# Synchronizing Translated Movie Subtitles

**Jörg Tiedemann**

Information Science
University of Groningen
PO Box 716
9700 AS Groningen, The Netherlands
j.tiedemann@rug.nl

**Abstract**

This paper addresses the problem of synchronizing movie subtitles, which is necessary to improve alignment quality when building a parallel corpus out of translated subtitles. In particular, synchronization is done on the basis of aligned anchor points. Previous studies have shown that cognate filters are useful for the identification of such points. However, this restricts the approach to related languages with similar alphabets. Here, we propose a dictionary-based approach using automatic word alignment. We can show an improvement in alignment quality even for related languages compared to the cognate-based approach.

## 1. Introduction

Movie subtitles in various languages are available on-line in ever growing databases. They can be compiled into diverse collections of parallel corpora useful for many cross-lingual investigations and NLP applications (Lavecchia et al., 2007; Volk and Harder, 2007; Armstrong et al., 2006). Although they ought to be aligned to the corresponding movies, on-line subtitles suffer from a serious problem of synchronization differences. This is due to the process of creating textual subtitle files which is mainly done by "ripping" (i.e. scanning) them from DVDs using various tools.

In previous studies, we have shown that time information is a valuable feature for proper subtitle alignment (Tiedemann, 2007). However, synchronization differences in terms of starting time and speed cause serious alignment problems as shown in the same study. In (Tiedemann, 2006a), several ways of synchronizing misaligned subtitles have been discussed already. Synchronization is done by re-computing time information in one subtitle file by adjusting speed and time offset according to two aligned fix-points in a pair of subtitles. The remaining problem is to find appropriate fix-points that can be used for this procedure. Besides of manually defining them two automatic methods have been compared in (Tiedemann, 2006a). They are both based on a "cognate filter" using string similarity measures and some heuristics for selecting the two points necessary for synchronization. Although this technique produces promising results there are some obvious shortcomings of using simple string similarity measures. First, there is the risk of finding false friends. However, the chance of false friends in corresponding sentences and their local context is rather low. Secondly, the risk of selecting the wrong candidate is high in cases where names are frequently repeated in close context. The impact of such erroneous selections is minimized by considering all combinations of candidates and selecting the most promising pair according to some heuristics as described in (Tiedemann, 2006a). Finally, the most severe drawback of string similarity measures is the restriction to languages with at least similar alphabets. This problem will be addressed in this paper.

The remaining part is organized as follows: First, we will briefly describe the data we collected. Thereafter, we will shortly summarize our sentence alignment approach. Finally, we will discuss the synchronization of subtitles using a dictionary filter including an evaluation of some sample data.

## 2. Data Collection

We collected data from one on-line provider of movie subtitles, http://www.opensubtitles.org. All data files have been converted to a standalone XML format and UTF8 encoding. We also applied a language classifier to clean the database. Details are given in (Tiedemann, 2007). The current collection contains 22,794 pairs of subtitles in 29 languages covering 2,780 movies. Figure 1 lists some statistics of the 15 largest bitexts in our collection.

| language pair | nr sentences | | nr words | |
|---|---|---|---|---|
| | source | target | source | target |
| eng-spa | 592,355 | 524,412 | 4,696,792 | 4,071,345 |
| por-spa | 443,521 | 414,725 | 3,124,539 | 3,170,790 |
| cze-eng | 403,605 | 421,135 | 2,581,318 | 3,260,751 |
| eng-por | 397,085 | 370,866 | 3,071,277 | 2,611,508 |
| eng-slv | 394,941 | 376,971 | 3,036,584 | 2,343,233 |
| eng-swe | 386,269 | 339,953 | 2,971,600 | 2,441,469 |
| dut-eng | 378,475 | 425,600 | 2,804,742 | 3,338,842 |
| dut-spa | 367,421 | 359,944 | 2,729,557 | 2,739,981 |
| cze-por | 365,676 | 366,861 | 2,311,908 | 2,532,080 |
| cze-spa | 361,038 | 335,278 | 2,278,212 | 2,532,657 |
| cze-rum | 347,454 | 345,553 | 2,220,880 | 2,491,271 |
| por-rum | 340,227 | 335,356 | 2,352,743 | 2,412,681 |
| cze-slv | 328,751 | 335,555 | 2,093,731 | 2,123,347 |
| eng-pob | 323,621 | 308,458 | 2,525,747 | 2,183,897 |
| pob-spa | 320,934 | 293,701 | 2,280,703 | 2,340,992 |

Figure 1: The 15 largest bitexts in the subtitle corpus

## 3. Time Slot Alignment

An important steps in building parallel corpora is the alignment of textual units at some level. Commonly, corresponding sentences are aligned assuming that they are the smallest linguistic units that still can be aligned monotonically

between two languages. In (Tiedemann, 2007), we have shown that standard length-based approaches fail for our kind of data and that sentence alignment using time information is superior to these techniques. Basically the *time-slot alignment* approach is based on the assumption that corresponding texts are shown at approximately the same time on screen and, hence, text fragments with the largest time overlap are aligned to each other. The details are described in (Tiedemann, 2007). This approach nicely handles cases of deletions and insertions which usually cause major problems in alignment.

## 4. Subtitle Synchronization

Despite the fact that the simple alignment approach described in the previous section works well for perfectly synchronized subtitles, it badly fails for pairs of subtitles even with only slight timing differences. Unfortunately, synchronization differences are quite common among the data files we have collected. The mis-synchronization problems come down to differences in starting time and the speed of showing subtitle frames together with the movie. This is caused by the software used for creating the plain text subtitle files. Therefore, media players commonly include features to adjust the timing manually.

For sentence alignment we do not require proper alignment of subtitles to movies but proper alignment of subtitles to each other. However, this involves the same adjustments of starting time and displaying speed for one of the subtitle files in order to apply the time-slot alignment method. The approach suggested in (Tiedemann, 2006a) uses two fixed anchor points to compute the speed and time offset for a given subtitle pair to recalculate time stamps in one of the files. Having two anchor points with time stamps $< src_1, trg_1 >$ and $< src_2, trg_2 >$ ($src_x$ corresponds to the time in the source language subtitle file and $trg_x$ to the time in the target language file) we can compute the $time_{ratio}$ and the $time_{offset}$ as follows:

$$\begin{aligned} time_{ratio} &= \frac{(trg_1 - trg_2)}{(src_1 - src_2)} \\ time_{offset} &= trg_2 - src_2 * time_{ratio} \end{aligned}$$

Using these two parameters we now adjust the time stamps in the source language file by simply multiplying each of them by the $time_{ratio}$ and adding the value of $time_{offset}$.

A crucial step for this technique is to find appropriate anchor points for the synchronization. Essentially, we need to find pairs of subtitle frames which truly correspond to each other. We can then use the time stamps given at the beginning of each of these subtitle frames to compute the synchronization parameters. The best result can be expected if the two anchor points are as far away from each other as possible. Therefore, we need to look for corresponding frames in the beginning and at the end of each subtitle pair. There are several ways of finding such anchor points. In the following we first discuss the strategies previously used and, thereafter, we present our new extension using bilingual dictionaries derived from automatic word alignment.

### 4.1. Manually Adding Anchor Points

The safest way of synchronizing subtitles using anchor points is to manually mark corresponding frames. For this, the interactive sentence alignment front-end, ISA (Tiedemann, 2006b) can be used. ISA includes features to manually add hard boundaries before aligning a bitext. It allows to easily jump to the end and back to the beginning of the current bitext and hard boundaries can simply be added and deleted by clicking on corresponding sentences. A screenshot of the interface is shown in figure 2.

The alignment back-end is defined in a corpus specific configuration file. ISA passes all existing hard boundaries as parameters to the back-end in case the time-slot aligner for subtitles is used.

The advantage of the manual approach is that the alignment can be done interactively, i.e. the resulting alignment can be edited or synchronization can be repeated using different anchor points. Furthermore, the user may decide if synchronization is necessary at all. However, manual synchronization and interactive alignment is not an option for large amounts of data. Therefore, we use automatic techniques for anchor point identification as discussed in the following two sections.

### 4.2. Using Cognates for Synchronization

An obvious idea for anchor point identification is to use so-called "cognates"[1] known to be useful for sentence alignment (see, e.g., (Simard et al., 1992; Melamed, 1996)). The cognate approach for anchor point identification in subtitle alignment has already been used in the study presented in (Tiedemann, 2007). We basically scan the beginning and the end of the bitext for cognate pairs using a string similarity measure and a sliding window. The two pairs of cognate candidates which are furthest away from each other are then used for synchronization by simply using the time stamps given at the beginning of the sentences in which they appear.

The cognate approach works well for languages which use identical or at least similar alphabets especially because subtitles usually contain a lot of names that can easily be matched. Obviously, simple string matching algorithm do not work for language pairs with different alphabets such as English and Bulgarian as shown in figure 2 even though there might be a lot of closely related words (such as names transliterated using the respective alphabet). One possibility to use string similarity measures for such languages is to define scoring schemes to match arbitrary character pairs from both alphabets. The problem here is to define appropriate matching functions for each language pair under consideration. There are certainly ways of learning such functions and it would be interesting to investigate this direction further in future work. Another possibility for finding anchor points is to use bilingual dictionaries. This approach will be discussed in the following.

### 4.3. Word Alignment for Synchronization

Using bilingual dictionaries is a straightforward idea to find candidates for aligned anchor points in the same fashion

[1]The term cognate is used here in terms of words which are similar in spelling.

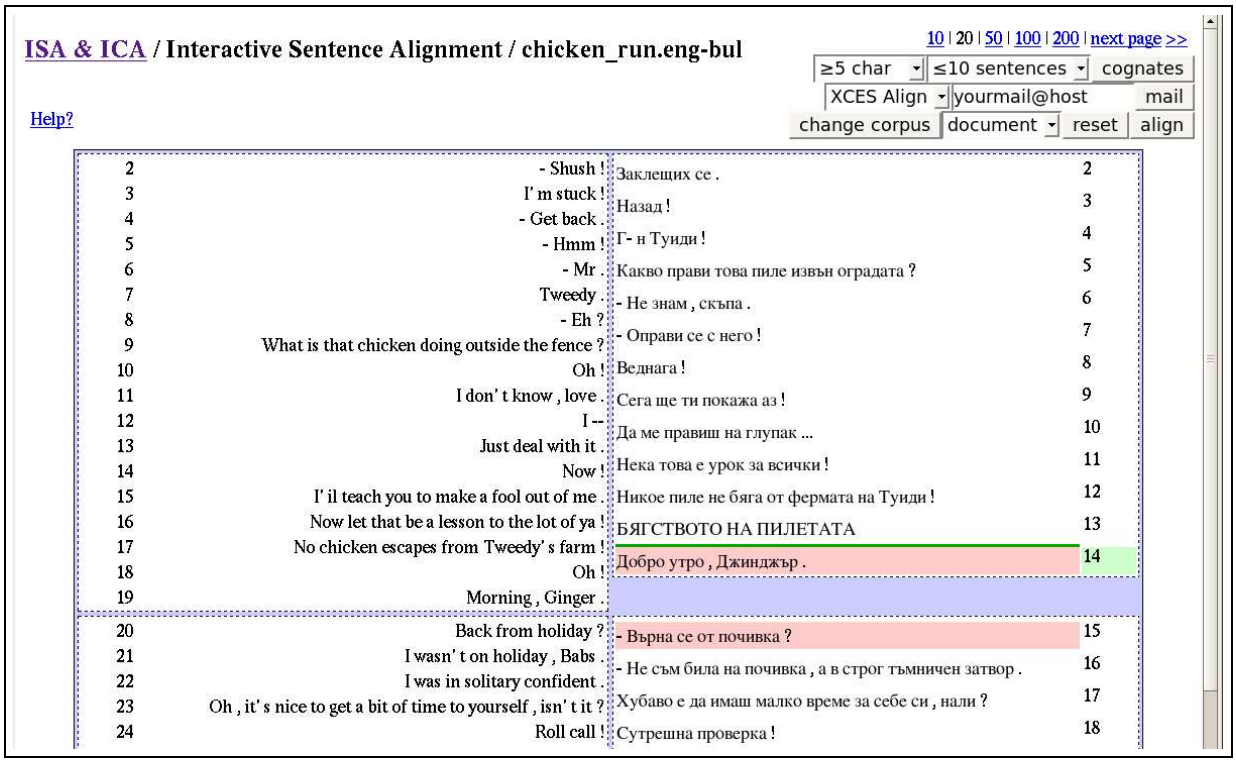


Figure 2: Manually adding hard boundaries to an English/Bulgarian subtitle pair (from the movie “Chicken Run”) using the Interactive Sentence Aligner ISA

as it is done by the cognate filter described earlier. Now the task is to obtain appropriate dictionaries. We still like to keep the alignment approach as language independent as possible and therefore we do not want to rely on existing machine-readable dictionaries. Naturally, dictionaries are the opposite of a language-independent resource. However, this is not an issue if they can be created automatically for any given language pair. Fortunately, word alignment software is capable to find common translational equivalences in a given parallel corpus and, hence, “rough” bilingual dictionaries can be extracted from word aligned corpora. For the following we assume that word alignment is robust enough even for parallel corpora with many sentence alignment errors. In particular, we assume that at least the frequently aligned words correspond to good translation equivalents. The procedure is as follows:

1. sentence align the entire corpus using the time-slot overlap approach (using cognate filters if applicable)
2. word-align each parallel corpus using GIZA++ (Och and Ney, 2003) using standard settings in both alignment directions
3. use the intersection of both Viterbi alignments (to obtain high precision) and extract relevant word correspondences (using alignment frequency thresholds)
4. run the sentence aligner once again with language-pair specific dictionaries

Important in our setting is to extract reliable word translations. Using the intersection of statistical word alignments reduces already the noise of the alignment dramatically. Furthermore, we simply set a frequency threshold for the extraction of translation equivalents and restrict ourselves to tokens of a minimal length that contain alphabetic characters only.

### 4.4. Anchor Point Selection

An obvious drawback of automatic synchronization approaches compared to manual synchronization is the risk of mis-synchronization. Both, cognate and dictionary based approaches bare the risk to select inappropriate anchor points which may cause an even worse alignment than without any synchronization. However, anchor point selection is cheap when considering limited windows (initial and final sections) only. Dictionaries are static and do not have to be re-compiled each time an alignment has to be repeated and token based string comparison is also fast and easy. Hence, various candidate pairs can be tested when applying synchronization. The main difficulty is to choose between competing candidates in order to select the one that yields the best sentence alignment in the end. Here, we apply a simple heuristics that works well in our experiments. Knowing that incorrect synchronization causes a lot of mismatches between time slots we assume that the alignment in such cases includes many empty alignments (one sentence in one language aligned to no sentence in the other language). On the other hand we expect well synchronized subtitles to align nicely with only a few empty links. Hence, we use the alignment-type ratio (Tiedemann, 2006a) to measure the relative “quality” of an alignment compared to other possible alignments:

$$algtype_{ratio} = \frac{\text{number of non-1:0-alignments} + 1}{\text{number of 1:0-alignments} + 1}$$

Using $algtype_{ratio}$ as indicator for alignment quality we can now apply all possible pais of anchor point candidates and select the one that maximizes this ratio. This strategy is also applied in the experiments described below.

## 5. Experiments

In our experiments we used Dutch-English and Dutch-German subtitles (mainly because evaluation data was readily available from previous experiments). The intersection of word token links resulted in 111,460 unique word type pairs for Dutch-English and 45,585 pairs for Dutch-German. We extracted all word pairs with alignment frequency larger than 5 and, furthermore, we removed all pairs that contain non-alphabetic characters and words shorter than 5 characters. In this way we discarded most of the alignment errors and also dismissed most of the highly frequent function words which would cause a lot of false hits when looking for anchor points for synchronization. The resulting dictionaries consist of 4,802 pairs (Dutch-English) and 1,133 pairs (Dutch-German). Figure 3 shows a small sample of the Dutch/English and Dutch/German word alignment dictionaries.

| Dutch/English | Dutch/German |
|---|---|
| misschien maybe | waarom warum |
| hallo hello | hebben haben |
| sorry sorry | alles alles |
| bedankt thanks | leven leben |
| omdat because | niets nichts |
| alsjeblieft please | vader vater |
| luister listen | misschien vielleicht |
| bedankt thank | weten wissen |
| niets nothing | zeggen sagen |
| mensen people | moeten müssen |
| alles everything | kunnen können |
| dacht thought | altijd immer |
| nooit never | gezien gesehen |
| jezus jesus | terug zurück |
| vader father | bedankt danke |
| ... | ... |
| alleen wanna | begonnen begonnen |
| alleen there | bedoelt meinst |
| alleen people | banken banken |
| alleen about | badal badal |
| alden alden | atlanta atlanta |
| akkoord agreed | alsjeblief bitte |
| afdrukken prints | aiies aiies |
| adios adios | achter hinten |
| aanvallen attack | aartsbisschop erzbischof |
| aanval attack | aarde erden |

Figure 3: The top 15 and the last 10 entries from the Dutch/English and Dutch/German word type alignments (sorted by alignment frequency which is not shown here)

As the figure indicates, the quality of these dictionaries is quite good but not perfect. Especially among less frequent alignments we can see several mistakes even after filtering. For example, in the Dutch/English dictionary there are 5 errors among the last 10 pairs as shown in figure 3. However, due to the exhaustive test of candidate pairs as described in section 4.4. we assume that our method is very robust to such noise in the dictionaries.

In the Dutch/German sample we can see another typical error in our data caused by the software used for ripping the subtitles from the DVDs. Quite frequently we can observe OCR errors such as “aiies” (instead of “alles”) in the data. This, however should not have a negative impact on the synchronization approach. On the contrary, these links might even be very useful for our data collection.

The word alignment dictionaries were now used without further processing for movie synchronization as described earlier. For evaluation purposes we used 10 randomly selected movies and manually aligned 20 initial, 20 intermediate and 20 final sentence of each movie and language pair. The reason for focusing on different regions in the subtitles was originally to investigate the differences between various approaches on specific alignment problems. For example, subtitles often include different amounts of information in the beginning and at the end of a movie. Titles, songs and credits may be (partly) translated in some languages whereas they are not in others. Insertions and deletions usually cause large problems for traditional approaches. However, we could not observe significant differences in these regions between the length-based approach and our time-slot approach. We also wanted to look at the ability to synchronize after initial mistakes but could not see significant differences between the two types of alignment approaches either. Therefore, we will not include separate scores for the three regions but present the overall results only (see table 1).

| approach | correct | partial | wrong |
|---|---|---|---|
| *Dutch - English* | | | |
| length | 0.397 | 0.095 | 0.508 |
| time | 0.599 | 0.119 | 0.282 |
| time-cog | 0.738 | 0.115 | 0.147 |
| time-dic | 0.765 | 0.131 | 0.104 |
| *Dutch - German* | | | |
| length | 0.631 | 0.148 | 0.220 |
| time | 0.515 | 0.085 | 0.400 |
| time-cog | 0.733 | 0.163 | 0.104 |
| time-dic | 0.752 | 0.148 | 0.100 |

Table 1: The quality of different alignment approaches: *length* refers to the baseline using a length-based alignment approach, *time* refers to the time-slot overlap approach. The extension *cog* refers to the application of the cognate filter and *dic* to the dictionary approach.

The following settings have been used for the time-based alignment with movie synchronization: Anchor points are searched in a window of 25 sentence from the beginning and from the end of each subtitle file. As discussed earlier, all combinations of candidate pairs in the initial window and the ones in the final window are tried out and the one that gives the largest alignment-type ratio is taken.

The parameters of the cognate filter are set as follows: min-

imal string length is 5 characters and the similarity measure is the longest common sub-sequence ratio with a threshold of 0.6.

## 6. Discussion and Conclusions

Table 1 shows that there is a clear improvement in alignment quality using the dictionary approach compared to the baseline and also to the other time-based alignment approaches. Note that we consider close related languages with similar alphabets only. We still outperform the cognate based approach, which is an encouraging result considering the noisy word alignment dictionaries used. Concluding from this we expect that the dictionary approach also helps to improve the alignment of more distant language pairs with incompatible alphabets for which the cognate method does not work at all. However, we should be careful with our expectations for several reasons. First of all, there is often less data available for such language pairs. They also often come from less reliable sources and include many corrupt and incomplete files. Furthermore, for many languages various character encodings are used which complicates the pre-processing step. We should also not forget that less related language pairs are more difficult to word-align anyway because of syntactic, morphological and semantic differences. Also we might see even more OCR errors because of the limited language support of the subtitle ripping software used. Finally, the word alignment will be based on a non-synchronized parallel corpus (because the cognate-based synchronization is not applicable). All of these issues will cause smaller and noisier bilingual dictionaries.

We word-aligned the entire corpus with all its language pairs and applied the dictionary approach to all bitexts. Unfortunately, due to time constraints, we were not able to measure the success of the this approach on some example data of distant language pairs. This has to be done in future work and is just a matter of producing appropriate gold standard alignments (which can also be done using ISA). We still expect an improvement even with small and noisy dictionaries due to the robustness of our approach as we discussed earlier. The improvements might be smaller, though, and it could be an idea to iteratively alternate between word alignment and sentence alignment to push the quality further up. However, word alignment is expensive and, therefore, this approach might not be reasonable with current technology. Further investigations in this directin should be carried out in the future.

## 7. Availability

The parallel subtitle corpus is part of OPUS (Tiedemann and Nygard, 2004), a free collection of parallel corpora. The corpora are available from the project website at `http://www.let.rug.nl/tiedeman/OPUS/` including the latest sentence alignment and links to tools and search interfaces. The entire OPUS corpus has been indexed using the Corpus Work Bench from IMS Stuttgart and can be queried on-line. Furthermore, we provide a word alignment database with access to multi-lingual dictionaries derived from automatic word alignment. The database can be queried at `http://www.let.rug.nl/tiedeman/OPUS/lex.php`.

We would like to thank the University of Groningen and Oslo University for providing hard disk space and Internet bandwidth for our resources.

## 8. References

Stephen Armstrong, Colm Caffrey, Marian Flanagan, Dorothy Kenny, Minako O'Hagan, and Andy Way. 2006. Leading by example: Automatic translation of subtitles via ebmt. *Perspectives: Studies in Translatology*, 14(3):163 – 184.

Caroline Lavecchia, Kamel Smaili, and David Langlois. 2007. Building parallel corpora from movies. In *Proc. of the 4th International Workshop on Natural Language Processing and Cognitive Science - NLPCS 2007*, Funchal, Madeira.

I. Dan Melamed. 1996. A geometric approach to mapping bitext correspondence. In Eric Brill and Kenneth Church, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–12, Philadelphia, PA.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Michel Simard, George F. Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–81, Montreal, Canada.

Jörg Tiedemann and Lars Nygard. 2004. The OPUS corpus - parallel and free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal, May.

Jörg Tiedemann. 2006a. Building a multilingual parallel subtitle corpus. In *Proceedings of 17th CLIN, to appear*, Leuven, Belgium.

Jörg Tiedemann. 2006b. ISA & ICA - two web interfaces for interactive alignment of bitexts. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, (LREC'2006)*, Genova, Italy.

Jörg Tiedemann. 2007. Improved sentence alignment for movie subtitles. In *Proceedings of RANLP 2007*, pages 582–588, Borovets, Bulgaria.

Martin Volk and Søren Harder. 2007. Evaluating mt with translations or translators. what is the difference? In *Machine Translation Summit XI Proceedings*, Copenhagen.