Reduction of Dutch Sentences for Automatic Subtitling

Erik F. Tjong Kim Sang, Walter Daelemans and Anja Höthker

CNTS–Language Technology Group, University of Antwerp, Belgium

Abstract

We compare machine learning approaches for sentence length reduction for automatic generation of subtitles for deaf and hearing-impaired people with a method which relies on hand-crafted deletion rules. We describe building the necessary resources for this task: a parallel corpus of examples of news broadcasts of the Flemish VRT broadcasting corporation, and a Dutch shallow parser based on the material of the Spoken Dutch Corpus (CGN). We evaluate the sentence simplifiers and discuss their performance.

1 Introduction

The goal of the Flemish research project ATraNoS (Automatic Transcription and Normalization of Speech) is to contribute to the development of better products for automatic transcription of speech. As a case study, we have chosen automatic generation of TV subtitles for deaf and hearing-impaired people. This task requires both speech recognition for converting audio signals to text, and sentence reduction for decreasing sentences to lengths which enable them to fit in the available subtitle space. This paper describes the sentence reduction part of the project.

We have defined sentence simplification as a classification task which can be performed by a machine learning system. We needed examples in order to be able to train the learner and therefore we have built a parallel corpus of transcribed news broadcasts and the associated subtitles, provided by two public broadcasting companies: VRT (Belgium) and NOS (The Netherlands). Transcript and subtitle sentences in the corpus have been aligned automatically and all alignments have subsequently been checked manually.

We have generated a linguistic analysis for all sentences in the corpus. This analysis is helpful for the learner for understanding the relation between the transcripts and the subtitles. The linguistic analysis also enables the learner to deal with unknown words. Rather than performing a full analysis of the sentences we have chosen for a shallow analysis because it is more robust and because we expect that such an analysis will be sufficient for the majority of the sentence reduction actions. Our shallow linguistic analysis produces part-of-speech tags, lemmas, text chunks (sequences of syntactically related words) and phrase relations like the subject-verb relation.

As an alternative to the machine learning approach, we have also explored performing sentence reduction with hand-crafted phrase deletion rules. The rules have been developed on different data but they have been tested on the same data as the machine learner. Compiling the rules has enabled us to get a better understanding of the problem of sentence compression. It also made it possible to define the problems of the machine learning approach more precisely and this made it easier to tackle these problems.

		transcriptions		subtitles		compression
corpus	files	sentences	words	sentences	words	rate
VRT	101	48,871	579,985	50,141	431,190	78.0%
NOS	125	17,393	225,603	26,830	230,295	73.3%
Thuis	7	4,145	24,853	3,963	20,387	85.7%

Table 1: An overview of the sizes of the three parts of the parallel ATraNoS corpus. The corpus contains two news broadcast parts (VRT and NOS) and a soap (Thuis). Average character compression rates (percentages of remaining characters) have been computed for alignment structures which included both transcribed text and subtitle text.

The next three sections describe building the parallel corpus of examples, constructing the Dutch shallow parser and our sentence simplification work. The final section contains some concluding remarks.

2 Building a parallel subtitle corpus for Dutch

We will use machine learning for performing sentence simplification and for this approach we need a corpus of examples. This section describes our parallel Dutch subtitle corpus. It deals with data collection and sentence alignment.

2.1 Data collection

The parallel corpus contains three sections. The first one consists of broadcasts of the daily 19:00 news edition of the Flemish broadcasting corporation VRT. We have obtained HTML files with transcripts of the programmmes between December 16, 2001 and March 31, 2002 from the broadcasting company. The associated teletext subtitles were collected at our own site with special teletext hardware.

The second part of the corpus consists of broadcasts of the daily 20:00 news edition of the Dutch broadcasting corporation NOS. We have obtained SGML files with autocues and subtitles of the programmes between January 1, 1999 and December 31 of the same year. These files were provided to us by the University of Twente in The Netherlands who have used them in DRUID project[1].

The third part of the corpus contains transcripts and subtitles of the VRT soap *Thuis*. From the broadcasting company we have obtained a sample of five pairs of transcripts and the associated subtitles. Another two programmes were transcribed at our site while the subtitles for these programmes were collected with our teletext hardware.

All files in the corpus have been converted to SGML with explicit sentence boundary markers. An overview of the size of the different parts of the corpus can be found in Table 1.

[1] http://dis.tpd.tno.nl/druid/

```
repeat three times
    for each transcribed sentence
        for each subtitle sentence within a fixed window
            check if subtitle is a better match than current best
        if (similarity value best $>$ minimal similarity threshold) create link
    remove crossing links
    decrease current similarity threshold with 50%
for each transcribed sentence
    find other neighboring subtitle sentences that match
```

Table 2: High-level pseudo-code of the sentence alignment algorithm

2.2 Sentence alignment

In order to create a usable parallel corpus, we need to create links between the sentences in the transcript files and the associated sentences in the subtitle files. This is not an easy task since in any of the files some sentences might have been omitted. In the NOS part of the corpus we do not have complete transcripts of the programmes but autocues, which only contain the text of the news anchor, the person that reads the news. Interviews or background comments of other people are included in the subtitles but not in the autocues. By definition, the subtitle files lack some of the sentences present in the transcripts because due to the lack of available space complete sentences might need to be deleted.

We have developed an alignment method which takes into consideration the contents of the sentences. It estimates the probability that two sentences are counterparts by the number of characters in words that they share divided by the length of the shortest of the two sentences. The value that results from this computation is called the similarity value of the two sentences. The algorithm contains four stages (Table 2). In the first three stages it attempts to find the best subtitle sentence for each transcribed sentence. A link between two sentences will only be made if their similarity value is larger than a threshold value, the minimal similarity value. After each stage this threshold value decreases. The algorithm does not allow crossing links and therefore the set of candidate sentences will be limited by links created in previous stages. The first three stages only allow a sentence to be linked to exactly one other sentence (1→1 alignments). In the fourth stage the algorithm searches for isolated sentences and combines these in larger alignment structures when that is appropriate (n→m alignments).

This algorithm has two parameters: the minimal size of the similarity value which is acceptable for making links (t) and the size of window in which candidate sentences will be searched measured in words (w). We have estimated the optimal values of the two parameters by applying the algorithm with different parameter settings to the VRT news broadcasts of December 2001 and January 2002. The performance was measured in $F_{\beta=1}$, the harmonic mean of the precision and recall figures computed for sentence pairs (Van Rijsbergen 1975). Based on these tests we have chosen initial threshold t=0.62 and window size w=70 words as the default values of the parameters. With these parameters, the alignment software obtains a precision score of 90.9% and a recall score of 91.7% on the VRT part of the corpus. In order to assure the quality of the corpus, all the proposed sentence alignments have been checked manually.

Apart from this alignment method we have also tested two alternatives. The sentence length based method of Gale and Church (1993) did not work very well for our data because of the large number of missing sentences. The Gale-Church alignment method does not deal well with sentences which do not have a counterpart in the other part of the corpus. We have also experimented with replacing the word matches in our approach with 4-gram matches like in the character alignment method described in Church (1993). This approach did not achieve the same performance levels as the word-matching technique we have used in our alignment method.

3 Shallow parsing of Dutch

A linguistic analysis of the source and target sentences in the example corpus is very useful for learning sentence simplification. Therefore we have developed a shallow parser for Dutch in the framework of this research project. The next sections describe the data and learning methods which we have used for this, the linguistic tasks which we have worked at and the results obtained for the different tasks.

3.1 Data and methods

Our goal is to build a modular shallow parser for Dutch with machine learning components. For this purpose, we need data for training the machine learners. We have examined two candidates. The first is the Corpus Gesproken Nederlands (CGN)[2], a 10-million word corpus of spoken Dutch. All words in the corpus are annotated with part-of-speech tags and about 10% of the sentences are annotated with syntactic trees.

The fully annotated part of the CGN corpus contains the linguistic information we need for building a shallow parser. However, the corpus annotation format has two features which may cause problems. First, the trees contain crossing brackets which means that phrases can be discontinuous. This will make it harder to identify phrase boundaries automatically, given the nature of Dutch, this problem is difficult to avoid. The second problematic feature is that trees in the corpus are relatively flat. Some expected phrase annotations are missing and this will make automatic phrase recognition difficult as well.

The second annotated corpus which we looked at is the Alpino Treebank[3], a corpus of about 140,000 words of linguistically annotated newspaper text. The annotation format of the corpus is similar to the one of the CGN corpus and this

[2] http://lands.let.kun.nl/cgn/

[3] http://www.let.rug.nl/~vannoord/trees/

means that the two problems we mentioned for the CGN corpus apply to this corpus as well. Additionally it should be noted that the part-of-speech classes in the Alpino Treebank are not as specific as those in the CGN corpus. We have chosen the CGN corpus as training material for our linguistic modules, primarily because it contains more material.

We chose the memory-based tagger MBT as machine learner (Daelemans, Zavrel, Van der Sloot and Van den Bosch 2003a). MBT performs well for part-of-speech tagging (Van Halteren, Zavrel and Daelemans 2001). For more complex tasks which require using information from different sources, it is probably not the optimal choice. However, MBT is easy to use and it enabled us to generate reasonable modules for different linguistic analysis tasks quickly. This tagger stores all training data and classifies test data by selecting the classification of the closest training data item. We have used the default settings of the learner: the nearest-neighbor algorithm (IB1) with the gain ratio variant of information gain weighting of features combined with the overlap distance metric (see Daelemans, Zavrel, Van der Sloot and Van den Bosch (2003b) for background information).

3.2 Linguistic tasks

We have built linguistic analysis modules for four tasks. The first is part-of-speech tagging: assigning word classes to words. Here is an example analysis:

> word/WW(pv,tgw,ev) je/VNW(pers,pron,nomin,red,2v,ev)
> hier/VNW(aanw,adv-pron,obl,vol,3o,getal) nou/BW()
> wakker/ADJ(vrij,basis,zonder) van/VZ(fin) ?/LET()

The part-of-speech classes of the CGN corpus are rich. Apart from defining that a word is a pronoun (VNW), a verb (WW) or something else, a part-of-speech tag contains several other features of the word.

The second task is lemmatization: finding the base form of words. Here is an example:

> basketballer/basketballer Dennis/Dennis Rodman/Rodman
> heeft/hebben van/van zich/zich laten/laten horen/horen ./.

The output of this module will be useful for identifying similarities between different forms of verbs, nouns and adjectives.

The third task we examined was text chunking: dividing sentences in groups of adjacent syntactically related words:

> [NP De bonden NP] [VP eisen VP] [NP meer duidelijkheid NP]
> [PP over PP] [MWU Ford Genk MWU] .

Discontinuous phrases will be labeled as two separate phrases of the same type. This task proved to be extremely hard, among others because of the missing annotation levels mentioned in section 3.1. Therefore we have required from the learner that it produced only the five most frequent and most reliable phrase types: clause

Task	Input	Parameters	Precision	Recall	$F_{\beta=1}$
POS tagging	words	dfWaw,chssswdFw	96.5%	96.5%	96.5
lemmatization	words	wdfa,cssswdF	99.0%	99.0%	99.0
text chunking	POS	wdfWw,wF	92.6%	93.0%	92.8
relation finding	POS	wddfWaa,pdFa	93.0%	92.5%	92.7

Table 3: Best performances of MBT obtained on the CGN development data on the four linguistic analysis tasks. The parameter strings show the best MBT parameters found for known words (left) and unknown words.

start markers (CLB), multi-word units (MWU), noun phrases (NP), prepositional phrases (PP) and verb phrases (VP).

The fourth and final task which we looked at was relation finding. This involves finding head verbs and their associated semantic roles like subjects and objects. Here is an example:

> [SU De bonden SU] [HD eisen HD] meer duidelijkheid
> over Ford Genk .

Although some objects have been marked up in the CGN corpus we have been unable to extract them in a consistent way. Therefore, we have only trained this module to identify head verbs (of types SMAIN, SSUB, SV1 and PPRES) and their associated subjects.

3.3 Experiments and results

The memory-based tagger MBT has several parameters which influence its performance. We did not know in advance what parameters would be best for our tasks and therefore we have experimented with different settings for context tokens (parameters a, d, f and w), prefix and suffix characters of the current word (p and s) and lexical information flags regarding the current word (c, h and n). The part-of-speech tagging and lemmatization modules take words as input but for the chunking module and the relation module we have also tested respectively part-of-speech tag input, and input from either part-of-speech tags or chunk tags. For the part-of-speech tagging and lemmatization experiments we had about 6 million words available (CGN release 0.6) of which we used 1% as development material for evaluating the different parameter settings of the tagger (file names ending in 01), 1% as final test data (file names ending in 69) and 98% as training material. For the other two modules these figures were 460,000 words in total, 10% for development (file names ending in 2), 10% for testing (file names ending in 9) and 80% for training[4].

[4] The segmentation in train, development and test sets was based on our wish to have development and test sets between 50,000 and 100,000 words. Smaller data sets produce evaluation scores that are less reliable and larger data sets require too much processing time during system development.

Training MBT on the part-of-speech task was straightforward. The lemmatization task was more problematic. In order to enable the learner to generalize, we could not use words as output classes but instead we employed morphological patterns which defined how the lemma is built from the source word. An example of this is the pattern for *toegekend*: `-toege-d+toe+Len`. In order to get the lemma we need to remove *toege* from the start of the word, remove *d* from the end of the word, add *toe* to the start of the remaining string and add the final character plus *en* to the end of the string. This results in *toekennen*. This pattern works for many different verbs. Using it as output class rather than specifying the required lemma, enables the learner to correctly analyze words that do not occur in the training data.

After restricting the chunk types to the five we mentioned in the previous section, text chunking performed reasonably. We obtained the best performance with input that consisted of part-of-speech tags rather than words. Relation finding was restricted to head verbs and their associated subjects. This module performed best with part-of-speech input as well. An overview of the performances of the four modules can be found in Table 3.

4 Sentence simplification

This section describes our sentence reduction work with the sentences from the corpus described in section 2 annotated with the shallow parser outlined in section 3. After an overview of earlier work in sentence simplification, we will explain the features and output classes used in our experiments, and describe the learning methods which we will apply to the data. After this we list the results of the experiments we have performed and discuss the evaluation of the results

4.1 Related work

There exists a large body of work on text summarization but the largest part of this concerns extraction approaches in which a subset of the sentences in a text is proposed as a summary of the text. Most of the work on sentence simplification (also called sentence reduction, sentence condensation and sentence compression) relies on an automatic syntactic analysis after which handcrafted rules, sometimes with learned probabilities, remove optional branches in the syntactic trees. For evaluation, most systems rely on human judges which examine the grammaticality and the information content of the reduced sentence although some authors also propose automatic evaluation methods which are shown to correlate with the decisions made by the judges.

Knight and Marcu (2000) compare two machine learners, a noisy channel model and decision trees, on compressing sentences by learning to map syntactic trees to other trees. After training on a corpus of about 1000 sentences, both systems performed significantly better than a baseline based on word bigram occurrences, according to a blind evaluation by human judges. Hori, Furui, Malkin, Yu and Waibel (2002) compute various word and word relation scores, and build

```
for each pair of transcriptions and subtitles
    remove punctuation and capitalization
    link every word to copies and synonyms
    remove duplicate links based on longest chains
    remove remaining crossing links
    link multi-word synonyms
    link remaining isolated words
```

Table 4: High-level pseudo-code of the word alignment algorithm

compressed spoken sentences by finding optimal sets of words with dynamic programming. They show that this approach performs better than randomly choosing words from the original sentence, according to an automatic evaluation.

Most other work on sentence condensation involves (shallow) parsing, sometimes with additional lexical and morphological tools, together with rule-based reduction, occasionally with statistical scores like tf-idf. An interesting approach to sentence compression is by paraphrasing, that is replacing phrases by shorter phrases with the same meaning. Shinyama, Sekine, Sudo and Grishman (2002) outline how paraphrases can be derived automatically from related documents. There is more work on sentence summarization than we can discuss here. For additional references, please see Daelemans, Höthker and Tjong Kim Sang (2004). The application of sentence simplification for automatically generating subtitles for TV programs is mentioned in Robert-Ribes, Pfeiffer, Ellison and Burnham (1999) but to our knowledge there has been no practical study linking the two.

4.2 Data

We will define sentence simplification as a word classification task. In comparison with a subtitle, the words from the transcribed sentence can be copied, deleted or replaced by another word. Furthermore, words that do not appear in the transcription can be inserted in the subtitle. Since the subtitles are often quite similar to the transcription, the word copy action is the most frequent action. Here is an example from the corpus:

de politici vinden de euro natuurlijk/DELETE een goeie/goede zaak

In this sentence, two words need to be changed. The adverb *natuurlijk* needs to be deleted and the adjective *goeie* needs to be replaced by *goede*. In order to find out what word actions are required to transform the transcription to the subtitles, the available sentence-level alignment is not sufficient. We need alignment at word level. Automatically retrieving word-to-word links is more difficult than obtaining sentence alignments. Words in a subtitle may occur in a different position than in the transcript and this makes it hard to match corresponding words. Generating a rough alignment and correcting this manually requires too much work.

	sentences	words	copied	deleted	replaced	CR
train	9,876	125,519	93,506	22,986	9,027	77.1%
development	1,345	15,577	11,660	2,767	1,150	77.8%
test	1,314	15,605	11,737	2,836	1,032	77.6%

Table 5: Size of the three parts of the selected news broadcast data measured in number of sentences and number of words. The final four columns show the relation between the transcriptions and the subtitles: the number of copied words (75%), the number of deleted words (18%), the number of replaced words (7%) and the average character compression rate: the percentage of remaining characters in the subtitles.

In order to get some useful material, we have performed an automatic word alignment and discarded all sentences in which the alignment could be suspected to have been unsuccessful. Word alignment started with removing all punctuation signs and converting capital characters to lower case, thus simulating the future input of automatically transcribed spoken text. This cleanup action also made word alignment easier because there were inconsistencies in the punctuation and capitalization between the transcriptions and the subtitles. An overview of the other actions performed by the word alignment algorithm can be found in Table 4.

After performing the word alignment, we have selected sentence pairs from the news corpus that contained sentences sharing at least half of their words or contained at most three different words. This restriction was enforced in order to avoid including pairs in the training data that were difficult to learn from. Pairs of copied sentences have also been excluded from the material since those were uninteresting from a sentence simplification point of view: a future practical system will only be applied for sentences for which compression is required. We have only used the VRT part of the corpus. The dialogues in the soap part are different from the news parts and we wanted to create training material with a more-or-less consistent content.

The data selection method has resulted in a collection of 12,535 sentences (156,701 words). We have divided this in a training part (9,876 sentences, 125,519 words), a development part (1,345 sentences, 15,577 words) and a test part (1,314 sentences, 15,605 words, see Table 5). Each word in the three data sets was specified by 34 features: the six features word, lemma, part-of-speech tag, chunk tag, relation tag and person name tag for the word and its four nearest neighbors as well the classes of the four nearest neighbors (estimated in an earlier run of the learner). Person name tags have been generated by a basic named entity tagger. For each sentence we have computed the compression rate: the number of characters in the subtitles divided by those in the transcripts. This compression rate will be a target for the machine learners.

4.3 Methods

We have applied three different systems to the data. The first is a baseline system which deletes words at the end of the sentence until the required compression rate is met. Additionally the system removes sentence-final articles, prepositions and conjunctions in order to assure some syntactic correctness. The second system is the memory-based learner TiMBL (Daelemans, Zavrel, Van der Sloot and Van den Bosch 2003b) and the third is a system based on hand-crafted deletion rules. In this section we will describe the second and the third approach in more detail.

Memory-based learning involves storing training data items and assigning classes to test items which correspond to training items that are similar. There are different memory-based learning algorithms and different ways for computing item similarity. We have used the default settings of TiMBL in combination with uniform feature weighting (the default feature weighting performed worse). TiMBL has an important advantage over the memory-based tagger MBT: it allows complex data items, like for example words combined with their part-of-speech tag, while MBT is restricted to simple data items, usually words *or* part-of-speech tags.

We have applied three extensions to the memory-based learner. The first is feature selection which we have implemented with bidirectional hill-climbing (Caruana and Freitag 1994). It starts with separately evaluating the learner for each individual feature and continues with evaluating with all feature pairs containing the best individual. This process is continued until adding or removing a single feature to the current set does not result in an increased performance. Feature selection is necessary because the chosen machine learner is sensitive to feature choice: it might perform better with a subset of features than with the complete set.

The second extension to the memory-based learner which we employed was classifier stacking. We want to use of the classes of the neighbor words. In order to obtain these we ran preliminary experiments and used the generated output classes as context class features in the next experiments. The third extension was bypassing the class selection process of the memory-based learner, which chooses the class with the highest score for each data item. We have only used this approach for word replacement actions: if the highest class score was related to a word replacement, then this output class was chosen. For all other data items, word deletions suggested by the learner were ranked and the highest ranked candidates were selected until the required compression rate was met or until there were no candidates left. This enabled the system to remove a word even if the training data contained more examples of copying the word than deleting it.

The sentence simplification method based on hand-crafted deletion rules also processes the data in two steps. First, it finds all words and phrases in the data that are candidates for deletion and then it selects phrases for deletion until the required compression rate is met. We have defined different deletion rules, among others for removing adverbs, adjectives, first names, interjections, prepositional phrases, phrases between commas or brackets, relative clauses, numbers and time phrases.

test data	CR	Coverage	Precision	Recall	$F_{\beta=1}$
Baseline	69.1%	100.0±0.0%	23.5%	23.2%	23.3±1.9
Rules	71.0%	88.9±1.5%	29.3%	27.4%	28.3±1.5
TiMBL	73.2%	99.7±0.3%	40.7%	40.3%	40.5±1.1

Table 6: Sentence reduction performances of a baseline system, the hand-crafted deletion rules and the memory-based learner TiMBL, measured on the test data set by the average percentage of remaining characters (compression rate: CR, target: 77.6%), the percentage of sentences for which the compression rate was met, and precision, recall and $F_{\beta=1}$ computed for word deletions and word replacements. The baseline system removes words at the end of the sentence until the required compression rate is met. The intervals in the table are 90% confidence intervals.

In order to avoid tuning the rules to the test data, they were developed based on other news text (the Dutch NOS Teletext) and a small part of the training data. As an example, we show a simplified version of the rule for adverb deletion (BW is the POS tag for adverbs, base verbs are verbs like *be* and *become*):

```
if (currentPOS == BW and not currentWord is sentenceFinal and
    not previousVerb in baseVerbs and not currentWord in negations)
then
     mark currentWord as deletion candidate
```

After identifying all candidates for deletion, a selection of these phrases will be removed. The selection process starts with removing the shortest phrases, measured in number of words. If phrases have the same length, preference is given to phrases closer to the end of the sentence. The selection process continues until the required compression rate is met or until there are no more phrases left to delete.

4.4 Experiments and results

We have performed feature selection for TiMBL while training with the available training data and testing with the development data. The word at focus position, the focus part-of-speech tag, the focus and successor chunk tag and all available named entity tags were selected as useful features for the memory-based learner but the lemma and relation information as well as the classes of the neighboring words were deemed useless. Performance was measured in $F_{\beta=1}$ rate, the harmonic mean of precision and recall, for deletion and replacement actions. The feature set corresponding with the best performances has been used for processing the test data. An overview of the performances on this data set can be found in Table 6.

Both the rules and the memory-based learner TiMBL performed significantly better than the baseline with respect to $F_{\beta=1}$ rates ($p \ll 0.01$ according to a bootstrap sampling test (Noreen 1989)). TiMBL outperformed the rules ($p \ll 0.01$)

	manual evaluation			automatic evaluation		
first 200	syntax	seman.	both	$F_{\beta=1}$	BLEU	ROUGE
Baseline	47±7%	39±6%	27±6%	22±5	51±3%	72±2%
Rules	68±7%	57±7%	49±7%	27±4	52±3%	74±2%
TiMBL	46±7%	52±7%	38±7%	36±3	56±3%	77±2%

Table 7: Evaluation of the first 200 sentences of the test data. A human annotator has performed a blind manual evaluation of the sentences with respect to syntactic correctness and semantic completeness. The third column shows how often sentences were judged to be perfect with respect to both criteria. Three automatic scoring systems have been applied to the same sets of sentences: $F_{\beta=1}$ applied to word deletions and word replacements, the n-gram precision method BLEU and the unigram recall method ROUGE. The intervals represent 90% confidence intervals.

both with respect to $F_{\beta=1}$ rates and with respect to coverage, the percentage of sentences that were compressed to the required size. The rules only perform part of the task (word deletion) while the learner carried out both deletions and replacements. However, even without taking the replacements into account, the learner was better than the rules ($F_{\beta=1}$=38.6).

4.5 Evaluation

We have chosen to evaluate the output of the systems by examining the deleted and replaced words and comparing them with a gold standard. This will not always be fair for the systems, as the following example shows:

> in afwachting komt er *nieuw* overleg *en* met *de* landbouw en *met de* milieubeschermers en in de regering
>
> in afwachting komt er overleg met landbouw en milieubeschermers en in de regering
>
> er komt overleg en met de landbouw en met de milieubeschermers en in de regering

The first sentence comes from the transcriptions, the second is the associated subtitle and the third is the output of the rule-based system. The system output is fine: it satisfies the required compression rate and it is grammatically sound. Yet, only one of the three word deletions proposed by the system occurs in the subtitle (precision: 33%) while the system suggests only one of the five deletions in the subtitle (recall: 20%; $F_{\beta=1}$: 25). According to the evaluation measure the suggested simplification is not very good.

The mismatch between true quality and automatically estimated quality can be avoided by performing a manual evaluation of the system output. This evaluation can address different issues: whether the compression rate has been met, whether simplified sentence still contains the same information content as the original one and whether the simplification process has resulted in a grammatical sentence. However, a full manual evaluation is costly and in the process of system development, when many different versions of the systems need to be evaluated, performing a manual evaluation of all of their output is infeasible.

A compromise which was recently suggested, is to use automatic scoring systems which are assumed to have a high correlation with manual evaluation. We have performed a small comparative study with two of these methods: BLEU (Papineni, Roukos, Ward and Zhu 2002), an n-gram-based precision method, and ROUGE (Lin and Hovy 2003), a unigram recall method. We have done a blind manual evaluation on the first 200 sentences produced by the baseline system, the rules, the memory-based learner and the gold standard data, and applied the two automatic scoring methods as well as our $F_{\beta=1}$ evaluation to the same sets of sentences. The results can be found in Table 7.

The numbers obtained in this evaluation study do not reveal a clear relation between the manual evaluation and the automatic scoring methods. According to the human, the rules outperform the learner with respect to producing perfect sentences but all automatic scoring systems rate TiMBL higher than the rule-based system. It is no surprise that the memory-based learning system does not rate well with respect to syntactic correctness because, unlike the rule-based system and even the baseline, it does not include explicit or implicit syntactic checks.

We believe that a main cause of the scoring differences lies in the checks performed to determine the syntactic correctness of the reduced sentences. While the human can rely on a vast body of knowledge for determining syntactic correctness of sentences, the automatic scorers compare the reduced sentence only with a single source sentence. This immediately suggests an improvement for the automatic evaluation methods, at least for the precision-based BLEU approach: comparisons should be made with a larger corpus rather than with a single sentence.

5 Concluding remarks and future work

We have presented work on sentence simplification targeted at the automatic generation of Dutch TV subtitles. We have performed this task with machine learners. We have described building the required training corpus for the learners: collecting the data, tokenizing them and aligning sentences. After this we have built a shallow parser for Dutch based on the material generated by the project Corpus Gesproken Nederlands. The shallow parser performs part-of-speech tagging, lemmatizing, text chunking and relation finding.

In our sentence simplification work we have used the pairs of sentences from the corpus that we expected to be most appropriate for the learners: those that were different but shared enough words to enable the learner to successfully reduce them. We have applied a general memory-based learner (TiMBL) and a set of hand-crafted deletion rule to this data. Both systems outperformed a baseline system. We have subsequently shown that our evaluation procedure is not always fair to the systems. We have briefly examined other automatic scoring methods and found that none that we had to our disposal attained the quality we can expect from a (labor-intensive) manual evaluation. Although the automatic evaluators rated the memory-based learner higher than the rules, a manual evaluation showed the opposite.

In this study we have approached sentence reduction as a two-step process: first mark all possible deletion and replacement candidates and then decide which words to change based on the required compression rate. We believe that this is the best approach to this task, better than a single step approach, something which we have attempted earlier but found unsuitable (Daelemans, Höthker and Tjong Kim Sang 2004).

With respect to further improvement of our system, we believe that improving the shallow parser preprocessor will lead to better results. Currently it generates only one relation, the verb-subject relation. It would be very useful if the parser would recognize a large range of verb-related phrases: agents, patients and modifiers of different types, like those containing time and space specifications. Such a shallow parser would allow learning sentence simplification from non-parallel corpora: from the relations present in the corpus the learner would discover which types of phrases are compulsory for a verb and which are optional.

However, the prime challenge in this sentence reduction work remains finding better automatic evaluation methods: automatic scoring systems that correlate strongly with human evaluation. We expect much from an expansion of BLEU in which the system uses larger bodies of text as reference rather than a single sentence. Usually the performance of BLEU is improved by creating extra task-specific reference sentences but using a language model build from raw text is a promising cheap alternative.

Acknowledgements

A summary of the approach to sentence simplification put forward in this paper has been presented in Daelemans, Höthker and Tjong Kim Sang (2004). We would like to thank Stans De Meulder for assisting in transcribing the material from the soap Thuis and two anonymous reviewers for useful comments on an earlier version of this paper. Anja Höthker is supported by the European Union's Fifth Framework Programme as a researcher in the MUSA project. Erik Tjong Kim Sang is funded by IWT STWW as a researcher in the ATraNoS project. Both the memory-based version and the rule-based version of the sentence simplifier are available for testing at http://cnts.uia.ac.be/cgi-bin/atranos

References

Caruana, R. and Freitag, D. (1994), Greedy Attribute Selection, *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, USA, Morgan Kaufman, pp. 28–36.

Church, K. W. (1993), Char_align: A Program for Aligning Parallel Texts at the Character Level, *Proceedings of ACL-93*, Columbus, OH, USA, pp. 1–8.

Daelemans, W., Höthker, A. and Tjong Kim Sang, E. (2004), Automatic Sentence Simplification for Subtitling in Dutch and English, *Proceedings of LREC-2004*, Lisbon, Portugal.

Daelemans, W., Zavrel, J., Van der Sloot, K. and Van den Bosch, A. (2003a), *MBT: Memory Based Tagger, version 2.0, Reference Guide*, ILK Technical Report ILK-0313. http://ilk.uvt.nl/.

Daelemans, W., Zavrel, J., Van der Sloot, K. and Van den Bosch, A. (2003b), *TiMBL: Tilburg Memory Based Learner, version 5.0, Reference Guide*, ILK Technical Report ILK-0312. http://ilk.uvt.nl/.

Gale, W. A. and Church, K. W. (1993), A Program for Aligning Sentences in Bilingual Corpora, *Computational Linguistics* **19**(1), 75–102.

Hori, C., Furui, S., Malkin, R., Yu, H. and Waibel, A. (2002), Automatic Speech Summarization Applied to English Broadcast News Speech, *Proceedings of ICASSP 2002*, Orlando, FL, USA, pp. 9–12.

Knight, K. and Marcu, D. (2000), Statistics-Based Summarization – Step One: Sentence Compression, *Proceedings of AAAI-2000*, Austin TX.

Lin, C.-Y. and Hovy, E. (2003), Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics, *Proceedings of HLT-NAACL 2003*, Edmonton, Canada.

Moortgat, M., Schuurman, I. and Van der Wouden, T. (2002), *CGN Syntactische Annotatie*, Progress report Spoken Dutch Project (in Dutch).

Noreen, E. W. (1989), *Computer-Intensive Methods for Testing Hypotheses*, John Wiley & Sons.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. (2002), BLEU: a method for Automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318.

Robert-Ribes, J., Pfeiffer, S., Ellison, R. and Burnham, D. (1999), Semi-automatic captioning of TV programs, an Australian perspective, *Proceedings of TAO Workshop on TV Closed Captions for the Hearing-impaired People*, Tokyo, Japan, pp. 87–100.

Shinyama, Y., Sekine, S., Sudo, K. and Grishman, R. (2002), Automatic Paraphrase Acquisition from News Articles, *Proceedings of HLT 2002*, San Diego, CA, USA.

Van der Beek, L., Bouma, G., Daciuk, J., Gaustad, T., Malouf, R., Van Noord, G., Prins, R. and Villada, B. (2002), *Algorithms for Linguistic Processing*, NWO PIONIER Progress report.

Van Halteren, H., Zavrel, J. and Daelemans, W. (2001), Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems, *Computational Linguistics* **27**(3), 199–229.

Van Rijsbergen, C. (1975), *Information Retrieval*, Buttersworth.