

Using a Parallel Transcript/Subtitle Corpus for Sentence Compression

Vincent Vandeghinste*, Erik Tjong Kim Sang†

*Centre for Computational Linguistics
K.U.Leuven, Belgium
vincent.vandeghinste@ccl.kuleuven.ac.be

†CNTS
University of Antwerp, Belgium
erik.tjongkimsang@ua.ac.be

Abstract

In this paper we describe the collection of a parallel corpus (in Dutch) and its use in a sentence compression tool with the intention to automatically generate subtitles for the deaf from transcripts of a television program. First, the collection of the corpus is described, together with the manipulations and transformations performed on that corpus. Second, a hybrid sentence compression tool is described together with its evaluation.

1. Introduction

When presenting subtitles on a television screen, there is a technical requirement that there is only room for two lines of 32 characters¹. Research shows that a reading time of six seconds is enough for a subtitle of two full lines, and five seconds is the absolute lower limit for prelingual deaf people. The presentation time is between 690 and 780 characters per minute which is more or less 5.5 seconds for two lines (ITC, 1997; Dewulf & Saerens, 2000). Shorter subtitles are time-scheduled proportionally, although research showed that people spend proportionally more time on two-line subtitles than on one-line subtitles. (De Bruycker & d'Ydewalle, 2003). The minimum duration of a short subtitle is 1 second and 12 frames (Dewulf & Saerens, 2000), while the maximum duration of a two-line subtitle is six seconds (van Son et al., 1998)².

If a television program contains fast speech, transcribing the speech would lead to a lot more than 64 characters for 6 seconds. This is why we need sentence compression. The parallel corpus described in this paper consists of transcripts of television programs on one hand and subtitles of these television programs on the other hand. This parallel corpus is used to do sentence compression with a hybrid tool which uses data gathered from this parallel corpus. Hence we can estimate probabilities of removal of certain sentence parts based on the behavior of human subtitlers. To avoid generating ungrammatical sentences the system also uses a set of rules about which sentence parts should not be removed from the sentence.

The creation of the parallel corpus, and the estimation of the removal-probabilities is described in section 2. The sentence compression tool and its evaluation are described in section 3. and conclusions are drawn in section 4.

¹Some broadcasts, especially the news can have three lines of 32 characters (Dewulf & Saerens, 2000).

²Exceptionally the subtitle can have a longer presentation time (e.g. when there is a very slow speaker) (Dewulf & Saerens, 2000).

2. The Parallel Corpus

In this section we describe the corpus and its alignment.

2.1. Collecting the Parallel Corpus

The parallel corpus contains three sections. The first section consists of news broadcasts of the Flemish public broadcasting organization VRT. Rough transcripts of the daily 19:00 news broadcasts have been provided by the organization. Teletext subtitles have been downloaded daily at our systems which are equipped with external Teletext receiver hardware³. The second section of the corpus consists of news broadcasts of the Dutch public broadcasting station NOS. Autocue text and subtitles of the daily 20:00 broadcasts were provided to us by the University of Twente, The Netherlands who have used this material in the DRUID project⁴.

The third section of the corpus contains transmissions of the Flemish soap *Thuis* (VRT). Teletext subtitles have been obtained in the same way as for the VRT news broadcasts. However, since we could not acquire scripts for this programme, the transcriptions have been made by volunteers of the University of Antwerp. The *Thuis* section is the smallest of the three: 7 broadcasts with 20,387 words in the subtitle part (from the years 2000 and 2002). The NOS section contains 125 broadcasts with 230,295 words (1999, 2002). The VRT section is the largest: 101 broadcasts containing a total of 431,190 words (2001, 2002).

2.2. Aligning the Corpus

Once the corpus is collected we need to determine which are the corresponding parts. The corpus is aligned on two levels: sentence level and chunk level.

2.2.1. Sentence Alignment

Our initial plan was to use the standard sentence alignment method of Gale and Church (1993) for aligning the

³<http://www.opt.com/>

⁴<http://dis.tpd.tno.nl/druid/>

sentences of the transcripts to the subtitle sentences. However, this character length based method proved to be inappropriate for our data because they contain gaps. In the VRT section, interviews with sport people were subtitled but not transcribed. The NOS section lacks any transcript for non-anchor text: the autocue text only holds the text of the host of the programme. In both news sections, listings printed on the screen, like for example quotes, are transcribed but not subtitled. A length-based alignment method like the one of Gale and Church does not work very well for parallel texts in which parts are missing.

We have developed a lexicalized alignment method which links sentences to each other when they occur in similar locations in two texts and contain similar words. The algorithm makes four passes over the data. The first pass only aligns sentence pairs which are almost identical. The second pass and the third pass perform the same task as the first but with a relaxed acceptance threshold. After every pass crossing links are removed and n-to-1 sentence links are reduced to 1-to-1 links. In the final pass, sentences that have not yet been linked to others are added to the alignment structure of one of their neighbors if their lexical contents makes such a link appropriate.

The material of the corpus has been aligned with this automatic method. After this, all alignment links have been checked manually. As a result of this, it was possible to evaluate the alignment algorithm by comparing its output with the corrected versions. Both the precision and recall figures for finding sentence pairs in the VRT section of the corpus are 91%.

2.2.2. Chunk Alignment

The complete processing flow of the sentence aligned parallel corpus is sketched in figure 1. Before chunk alignment can be applied the sentence-aligned parallel corpus needs to undergo a few preprocessing steps:

The sentence-aligned files need to be tagged to estimate their parts-of-speech. This is done by applying TnT (Brants, 2000), which was trained on internal release 6 of the Spoken Dutch Corpus (CGN)⁵: we used the part-of-speech tagset which was developed for CGN (Van Eynde, 2004). As in CGN all occurring accents are represented by their html equivalents (e.g. ð becomes ë); this transformation is applied on the transcription part of the parallel corpus. There is no need to do this transformation on the subtitle part of the corpus as all accents are lost during the collection of teletext subtitling. After tagging, these accents will be removed from the corpus to allow a more exact chunk alignment. The accuracy of the part-of-speech tagger TnT trained on CGN is reported to be 96.2% (Oostdijk et al., 2002).

After part-of-speech tagging, the sentence-aligned corpus needs to be chunked. For chunking we used ShaRPa (Vandeghinste, submitted), a rule-based chunker with chunking grammars developed for Dutch. The chunking accuracy for noun phrases has an F-value⁶ of 94.7%, while the chunking accuracy of prepositional phrases has an F-

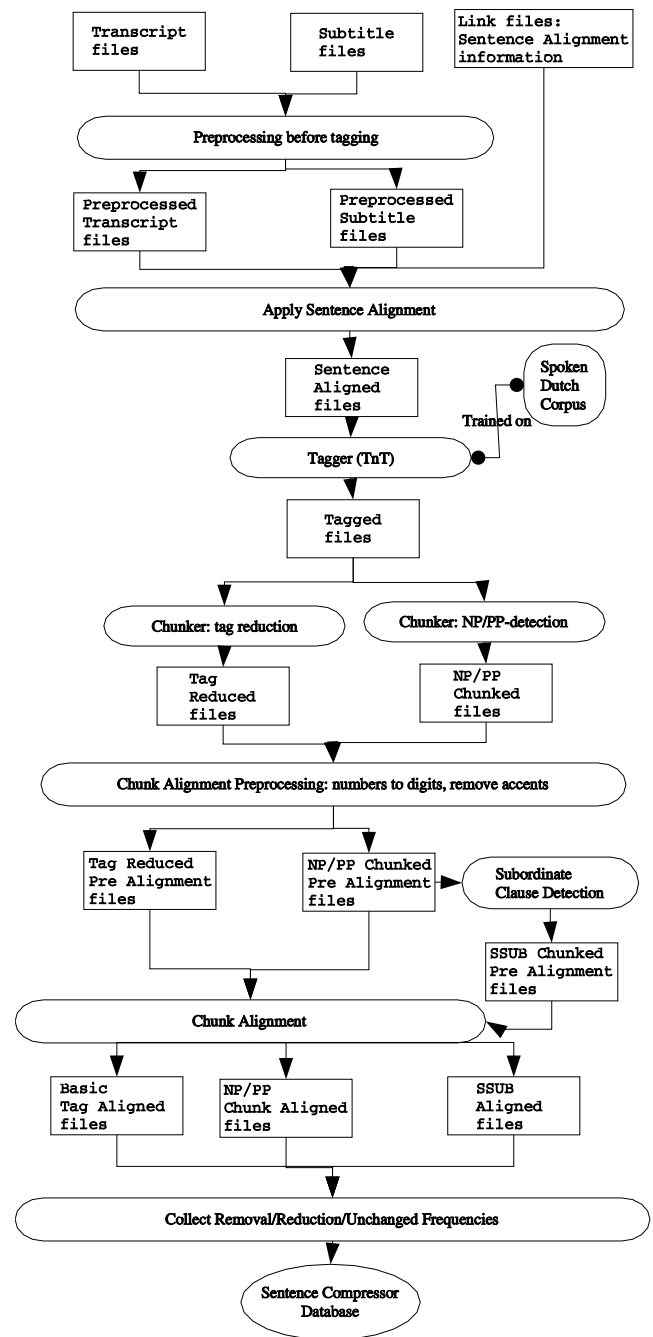


Figure 1: Parallel Corpus Processing Flow

value of 95.1%. Because CGN contains a large fine-grained tagset (over 300 different Part-of-Speech tags) and to avoid data sparseness, the tagset was reduced. Chunk alignment is done at the token level for all tokens not belonging to a noun or prepositional phrase.

The source and target files undergo some more preprocessing. All numbers written in full are converted to digits, both for the source as for the target side of the corpus, as well for cardinal as for ordinal numbers, because the target sentence may already contain digits, and this facilitates chunk alignment.

Now, both the source part (transcripts) and the target part (subtitles) of our corpus are chunked. As an align-

⁵<http://lands.let.kun.nl/cgn/ehome.htm>

⁶The F-value is defined according to Manning & Schütze (2000:268-9).

ment procedure, we compare each chunk from the source sentence with each chunk from the target sentence. Every 4-gram of letters from the source chunk is compared with every 4-gram of letters from the target chunk. The alignment measure (A) is calculated like this⁷:

$$A = \frac{m}{m+n} \frac{L_1 + L_2}{2}$$

where

m is the number of matching 4-grams

n is the number of non-matching 4-grams

L_1 is the length of the source chunk

L_2 is the length of the target chunk

When there is a perfect match, the alignment measure is 1. When there are no matching 4-grams, the alignment measure is 0. Partially matching 4-grams have a value somewhere inbetween.

When aligning the chunks, we need to set a threshold on this alignment measure. With the threshold set to 0.315, the F-value is approximately 95% for NP and PP alignment and for basic token alignment.

2.3. Estimating Removal Probabilities

From the chunk-aligned corpus, we can estimate a number of sentence compression parameters by checking how often certain phenomena occur.

For instance, we have checked how often a prepositional phrase in the source corpus, starting with the preposition *op* corresponds exactly to a prepositional phrase in the target corpus, how often we do not find a corresponding chunk in the target corpus, and how often we find a corresponding chunk which is shorter than the original chunk.

3. The Sentence Compression Tool

The Sentence Compression Tool takes a sentence as its input, and generates a reduced sentence. The amount of reduction needed for an appropriate subtitle generation is a parameter which can be set for each sentence separately. The approach of our system is inspired by Jing (2001). Jing uses multiple sources of knowledge on which his sentence reduction is based. He uses a corpus of sentences, aligned with human-written sentence reductions which is similar to our parallel corpus. He uses a syntactic parser to analyse the syntactic structure of the input sentences. As there was no syntactic parser available, we created ShaRPa, a chunker (Vandeghinste, submitted) which could give us a shallow parse tree of the input sentence. Jing uses several other knowledge sources, which are not used (not available for Dutch) or not yet used in our system.

3.1. Design of the Sentence Compression Tool

3.1.1. Preparing for Sentence Compression

In the same way that the sentences from the corpus are preprocessed before being chunk-aligned, the input sentence gets preprocessed: in a first stage, the sentence is

⁷If we have a perfect match with a 5-letter chunk, $\frac{m}{m+n} = \frac{5}{5+20} = \frac{1}{5}$. So we need to multiply this by the average chunk length $\frac{L_1+L_2}{2} = \frac{5+5}{2}$.

converted into a pretagger format. The tagger we use is Brants' (2000) TnT. The tagger requires an input file with one word per line and was trained on a preliminary version of the CGN corpus (Internal Release 6), and is a purely statistical tagger, based on mono-, bi-, and trigram tag frequencies.

In a second stage the sentence is sent to the Abbreviator. The Abbreviator connects to a database of common abbreviations, which are often pronounced in full words (E.g. *European union* becomes *EU*), and replaces the full form by the abbreviation. The output of the Abbreviator serves as the input for ShaRPa. A last step in the preparation of the sentence before actual reductions can be generated is detecting the subordinate clauses in the sentence.

3.1.2. The Actual Compression

For each chunk resulting from all the previous steps, the probability of removal and the probability of non-removal are estimated from the frequencies of removal and non-removal of chunks in the chunk-aligned parallel corpus. Besides the statistical component in the compression, there are also a number of rules in the compression program, which state which daughters should not be removed if the mother is of a certain type. For instance, the system should never remove the head noun of an NP, unless the whole NP is removed.

3.1.3. A Final Reduction

After compression is done, long words are sent to the WordSplitter. This module checks if a word can be split up in two parts and is not in a list of words which should not be split up⁸. If this is the case, the compound is replaced by its head. To check if a word can be split-up into two parts, the system checks if the two parts can be recompounded, and the WordSplitter module makes use of a hybrid automated compounding module, which is described in more detail in Vandeghinste (2002). This results in an extra reduction of the input sentence.

All the possible outcomes of the system are sorted by their probability, and the most probable result that complies with the length restrictions becomes the output of our system.

3.2. Evaluation

The evaluation of a sentence compression module is not an easy task. The output of the system needs to be judged manually for its accurateness. This is a very time consuming task. Unlike Jing (2001), we do not compare the system results with the human sentence reductions. Jing reports a 81.3% success rate for his program, but this is the percentage of decisions on which the system agrees with the human compressor. The results presented here are calculated on the sentence level: the amount of valid sentence reductions.

It should be made clear that the evaluation which is presented in this section is not an evaluation of a subtitling

⁸E.g.: the word *voetbal* [E: football] can be split up into *voet* [E: foot] and *bal* [E:ball], but *voetbal* should never be replaced by *bal*.

module, but the evaluation of a sentence compression module. A subtitling module should calculate a per-sentence reduction ratio and generate an appropriate reduction, based on the time it took the speaker to pronounce the sentence. The compression module is evaluated by reducing 10% and 20% on all sentences in the testset. This is not a situation as it would occur in real life, but it is a good measure to detect the flaws in the compression system.

The sentence compression system is evaluated on 200 real verbatim transcription sentences of television program. We produce two different reductions. The first reduction is a 10% reduction (when counting the number of characters in the sentence). On average, this allows a speaker rate of approximately 110 words per minute. The second reduction is a 20% reduction (allowing approx. 127 w/min).

The resulting reduced sentences are classified by human judgement according to their informativity. They score: + (accurate compression), +/- (a more or less accurate compression: some information is missing, but can possibly be derived from the context), - (an inaccurate compression), or 0 (no compression: this can be due to the fact that the source sentence cannot be validly reduced, not even by human compressors). Each sentence is evaluated by two judges. Only the compressions on which both judges score the same are taken into account, together with the compressions on which both judges score at least +/- . The test-sentences for which the system gives no output are not included in the compression accurateness percentages. Results are presented in table 1.

Reduction Rate	10%	20%
Interrater Agreement	84.6%	85.4%
No output (0)	6.0%	10.5%
Accurate Compression	41.5%	30.7%
+/- Acc. Compression	13.8%	12.8%
Reasonable Compression	55.3%	43.6%

Table 1: Evaluation results on 10% and 20% reductions

4. Conclusion

Using a parallel corpus provides a means for estimating removal and reduction probabilities in sentence compression, based on human behaviour. There are still several weak points in our system: most errors are due to tagging and chunking errors, which would be avoided if a full parse were available. Another type of errors concerns the removal of words which are predicates or parts of a collocation.

The setup of the system seems to yield promising results. Once the weak points can be avoided by using better sentence analysis tools and a collocation handling mechanism, the results of our system will be much better, while using the same setup. Our future research will hence focus on the enhancement of the sentence analysis tools used here and on a collocation handling mechanism.

Most important is the fact that, provided a correct sentence analysis, the removal estimates seem to often have the desired effect of removing the parts in the correct order with concern to their informativity.

It should also be taken into account that when applying the system to real life applications, the system can be tuned to perform better⁹, but this was not done in the evaluation presented here.

5. Acknowledgements

Research funded by IWT (Institute for Innovation in Science and Technology) in the STWW program, project ATraNoS (Automatic Transcription and Normalisation of Speech). For more information visit <http://atranos.esat.kuleuven.ac.be/>. I would like to thank I. Schuurman for judging the reduced sentences.

6. References

- Brants, T. (2000). TnT - A Statistical Part-of-Speech Tagger. <http://www.coli.uni-sb.de/thorsten.tnt>.
- De Bruycker, W., and d'Ydewalle, G. (2003). Reading native and foreign language television subtitles in children and adults. In J. Hyönä, R. Radach & H. Deubel (Eds.), *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research* pp. 671-684. Oxford, UK: Elsevier.
- Dewulf, B., and Saerens, G. (2000). *Stijlboek Teletekst Ondertiteling. Internal Subtitling Guidelines.* VRT. Brussels.
- Gale, W.A., and Church, K.W. (1993). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics* 19 (1), pp.75-102.
- ITC Guidance on Standards for Subtitling. (1997). Online at http://www.itc.org.uk/codes_guidelines/broadcasting/tv/sub_sign_audio/subtitling_stnds/
- Jing, H. (2001). *Cut-and-Paste Text Summarization.* PhD Thesis. Columbia University.
- Manning, C.D., and Schütze, H. (2000). *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, Massachusetts.
- Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.P., Moortgat, M., Baayens, H. (2002). Experiences from the Spoken Dutch Corpus Project. *Proceedings of LREC2002* pp340-347. ELRA. Paris.
- Vandeghinste (2002). Lexicon Optimization: Maximizing Lexical Coverage in Speech Recognition through Automated Compounding. *Proceedings of LREC2002.* ELRA. Paris.
- Vandeghinste (submitted). ShaRPa: Shallow Rule-based Parsing, focused on Dutch. *Proceedings of CLIN2003.*
- Van Eynde, F. (2004). Tagging and Lemmatisation for the Spoken Dutch Corpus. Internal report.
- Van Son, N., Verboom, M., Van Balkom, H. (1997). Toegankelijkheid van TV-programma's. Eindrapport van een bronnenonderzoek. Instituut voor Doven, Sint-Michielsgestel.

⁹E.g. by adding words to the list of words which cannot be split up