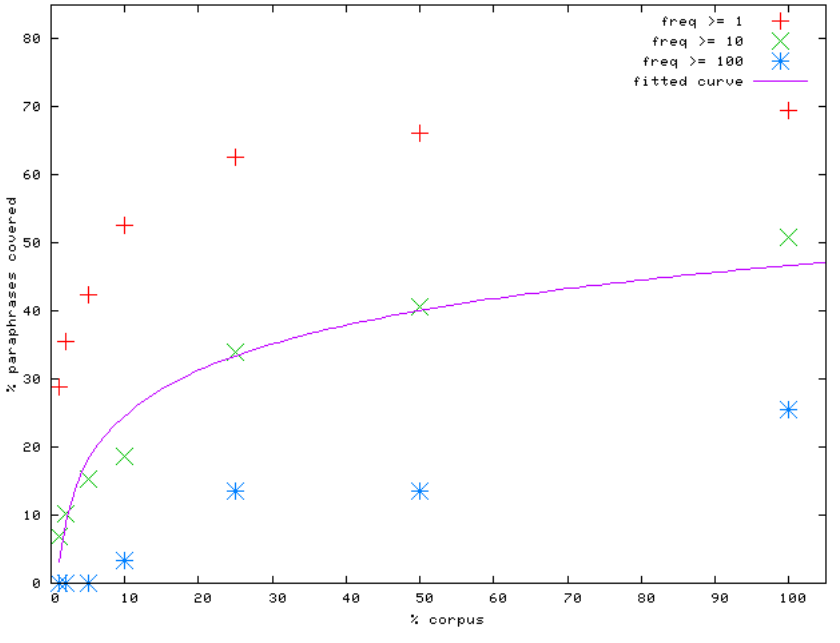

Fig. 3. Percentage of covered paraphrases as a function of the corpus size

News Corpus[6], which contains approximately 325M tokens and 20M sentences. We employed regular expressions to count the number of paraphrase pattern matches. The corpus turned out to contain 70% percent of all paraphrase pairs (i.e. both patterns in the pair occur at least once). We also counted how many pairs have frequencies of at least 10 or 100. To study the effect of corpus size on the percentage of covered paraphrases, we performed these counts on 1, 2, 5, 10, 25, 50 and 100% of the corpus. Figure 3 shows the percentage of covered paraphrases dependent on the corpus size. The most strict threshold that only counts pairs that occur at least 100 times in our corpus, does not retrieve any counts on 1% of the corpus (3M words). At 10% of the corpus size only 4% of the paraphrases is found, and on the full data set 25% of the pairs is found.

For 51% percent of the patterns we find substantial evidence (a frequency of at least 10) in our corpus of 325M tokens. We fitted a curve through our data points, and found a logarithmic line fit with adjusted R^2 value of .943 (which provides a measure between one and zero of how well future outcomes are likely to be predicted by the model). This suggests that in order to get 75% of the patterns, we would need a corpus that is 18 times bigger than our current one, which amounts to roughly 6 billion words. Although this seems like a lot of text, using the WWW as our corpus would easily give us these numbers. Today's estimate of the Index Dutch World Wide Web is 439 million pages[7]. If we assume that each page contains at least 100 tokens on average, this implies a corpus size of 43 billion tokens.

We are aware of the fact that these are very rough estimates and that the estimation method is to some extent questionable. For example, the extrapolation assumes that the relative distribution of words remains the same for a certain corpus and a superset of that corpus. Nevertheless we think that these estimates support the intuition that significantly more data is needed in order to extract the required paraphrases.

Not also that the patterns used here are word-based and in many cases express a particular verb tense or verb form (e.g. 3rd person singular) and word order. This implies that our estimations are the minimum number of matches one can find. For more abstract matching, we would need syntactically parsed data [21]. We expect that this would also positively affect the coverage.

5 Exploring Sentence Compression for Dutch

The preceding analysis of sentence compression in the context of subtitling provides evidence that deletion models are not sufficient for sentence compression. More elaborate models involving reordering and paraphrasing are therefore required. In recent work [7], Cohn & Lapata acknowledge the limitations of the deletion model and propose an interesting alternative which goes beyond word deletion. In this section, we describe a first attempt to apply this more powerful model to our Dutch subtitle data.

[6] `http://www.vf.utwente.nl/~druid/TwNC/TwNC-main.html`

[7] `http://www.worldwidewebsize.com/index.php?lang=NL`, as measured December 2009.

5.1 Sentence Compression as Tree Transduction

Cohn & Lapata [7] regard sentence compression as a tree-to-tree transduction process in which a source parse tree is transformed to a compressed parse tree. The formalization is based on Synchronous Tree Substitution Grammars (STSG) as proposed by [11], a formalism that allows local distortions of the tree structure and can therefore accommodate for substitutions, insertions and reordering. We refrain from a detailed formal description of their model, which can be found in [7], and instead provide an informal description by means of an (artificial) example.

The grammar rules of a STSG define aligned pairs of source and target tree fragments. For example, Rule 1 expresses the observation that the source NP *een nieuwe vakbond* 'a new union' can be rewritten to the compressed NP *een vakbond* 'a union' by deleting the adjective *nieuwe*.

```
Rule 1:

<NP, NP> ==> < (NP (Det een) (A nieuwe) (N vakbond)),
               (NP (Det) (N vakbond)) >
```

The tree fragments can be of arbitrary depth, allowing for an extended specification of syntactic context. Rule 2, for example, shows an example of substitution in which the source tree *medewerkers van banken* 'employees of banks' has a depth of four levels.

```
Rule 2:

<NP, NP> ==> < (NP (N medewerkers) (PP (P van) (NP (N banken)))),
               (NP (N bankmedewerkers)) >
```

To allow generalization, rules can contain variables. For instance, rule (3) has two NP slots, where the numeric indices define the alignment between the slots in the source and target tree (cf. example (5) for a gloss).

```
Rule 3:

<S, S> ==> < (S (NP [1]) (V nemen) (VP (NP (Det het)
             (N initiatief) (NP (P voor) (NP [2])))))),
             (S (NP [1]) (VP (V zetten) (NP [2]) (V_part op))) >
```

The variables are also the points of recursion in the transductive process. As in a normal context-free grammar, a source tree can be reproduced by top-down application of the left part of the synchronous grammar rules. Thus the source tree in the example below can be reproduced by first applying (the left part of) rule 3, followed by application of rules 1 and 2 to expand the two NP variables. Because of the aligned right part in the synchronous grammar rules, synchronous application of the aligned right parts produces the corresponding compressed tree.

```
Source tree:

(S  (NP (N medewerkers) (PP (P van) (NP (N banken)))) (V nemen)
(VP (NP (Det het) (N initiatief) (NP (P voor) (NP (Det een)
(A nieuwe) (N vakbond)) ))))

Target tree:

(S (NP (N bankmedewerkers)) (VP (V zetten) (NP (Det een)
(N vakbond) (V_part op)))
```

In order to automatically obtain a grammar, Cohn & Lapata rely on a parallel corpus of source and compressed sentences which are (automatically) aligned at the word level. From these word alignments, compatible constituent alignments are inferred. The resulting tree alignments are then input to an algorithm that derives the synchronous grammar rules.

Given an input source tree and a synchronous grammar, sentence compression amounts to finding the optimal target tree licensed by the grammar. One of the factors in scoring the output is an n-gram language model. Cohn & Lapata describe algorithms for training and decoding in this setting based on discriminative learning within a large margin framework.

5.2 Application to Dutch

Cohn & Lapata report state-of-the-art results for English, albeit on sentence compression by deletion only rather than the more general case which includes reordering, substitutions and insertions [7]. As they generously made their implementation publicly available as the Tree Transducer Toolkit [8] (T3), this paves the way to application to other corpora and languages. We think it is interesting to test their model in a different domain and for a new language. In the remainder of this section, we describe our experiences with a first attempt to apply this approach to Dutch.

The Dutch corpus has both advantages and disadvantages. An advantage is that it includes reordering, substitutions and insertions, and is therefore better suited to fully testing the claimed expressiveness of the model than the compression-by-deletion corpora used in [7]. A disadvantage is that it contains many sentences with a slightly ungrammatical word order. In order to understand the reason for this, it is necessary to know that the internal representation of a syntactic parse as used by the Alpino parser for Dutch is a dependency *graph* rather than a constituent tree, and may therefore contain crossing edges. Although the parser can output parse trees – which is what we use – crossing edges can only be resolved at the cost of moving some terminal nodes around to a different position in the tree. As a consequence, the yield of the parse tree, i.e., the sequence of terminals from left to right, is not always identical to the original input sentence. Hence the word order in the tree may differ slightly from that in the input sentence, and is in

[8] http://www.dcs.shef.ac.uk/~tcohn/t3/

fact often ungrammatical. A typical example of the difference between an input sentence and the yield of its parse tree is given in (8).

(8) a. *Met een flinke wind erbij kan de sneeuw zich ophopen*
with a strong wind included can the snow itself pile
"In combination with a strong wind, snow can pile up"
b. kan de sneeuw Met een flinke wind erbij zich ophopen

What is needed is a reverse step which converts the constituent tree back to a dependency graph, thereby restoring the grammatical word order. This task of word (re)ordering is addressed in for example [13] and [29] (this volume), but is currently missing in our experimental setup.

We used the corpus of Dutch autocue and subtitle sentences as described in Section 2. We divided it into 3865 sentence pairs for training and 1354 sentence pairs for testing, in such way that all test material comes from another month of news broadcasts than the training material. Syntax trees were converted to the labeled bracket format as used in the Penn Tree Bank. Word alignments were obtained by removing alignments involving non-terminal nodes from our manual tree alignments. A trigram language model was trained on a background corpus of over 566M tokens of Dutch news text from the Twente News Corpus [25]. In all subsequent steps, we used settings identical to those in the sample script provided with the T3 distribution. Grammar rules were harvested from the training data, amounting to a total of over 124k unique rules.[9] Features were derived, the model was trained, and subsequently applied to decode the test material.

Given the preliminary nature of this exercise, we performed no formal evaluation experiment. Instead we first discuss a number of issues specific to our Dutch data set that we encountered upon inspecting the output. We also came across some general problems with the approach, which will be discussed in the next subsection.

Overall it seems that most acceptable compressions are the result of only deletion. Even though we did not inspect all 1354 test sentences, we were unable to find clear examples in which a combination of reordering, substitution or insertion resulted in a shorter paraphrase.

The most obvious problem in the output is ungrammatical word order. However, this can to a large extent be blamed on the training material containing ungrammatical word order, as explained above. It is therefore to be expected that the word order in the compressed output is also somewhat distorted.[10]

Apart from word order issues, two other problems – as far as grammaticality is concerned – are grammatical incongruence and missing complements. In

[9] This includes *epsilon rules* and *copy rules*, but no *deletion rules*, because a *head table* for Dutch, which specifies the head for each syntactic phrase, was not readily available. These rules will be explained in Section 5.3.

[10] This word order issue may also affect the n-gram language model, which plays a part in the scoring function for the target tree. The n-gram model is trained on Dutch sentences with a proper word order, whereas the yield of the source tree to be scored may have a distorted word order.

addition to subject-verb agreement, Dutch has determiner-noun and adjective-noun agreement. It turns out that these agreement constraints are often violated in the compressed output, presumably because the n-gram model has a limited capability for capturing these phenomena. Likewise, obligatory arguments, usually verbal complements, are frequently missing. A related issue concerns wrong/missing functions words such as determiners, complementizers, prepositions or verbal particles. Note that that this grammatical information is in principle available, as the edges of the parse tree are labeled with dependency relations such as *su* (subject) and *obj1* (verbal/prepositional complement) and *predc* (predicative complement). If we can force the model to take this dependency information into account – perhaps simply by concatenating constituent and dependency labels into a specialized label – this may have a positive impact on the grammaticality of the output

The remaining problems have to do with content selection where deletions or substitutions radically change the meaning of the compressed sentence or even render it nonsensical. One frequent error is the reduction of a source sentence to just its subject or object NP.

5.3 Some General Issues

We encountered a number of issues which we feel are general drawbacks of the model. The first of these is that it has a tendency to insert ungrounded lexical material. That is, the compressed output contains information that was in no form present in the source sentence. In (9), for example, the NP *de grootste in z'n soort in de wereld* is replaced by the completely unrelated phrase *Macedonie*. Likewise, (10) contains an extra piece of information in the form of the PP *van het concertgebouw*, for which there is no ground in the input.

(9) a. *De high tech campus wordt de grootste in z'n soort in de wereld*
The high tech campus becomes the biggest in its sort in the world

b. *De high tech campus wordt Macedonie*
The high tech campus becomes Macedonia

(10) a. *Tot vorige week werkte ingenieur De Kwaadsteniet hier aan mee*
until last week worked engineer De Kwaadsteniet here on with
Until last week, engineer De Kwaadsteniet agreed to this

b. *van vorige week werkte De Kwaadsteniet ingenieur van het Concertgebouworkest hier aan mee*
from last week worked De Kwaadsteniet engineer of the Concertgebouw-orchestra here on with

The same problem is also observed in [24, p. 396] using automatically aligned comparable text as training material. One possible explanation is that the train-

ing material contains examples in which the source and compressed sentences are only *partly* overlapping in meaning. This would occur when multiple autocue sentences are compressed into a single subtitle sentence, so the subtitle may contain content that is not present in a least one of the autocue sentences. However, recall from Section 2 that we disregarded all cases of one-to-many and many-to-one sentence alignment. Just to check, we also ran an experiment in which we included these non-uniquely aligned sentences in the training material, and found that that indeed insertion of ungrounded material became a major problem, often giving rise to nonsensical output.

The second general problem is related to the compression ratio: it cannot be directly adapted to fit the level of compression desired by the user. As mentioned in [7], it can currently only be indirectly changed by modifying the loss function of the model. However, this means that different models must be trained for a discrete range of compression ratios, which seems impractical for real applications.

The third and final general problem has to do with coverage. The grammar rules induced from the development data are highly unlikely to cover all the lexical and syntactic variations in *unseen* data. This means that a source tree cannot be produced with the given rules, and consequently the transduction process will fail, resulting in no output. This problem seems unavoidable, even when training on massive amounts of data. The solution proposed by Cohn & Lapata is to add back-off rules. There are *epsilon rules* which delete unaligned constituents in the subtree, *deletion rules* which delete one or more non-head child nodes, and finally *copy rules* which simply copy a node and its child nodes from the source to the target tree (resulting in zero compression). However, the important thing to notice is that in the experiments reported, both here and in [6,7], it is implicit that these rules are derived not only from the development data, but also from the test data. While this is arguably a fair methodology in an experimental setting, it is problematic from a practical point of view. It means that for each unseen input sentence, we have to derive the epsilon, deletion and copy rules from the corresponding source tree, add them to the rule base, and retrain the model. Since training the support vector machine underlying the model takes considerable computing time and resources, this is clearly prohibitive in the case of online application.

To illustrate the point, we repeated our experiment for Dutch, with the difference that this time epsilon and copy rules were derived from the training data only, excluding the test data. In this setting, just 94 out of the total of 1354 test sentences (less than 7%) result in an output sentence.

To sum up, exploring sentence compression for Dutch with the tree transducer model from [7] gave results which are not immediately encouraging. However, these are preliminary results, and tuning of input and parameters may lead to significant improvements.

6 Discussion

In the first part of this chapter we performed an in depth analysis of sentence compression as observed in the context of subtitling for Dutch. We found that

only 16.11% of 5233 subtitle sentences were proper subsequences of the corresponding autocue sentence, and therefore 84% can not be accounted for by a deletion model. One conclusion appears to be that the subsequence constraint greatly reduces the amount of available training material for any word deletion model. However, an attempt to rewrite non-subsequences to semantically equivalent sequences with the same CR suggests that a deletion model could in principle be adequate for 55% of the data. Moreover, in those cases where an application can tolerate a little slack in the CR, a deletion model might be sufficient. For instance, if we are willing to tolerate up to two more tokens, we can account for as much as 169 (84%) of the 200 non-subsequences in our sample, which amounts to 87% (16% plus 84% of 84%) of the total data.

It should be noted that we have been very strict regarding what counts as a semantically equivalent subtitle: every piece of information occurring in the non-subsequence subtitle must reoccur in the sequence subtitle.[11]. However, looking at our original data, it is clear that considerable liberty is taken as far as conserving semantic content is concerned: subtitles often drop substantial pieces of information. If we relax the notion of semantic equivalence a little, an even larger part of the non-subsequences can be rewritten as proper sequences.

The remaining problematic non-subsequences are those in which insertion, substitution and/or word reordering are essential to obtain a sufficient CR. One of the issues we identified is that deletion of certain constituents must be accompanied by a change in word order to prevent an ungrammatical sentence. Since changes in word order appear to require grammatical modeling or knowledge, this brings sentence compression closer to being an NLG task.

Nguyen and Horiguchi [19] describe an extension of the decision tree-based compression model [18] which allows for word order changes. The key to their approach is that dropped constituents are temporarily stored on a *deletion stack*, from which they can later be re-inserted in the tree where required. Although this provides an unlimited freedom for rearranging constituents, it also complicates the task of learning the parsing steps, which might explain why their evaluation results show marginal improvements at best.

In our data, most of the word order changes appear to be minor though, often only moving the verb to second position after deleting a constituent in the topic position. We believe that unrestricted word order changes are perhaps not necessary and that the vast majority of the word order problems can be solved by a fairly restricted way of reordering, in particular, a parser-based model with an additional swap operation that swaps the two topmost items on the stack. We expect that this is more feasible as a learning task than an unrestricted model with a deletion stack.

Apart from reordering, other problems for word deletion models are the insertions and substitutions as a result of paraphrasing. Within a decision tree-based model, paraphrasing of words or continuous phrases may be modeled by a combination of a paraphrase lexicon and an extra operation which replaces the n

[11] As far as reasonably possible, because in a few cases the substitute contains extra information that is simply not present in the autocue.

topmost elements on the stack by the corresponding paraphrase. However, paraphrases involving variable arguments, as typical for verbal paraphrases, cannot be accounted for in this way. More powerful compression models may draw on existing NLG methods for text revision [16] to accommodate full paraphrasing.

We also looked at the perspectives for automatic paraphrase extraction from large text corpora. About a quarter of the required paraphrase patterns was found at least a hundred times in our corpus of 325M tokens. Extrapolation suggests that using the web at its current size would give us a coverage of approximately ten counts for three quarters of the paraphrases.

In the second part of this chapter we explored sentence compression with the tree transducer model as proposed by Cohn & Lapata [7]. We reported preliminary results of applying this more powerful compression model to the task of subtitle compression for Dutch. In theory the proposed model looks very promising, because it can handle and learn reordering, substitution and insertion in an elegant way. In practice, the results were not immediately encouraging. We identified a number of problems with the model, both when applied to Dutch and in general. We might interpret these findings as support for choosing a hybrid approach to sentence compression – explicitly modeling linguistic knowledge – rather than a fully data-driven approach, at least if the goal is to model more complicated forms of compression beyond deletion.

Incidentally, we identified two other tasks in automatic subtitling which are closely related to NLG. First, splitting and merging of sentences [17], which seems related to content planning and aggregation. Second, generation of a shorter referring expression or an anaphoric expression, which is currently one of the main themes in data-driven NLG [14].

In conclusion, we have presented evidence that deletion models for sentence compression are not sufficient, at least not as far as concrete application in subtitle compression is concerned. More elaborate models involving reordering and paraphrasing are therefore required, which puts sentence compression in the field of NLG.

Acknowledgments. We would like to thank Nienke Eckhardt, Paul van Pelt, Hanneke Schoormans and Jurry de Vos for the corpus annotation work, and Erik Tjong Kim Sang and colleagues for the autocue-subtitle material from the ATRANOS project, Martijn Goudbeek for help with curve fitting, and Peter Berck for text material to train the Dutch n-gram model. This work was conducted within the DAESO project funded by the STEVIN program (De Nederlandse Taalunie).

References

1. Barzilay, R., Lee, L.: Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pp. 16–23 (2003)

2. Belz, A., Reiter, E.: Comparing automatic and human evaluation of NLG systems. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, pp. 313–320 (2006)
3. Bouma, G., van Noord, G., Malouf, R.: Alpino: Wide-coverage computational analysis of Dutch. In: Daelemans, W., Sima'an, K., Veenstra, J., Zavre, J., et al. (eds.) Computational Linguistics in the Netherlands 2000. Selected Papers from the Eleventh CLIN Meeting, Rodopi, Amsterdam, New York, pp. 45–59 (2001)
4. Clarke, J., Lapata, M.: Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Morristown, NJ, USA, pp. 377–384 (2006)
5. Clarke, J., Lapata, M.: Global inference for sentence compression an integer linear programming approach. Journal of Artificial Intelligence Research 31, 399–429 (2008)
6. Cohn, T., Lapata, M.: Sentence compression beyond word deletion. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 137–144. Association for Computational Linguistics (2008)
7. Cohn, T., Lapata, M.: Sentence compression as tree transduction. J. Artif. Int. Res. 34(1), 637–674 (2009)
8. Corston-Oliver, S.: Text compaction for display on very small screens. In: Proceedings of the Workshop on Automatic Summarization (WAS 2001), Pittsburgh, PA, USA, pp. 89–98 (2001)
9. Daelemans, W., Höthker, A., Tjong Kim Sang, E.: Automatic sentence simplification for subtitling in Dutch and English. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, pp. 1045–1048 (2004)
10. Dolan, B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: Proceedings of the 20th International Conference on Computational Linguistics, Morristown, NJ, USA, pp. 350–356 (2004)
11. Eisner, J.: Learning non-isomorphic tree mappings for machine translation. In: Proceedings of 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, pp. 205–208 (July 2003)
12. Filippova, K., Strube, M.: Sentence fusion via dependency graph compression. In: EMNLP 2008: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 177–185. Association for Computational Linguistics, Morristown (2008)
13. Filippova, K., Strube, M.: Tree linearization in English: improving language model based approaches. In: NAACL 2009: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 225–228. Association for Computational Linguistics, Morristown (2009) (Companion Volume: Short Papers)
14. Gatt, A., Belz, A.: Attribute selection for referring expression generation: New algorithms and evaluation methods. In: Proceedings of the Fifth International Natural Language Generation Conference, pp. 50–58. Association for Computational Linguistics, Columbus (2008)
15. Ibrahim, A., Katz, B., Lin, J.: Extracting structural paraphrases from aligned monolingual corpora. In: Proceedings of the 2nd International Workshop on Paraphrasing, Sapporo, Japan, vol. 16, pp. 57–64 (2003)

16. Inui, K., Tokunaga, T., Tanaka, H.: Text revision: A model and its implementation. In: Proceedings of the 6th International Workshop on Natural Language Generation: Aspects of Automated Natural Language Generation, pp. 215–230. Springer, London (1992)
17. Jing, H., McKeown, K.: Cut and paste based text summarization. In: Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics, San Francisco, CA, USA, pp. 178–185 (2000)
18. Knight, K., Marcu, D.: Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence 139(1), 91–107 (2002)
19. Le, N.M., Horiguchi, S.: A new sentence reduction based on decision tree model. In: Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation, pp. 290–297 (2003)
20. Lin, C.Y.: Improving summarization performance by sentence compression - A pilot study. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, vol. 2003, pp. 1–9 (2003)
21. Lin, D., Pantel, P.: Discovery of inference rules for question answering. Natural Language Engineering 7(4), 343–360 (2001)
22. Marsi, E., Krahmer, E.: Annotating a parallel monolingual treebank with semantic similarity relations. In: Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories, Bergen, Norway, pp. 85–96 (2007)
23. Marsi, E., Krahmer, E.: Detecting semantic overlap: A parallel monolingual treebank for Dutch. In: Verberne, S., van Halteren, H., Coppen, P.A. (eds.) Computational Linguistics in the Netherlands (CLIN 2007): Selected papers from the 18th meeting, Rodopi, Amsterdam, pp. 69–84 (2008)
24. Nomoto, T.: A Comparison of Model Free versus Model Intensive Approaches to Sentence Compression. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 391–399 (2009)
25. Ordelman, R., de Jong, F., van Hessen, A., Hondorp, H.: Twnc: a multifaceted Dutch news corpus. ELRA Newsletter 12(3/4), 4–7 (2007)
26. Turner, J., Charniak, E.: Supervised and unsupervised learning for sentence compression. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, pp. 290–297 (June 2005)
27. Vandeghinste, V., Pan, Y.: Sentence compression for automated subtitling: A hybrid approach. In: Proceedings of the ACL Workshop on Text Summarization, pp. 89–95 (2004)
28. Vandeghinste, V., Tjong Kim Sang, E.: Using a Parallel Transcript/Subtitle Corpus for Sentence Compression. In: Proceedings of LREC 2004 (2004)
29. Wan, S., Dras, M., Dale, R., Paris, C.: Spanning tree approaches for statistical sentence generation. In: Krahmer, E., Theune, M. (eds.) Empirical Methods in NLG. LNCS (LNAI), vol. 5790, pp. 13–44. Springer, Heidelberg (2010)
30. Zajic, D., Dorr, B.J., Lin, J., Schwartz, R.: Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. Information Processing Management 43(6), 1549–1570 (2007)