Improved Sentence Alignment for Movie Subtitles

Jörg Tiedemann
University of Groningen
Alfa Informatica,
P.O. Box 716
9700 AS Groningen, the Netherlands
j.tiedemann@rug.nl

Abstract

Sentence alignment is an essential step in building a parallel corpus. In this paper a specialized approach for the alignment of movie subtitles based on time overlaps is introduced. It is used for creating an extensive multilingual parallel subtitle corpus currently containing about 21 million aligned sentence fragments in 29 languages. Our alignment approach yields significantly higher accuracies compared to standard length-based approaches on this data. Furthermore, we can show that simple heuristics for subtitle synchronization can be used to improve the alignment accuracy even further.

Keywords

sentence alignment, parallel corpora, multilingual resources

1 Introduction

Sentence alignment is a well-known task applied to parallel corpora as a pre-requisite for many applications such as statistical machine translation [2] and multilingual terminology extraction [9]. It consists of finding a monotonic mapping between source and target language sentences allowing for deletions, insertions and some n:m alignments. Several algorithms have been proposed in the literature mainly based on translation consistency. We can distinguish between the following two main approaches: (1) sentence alignment based on similarity in length [1, 4], and, (2) alignment based on term translation consistency and anchor points [5, 3, 6]. Both techniques can also be combined [8, 7, 14]. It has been shown that these simple, often language independent techniques yield good results on various corpora (see, e.g. [12]) and the problem of sentence alignment is often regarded to as being solved at least to some reasonable degree.

In this paper, we focus on the alignment of movie subtitles, a valuable multilingual resource that is different to other parallel corpora in various aspects: Movie subtitles can be described as compressed transcriptions of spoken data. They contain many fragmental utterances rather than grammatical sentences. Translations of subtitles are often incomplete and very dense in the sense of compressing and summarizing utterances rather than literally transcribing them. They are often mixed with other information such as titles, trailers, and translations of visual data (like signs etc.). The amount of compression and re-phrasing is different between various languages, also dependent on cultural differences and subtitle traditions. A special type are subtitles for the hearing impaired which are closer to literal transcriptions combined with extra information about other sounds (such as background noise etc.). All this causes many insertions, deletions and complex mappings when aligning subtitles. Some of the challenges are illustrated in figure 1.

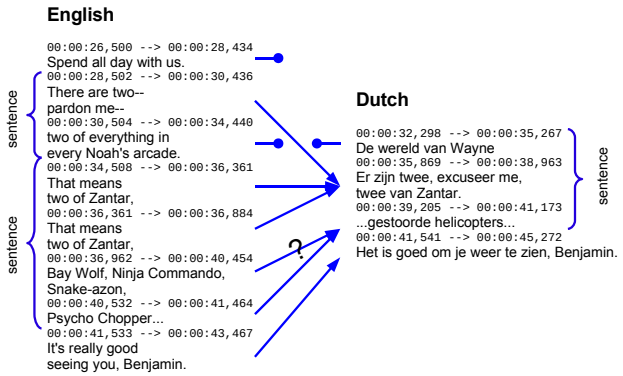

Fig. 1: *Alignment challenges: An example with English and Dutch subtitles.*

The figure shows a short example of English subtitles and their Dutch correspondences. There are untranslated segments such as the English fragments shown in subtitle screen one, three and six. The latter two are even embedded in surrounding sentences which makes it impossible to find a proper alignment with sentences as the basic unit. Furthermore, automatic tokenization and sentence splitting causes further errors. Obviously, sentences may span over several subtitle screens as illustrated in figure 1. However, in the Dutch example the first subtitle line is attached to the proceeding ones because the sentence splitter did not recognize a proper sentence boundary between line one and two. A sentence aligner has no other chance then to link the entire unit to corresponding ones in the other language even if the mapping is only partially correct. We can also see in the example that there is only one real 1:1 alignment whereas other types are more frequent than in other parallel resources.

From the discussion above, it seems obvious that traditional sentence alignment approaches are not ap-

propriate for this kind of data. Hence, we propose a new approach specifically designed for the alignment of subtitles. However, in the next section we firstly present the subtitle corpus we have collected including a brief discussion about pre-processing issues.

2 The Subtitle Corpus

Several databases are on-line that provide subtitles in various languages. All of them collect user uploads that can be searched in various ways. However, most of them are not very stable in the sense that they move to different locations and have a lot of down-time. This made us suspicious about their legal background. We found one provider, `http://www.opensubtitles.org`, that seems to be very reliable, which offers an extensive multilingual collection of subtitles without user registration necessary. They claim that their database only contains legal downloads that are free to distribute. Furthermore, we were pleased to obtain the entire database of about 308,000 files by the provider covering about 18,900 movies in 59 languages (status of July, 2006) for which we are very grateful.

In order to build our corpus, several pre-processing steps had to be taken. First of all, we had to identify the subtitle format and to convert it to a uniform corpus format. Several formats are used and we decided to support two popular ones, SubRip files (usually with extension '.srt') and microDVD files (usually with extension '.sub'). The latter were automatically converted to SubRip using a freely available script sub2srt (`http://www.robelix.com/sub2srt/`). Furthermore, subtitles use various character encodings. Unfortunately, we are not aware of a reliable tool for detecting character encodings and, therefore, we manually defined a conversion table (one encoding per language) after inspecting some sample data. We converted the subtitle files to a simple standalone XML format using Unicode UTF-8. An example is shown in figure 2.

Each subtitle file has been tokenized and marked with sentence boundaries as shown in figure 2. Both, tokenization and sentence splitting is done by means of regular expressions. The annotation is done automatically without any manual corrections and, therefore, contains errors especially for languages that do not use similar word and sentence boundaries as defined in our patterns. In future work, we would like to improve tokenization and especially sentence boundary detection which is crucial for the success of an alignment at the sentence level[1].

Another issue with the database we obtained is that it contains erroneous files, for example, files with corrupt character encodings and subtitles tagged with the wrong language. In order to remove such noise as much as possible, we included a language classifier to check the contents of all subtitles. For this we used

[1] Note that sentences may span several subtitle screens as also shown in figure 2. This makes it necessary to store time information (which we need for the alignment later on) in a special way to avoid crossing annotations that are not allowed in XML. Hence, *time slot* information is split into two time events, one for the starting time and one for the end of the slot.

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <s id="1">
    <time id="T1S" value="00:00:26,500" />
    <w id="1.1">Spend</w>
    <w id="1.2">all</w>
    <w id="1.3">day</w>
    <w id="1.4">with</w>
    <w id="1.5">us</w>
    <w id="1.6">.</w>
    <time id="T1E" value="00:00:28,434" />
  </s>
  <s id="2">
    <time id="T2S" value="00:00:28,502" />
    <w id="2.1">There</w>
    <w id="2.2">are</w>
    <w id="2.3">two</w>
    <w id="2.4">--</w>
    <w id="2.5">pardon</w>
    <w id="2.6">me</w>
    <w id="2.7">--</w>
    <time id="T2E" value="00:00:30,436" />
    <time id="T3S" value="00:00:30,504" />
    <w id="2.8">two</w>
    <w id="2.9">of</w>
    <w id="2.10">everything</w>
    <w id="2.11">in</w>
    <w id="2.12">every</w>
    <w id="2.13">Noah'</w>
    <w id="2.14">s</w>
    <w id="2.15">arcade</w>
    <w id="2.16">.</w>
    <time id="T3E" value="00:00:34,440" />
  </s>
```

Fig. 2: *Subtitles in XML*

`textcat` a freely available and trainable classifier designed for language identification [13]. It uses N-gram models trained on example texts and, therefore, relies on the given encoding used in the training data. We applied the language checker after encoding conversion and, therefore, built language models for UTF-8 texts. For simplicity we used the training data from the `textcat` package converted to Unicode using the Unix tool `recode`. Altogether, we created 46 language models. The classifier predicts for each given input file the most likely language according to the known models. The output of `textcat` is one of the following: (1) a certain classification of one language, (2) a ranked list of likely languages (in cases where the decision is not clear-cut), and, (3) a "resign" message in cases where the language classifier does not find any language that matches sufficiently enough. We accepted subtitles only in the case where the language classifier is certain that the language is the same as specified in the database and disregarded all other files.

After pre-processing and language checking we retained 38,825 subtitle files in 29 languages. From that we selected 22,794 pairs of subtitles for alignment (selecting only the ones corresponding to the same physical video file) covering 2,780 movies in 361 language pairs. Altogether, this corresponds to about 22 million sentence alignments created by the approach described below.

3 Sentence alignment

One of the essential properties of parallel corpora is that they can be aligned at some segmentation level. A common segmentation is to split on sentence boundaries and to link sentences or sequences of sentences in the source language with corresponding ones in the target language. Sentence alignment is assumed to be

monotonic, i.e. crossing links are not allowed. However, deletions and insertions are usually supported.

Subtitles can be aligned at various segmentation levels, for instance, mapping *subtitle screens* (text fragments shown together in one time slot on screen) or *sentences*. We opted for the latter for the following reasons: Sentences are linguistically motivated units and important for applications using the aligned data. Subtitle screens on the other hand often include various fragments by different speakers and their compilation highly depends on visual requirements and language dependent issues. The contents of these screens varies very much between different subtitles and, therefore, they are hard to align without partial overlaps with other screens. We therefore decided to align the data at the sentence level assuming that our sentence splitter works well for most of the languages included.

In the following, we first discuss a standard length-based approach applied to subtitles. Thereafter, we will present our new alignment approach based on time overlaps. Finally, some additional heuristics are discussed for further improvements.

3.1 Length-based approaches

One of the standard approaches to sentence alignment is the popular length-based approach proposed by [4]. It is based on the assumption that translations tend to be of similar lengths in characters (possibly factorized by a specific constant) with some variance. Using this assumption we can apply a dynamic algorithm to find the best alignment between sentences in one language and sentences in the other language. Alignments are restricted to the most common types (usually 1:1, 1:0, 0:1, 2:1, 1:2 and 2:2) with prior probabilities attached to them to make the algorithm more efficient and more accurate. In the default settings, there is a strong preference for 1:1 sentence alignments whereas the likelihood of the other types is very low. This is based on empirical studies of some example data [4].

It has been shown that this algorithm is very flexible and robust even without changing its parameters [12, 11]. However, looking at our data it is obvious that certain settings and assumptions of the algorithm are not appropriate. As discussed earlier, we can observe many insertions and deletions in subtitle pairs and typically, a length-based approach cannot deal with such cases very well. Even worse, such insertions and deletions may cause a lot of follow-up errors due to the dynamic algorithm trying to cover the entire text in both languages. In order to account for the special properties of subtitles we adjusted the prior probabilities set in the length-based alignment approach. For this we manually aligned a small subset of randomly selected subtitles from five movies in English, German, and Swedish. We aligned parts of all language combinations using the interactive sentence alignment tool ISA [10] resulting in a total of 1312 sentence alignment units. We used relative frequencies of each occurring alignment type to estimate the new parameters. For efficency reasons we omitted alignment types with probabilities below 0.001. Table 1 lists the final settings used for the length-based approach.

Using the settings above, 1:1 sentence alignment are still preferred but with a smaller likelihood (0.89 in the original settings). As expected, deletions and insertions (1:0 and 0:1 alignments) are more frequent in subtitles (0.0099 each in the original implementation) and two types are added: 1:3 and 3:1 alignments. On the other hand, 2:2 alignments are not considered in our model whereas they are in the original approach with a prior probability of 0.011). We are aware of the fact that there is a substantial variance among alignment types (depending on the language pair and other factors) and that our sample is not representative for the entire collection containing many more language pairs. However, we assume that these settings are still more appropriate than the default settings used in the original algorithm. Figure 3 shows example output of the approach with adjusted parameters.

alignment type	count	probability
1:1	896	0.6829
2:1	100	0.0762
0:1	91	0.0694
1:0	74	0.0564
1:2	72	0.0549
1:3	24	0.0183
3:1	16	0.0122

Table 1: *Adjusted priors for various alignment types (with probability > 0.001)*

English	Dutch
Spend all day with us . **There are two – pardon me – two of everything in every Noah' s arcade .**	*De wereld van Wayne* **Er zijn twee , excuseer me ,** *twee van Zantar . ... gestoorde helicopters ...*
That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snake- azon , Psycho Chopper ...	*Het is goed om je weer te zien , Benjamin . Je bent al heel lang niet meer in Shakey' s geweest .*
It' s really good seeing you , Benjamin .	*Ik heb het heel erg druk .*
You haven' t been into Shakey' s for so long .	*Het zijn er twee voor jou , want eentje zal het niet doen .*
Well , I' ve been real busy . It' s two for you ' cause one won' t do .	*De hele week , krijgen kinderen onder de zes elke vijfde ...*
All this week , kids under 6 get every fifth – **There' s a new pet .**	**Er is een nieuw huisdier** *Het Chia huisdier .*
Ch- Ch- Chia Chia Pet – **the pottery that grows .**	*Het aardewerk dat groeit .*
They are very fast .	**Zij zijn erg snel .**
Simple .	**Simpel .**
Plug it in , and insert the plug from just about anything .	**Plug het in .**

Fig. 3: *Length-based sentence alignment - text in italics is wrongly aligned.*

As the figure illustrates there are many erroneous alignments using the length-based approach. In fact, most of the alignments are wrong (in italics) and we can also see the typical problem of follow-up errors. For example, the alignment is shifted already in the beginning due to the deletion of some sentences fragments in Dutch.

3.2 Alignment with time overlaps

As seen in the previous sections, a length-based approach cannot deal very well with our data collec-

tion. Let us now consider a different approach directly incorporating the time information given in the subtitles. Subtitles should be synchronized with the original movie using the time values specified for each screen. Intuitively, corresponding segments in different translations should be shown at roughly the same time. Hence, we can use this information to map source language segments to target language segments. The timing is usually not exactly the same but the overlap in time in which they are shown should be a good predictor for correspondence. The main principle of an alignment approach based on time overlaps is illustrated in figure 4.

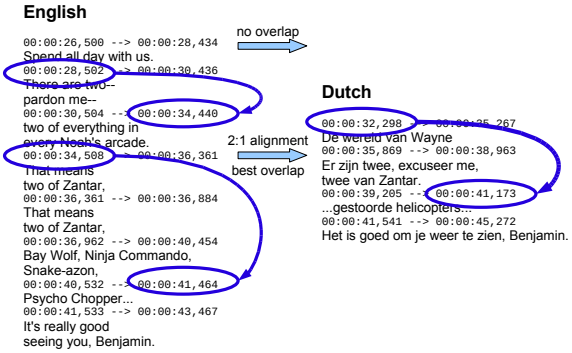

Fig. 4: *Sentence alignment with time overlaps*

The main problem with this approach is to deal with the differences in dividing texts into screens in various languages. The alignment is still done at the sentence level and, hence, we need time information for sentences instead of subtitle screens. Figure 4 illustrates some simple cases where sentences span over several screens but still start and end at screen boundaries. However, this is not always the case. Very often sentence boundaries are somewhere in the middle of a screen (even if they span over several screens) and, hence, start and end time are not explicitly given. In these cases we have to approximate the time boundaries to calculate overlaps between sentences in corresponding files. For this we used the nearest "time events" and calculated the time proportional to the strings in between. Computing a new time event t_{new} for a sentence boundary in this way is given by the following equation:

$$t_{new} = t_{before} + c_{before} * \frac{t_{after} - t_{before}}{c_{before} + c_{after}}$$

Here, t_{before} corresponds to the nearest time event before the current position and t_{after} is the time at the nearest time event after the current position. Similarly, c_{before} and c_{after} are the lengths of the strings before and after the current position up to the nearest time events. Hence, we interpolate the time linearly over the characters in the current segment. This is done dynamically from the beginning to the end of the subtitle file using approximated time values as well for further time estimations if necessary (in cases where more than one sentence boundary is found within one subtitle screen). Additionally, consistency of the time values is checked. Due to errors in the subtitle files it can happen that time events have identical or even decreasing values. In these cases a dummy time of 0.0001 seconds is added to the previous time event overwriting the inconsistent one. This is done iteratively as long as necessary.

Now, with time values fixed for all sentences in the subtitle we need to find the best alignment between them. We still want to support deletions, insertions and n:m alignments. In our approach, we define a set of possible alignment types (as in the length-based approach) which are then considered as possible alternatives when looking for the best mapping. In our experiments, we simply applied the same types as used in the length-based approach (see table 1). However, prior probabilities are not used in this model. The comparison is purely based on absolute time overlaps. The algorithm runs through the pair of subtitles in a sliding window, comparing alternative alignments according to the pre-defined types and picking the one with the highest time overlap. Note that we do not need any recursion and the alignment can be done in linear time because of the use of absolute time values. The result of the alignment with time overlaps for our little example is shown in figure 5.

English	Dutch
Spend all day with us .	
There are two – pardon me – two of everything in every Noah' s arcade . That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snake- azon , Psycho Chopper ...	*De wereld van Wayne* **Er zijn twee , excuseer me , twee van Zantar . ... gestoorde helicopters ...**
It' s really good seeing you , Benjamin .	**Het is goed om je weer te zien , Benjamin .**
You haven' t been into Shakey' s for so long .	**Je bent al heel lang niet meer in Shakey' s geweest .**
Well , I' ve been real busy .	**Ik heb het heel erg druk .**
It' s two for you ' cause one won' t do .	**Het zijn er twee voor jou , want eentje zal het niet doen .**
All this week , kids under 6 get every fifth – There' s a new pet .	**De hele week , krijgen kinderen onder de zes elke vijfde ... Er is een nieuw huisdier** *Het Chia huisdier .*
Ch- Ch- Chia Chia Pet – **the pottery that grows .**	**Het aardewerk dat groeit .**
They are very fast .	**Zij zijn erg snel .**
Simple .	
Simpel . **Plug it in , and insert the plug from just about anything .**	**Plug het in .** *Het is simple !*

Fig. 5: *Sentence alignment based on time overlaps - text in italics is wrongly aligned.*

One of the big advantages of this approach is that it can easily handle insertions and deletions at any position as long as the timing is synchronized between the two subtitle files. Especially initial and final insertions often cause follow-up errors in length-based approaches but they do not cause any trouble in the time overlap approach (look for example at the first English sentence in the example in figure 4). Remaining errors mainly occur due to sentence splitting errors and timing differences. The latter will be discussed in the following section.

3.3 Movie synchronization

Intuitively, the alignment approach based on time overlaps ought to produce very accurate results assuming that subtitles are equally synchronized to the original movie. Surprisingly, this is not always the case. Time information in subtitles often varies slightly resulting in growing time gaps between corresponding segments. In preliminary evaluations we realized that the alignments produced by the time overlap approach either is very accurate or very poor. After inspecting some problematic cases it became obvious that the errors where due to timing issues: a difference in speed and a difference in starting times. Considering the fact that alignment is entirely based on time information already small timing differences have a large impact on this approach.

Fortunately, timing differences can be adjusted. Assuming that speed is constant in both subtitles we can compute two additional parameters, speed difference (*time ratio*) and *time offset* using two anchor points that correspond to true alignment points. The following equations are used to calculate the two parameters:

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

Here, src_1 and src_2 corresponds to the time values (in seconds) of the anchor points in the source language and trg_1 and trg_2 to the time values of corresponding points in the target language. Using $time_{ratio}$ and $time_{offset}$ we adjust all time values in the source language file and align sentences using the time overlap approach.

The time synchronization approach described above is very effective and yields significant improvements where timing differences occur. However, it requires two reliable anchor points that should also be far away from each other to produce accurate parameter estimations. One approach (and the most reliable one) is to define these anchor points manually. Again, ISA can be used to do this job simply by adding two break points to the subtitle pair, one at the beginning and one at the end. We then use the times at the beginning of each break point and synchronize. This approach is simple and requires minimal human intervention. However, it is not feasible to use it for all subtitle pairs in our corpus.

An alternative approach is to restrict human intervention to cases where erroneous alignments can be predicted using some simple heuristics. For example, we can count the ratio between empty sentence links (1:0 and 0:1) and non-empty ones.

$$algtype_{ratio} = \frac{|\text{non-empty links}| + 1}{|\text{empty links}| + 1}$$

Assuming that an alignment should mainly consist of non-empty links we can use a threshold for this ratio (for example > 2.0) to decide whether an alignment is likely to be correct or not. The latter can be inspected by humans and corrected using the anchor point approach.

Another approach for synchronization is to use cognates in form of similar strings to identify corresponding points in source and target language. For this, subtitle pairs are scanned in a sliding window from the beginning and from the end in order to find appropriate candidates. Using string similarity measures such as the longest common subsequence ratio (LCSR) and thresholds on similarity we can decide for the most relevant candidate pairs with the largest distance (it is also advisable to set a threshold for the minimal length of a possible candidate). Alternatively, we can restrict the search to identical strings and/or to strings with initial capital letters or we may include pairs from a given bilingual dictionary to find anchor points in the subtitle pairs. Note that a candidate does not have to be limited to a single word. Using these pairs of corresponding candidates we can use the time (start or end) of the sentences they appear in to compute the timing differences. Clearly, the cognate approach is restricted to related languages with more or less identical character sets. A solution for more distant language pairs would be to use existing bilingual dictionaries to select appropriate candidate pairs. This, however, requires corresponding resources for all language pairs included which are not available to us.

Furthermore, selecting candidate pairs is not straightforward especially in our subtitle data. Names are often spelled in a similar way in different languages and therefore, they will frequently be selected as anchor point by the string similarity measure. However, the use of names may differ significantly in various languages. As we discussed earlier, subtitles are not transcriptions of the spoken data and, hence, names are often left out or replaced by referring expressions. Therefore, we may find a lot of false hits when using a general search for cognate pairs. In our initial experiments we observed that a general synchronization based on cognate pairs for all subtitle pairs is harmful for the overall alignment quality. Hence, heuristics based on alignment type ratios as mentioned above are again useful for selecting potentially erroneously aligned subtitle pairs for which synchronization might be useful. Another strategy to reduce synchronization errors made by wrongly selected anchor points is to average over all candidate pairs. However, this can lead to other errors. Finally, we can also try all possible combinations of anchor point candidates and use them iteratively for synchronization. We then pick the one that performs best according to the alignment type ratio as defined above. Fortunately, the time overlap approach is fast enough to make it feasible to apply this approach (see section 4 below).

4 Evaluation

For evaluation we randomly selected 10 movies with subtitles in three languages, English, German and Dutch. We manually aligned parts of all pairs of Dutch-English and Dutch-German subtitles from this set using ISA[2]. In particular, we selected about 15 initial, 15 intermediate and 15 final sentences in each

[2] Note that the alignments are symmetric and the direction of the alignment as mentioned here is only due to alphabetic sorting of the language name

aligned subtitle pair to account for differences in alignment quality at different document positions. The exact number of sentences aligned varies slightly between all subtitle pairs due to the amount of insertion, deletions and n:m alignments necessary. In total, we included 988 alignment units in our evaluation set, 516 for Dutch-English and 472 for Dutch-German. The 10 movies are all originally in English and, therefore, it is interesting to compare the alignments for the two selected language pairs. English subtitles are mainly produced for the hearing impaired and, therefore, contain much more information than the two translations into Dutch and German. However, let us first look at the overall accuracy of the alignment for the following four alignment approaches: (1) length-based sentence alignment with adjusted priors (*length*), (2) standard time-overlap alignment (*time1*), (3) time-overlap alignment with a cognate filter (LCSR) using a threshold of 0.8 which is applied in cases where $algtype_{ratio} < 2.0$ (*time2*), and, (4) time-overlap alignment with a cognate filter (threshold=0.6) and iterative selection of candidate pairs according to the alignment type ratio in cases the initial ratio is < 2.0 (*time3*). The minimal string length for the cognate filter is set to five characters for both, time2 and time3. The LCSR threshold for *time3* is lower than for *time2* to give it more flexibility when selecting anchor points. It is not recommendable to use such a relaxed threshold for *time2* because of the risk of finding false positives. *Time2* automatically selects the candidate pairs with the largest distance from each other and, therefore, the probability of selecting a wrong candidate pair is larger with lower thresholds for the cognate filter.

The results of the alignments measured on our evaluation data are shown in table 2. The scores are split into three categories: *correct* for exact matches, *partial* for partially correct alignments (some overlap with correct alignments in both, source and target language[3]), and *wrong* for all other alignments. Naturally, we count only scores for sentences included in the manually aligned data.

approach	correct	partial	wrong
length	0.515	0.119	0.365
time1	0.608	0.105	0.286
time2	0.672	0.136	0.192
time3	0.732	0.144	0.124

Table 2: *Different alignment approaches*

The scores in table 2 show that the alignment accuracy is significantly lower than otherwise reported for sentence alignment, which could be expected due to the difficulties in our data discussed earlier. However, the time-overlap approach yields major improvements compared to the length-based alignment. We can also see that the heuristics for enabling synchronization based on the type ratio is successful. The final approach using iterative candidate selection clearly outperforms the others. It is interesting to see where the strengths of the time overlap approach can be found. For this, we computed accuracy scores for the different alignment types (see table 3). In order to make it easier to compare the results we counted partially correct links as 50% correct and added them accordingly to the scores of the correct links.

type	nr	length	time1	time2	time3
1:1	685	0.734	0.676	0.763	0.842
0:1	106	0.000	0.566	0.575	0.594
1:2	70	0.429	0.529	0.671	0.743
1:0	52	0.000	0.904	0.923	0.885
2:1	43	0.535	0.686	0.791	0.849
1:3	16	0.469	0.594	0.625	0.688
2:2	5	0.300	0.300	0.400	0.400
3:1	3	0.333	0.667	0.833	0.833

Table 3: *Accuracy per alignment type (skipping 8 alignments with more than four sentences involved)*

The strength of the time-overlap approach is certainly in the non-1:1 alignments. The length-based approach is rather good in finding proper 1:1 links and yields even better results than the standard time-overlap approach. However, using synchronization heuristics brings about a significant improvement beyond the accuracy of the baseline approach even for this alignment type. The largest difference can be seen in the empty alignments. The time-overlap approach can handle insertions and deletions much better than the length-based approach. It also yields better results for the other types. Synchronization is helpful for almost all types. One exception is the score for 1:0 alignments which actually drops a little bit when applying the iterative anchor point selection. A reason for this is that such empty alignments are taken as indicators for erroneous alignments even when they are correct. In some cases this assumption causes a degradation of performance. This can also be seen when looking at the results for the individual subtitle pairs (tables 4 and 5).

movie (*dut-eng subtitles*)	length	time1	time2	time3
Location Production Footage: The Last Temptation of Christ	0.857	0.976	0.976	0.976
Finding Neverland	0.375	0.333	0.333	0.615
A Beautiful Mind	0.339	0.688	0.688	0.688
Under Fire	0.896	0.896	0.896	0.896
Batman Forever	0.976	0.988	0.988	0.988
The Last Samurai	0.043	0.928	0.942	0.928
Basic	0.737	1.000	1.000	1.000
Pulp Fiction	0.088	0.175	0.175	0.579
Return to Paradise	0.640	0.940	0.940	0.940
The Diary of Anne Frank	0.308	0.754	0.754	0.754
average (516 alignments)	0.476	0.754	0.756	0.825

Table 4: *Alignment accuracy per subtitle pair for Dutch-English*

There are indeed subtitle pairs for which the accuracy drops when using iterative anchor point selection as mentioned above. However, the overall performance is higher using this strategy compared to the fixed selection of the most distance candidates (time2). Tables 4 and 5 also include average accuracies per language pair. Here, we can observe a huge difference between the accuracy of the length-based ap-

[3] Note that empty alignments are always mapped to 1:0 or 0:1 alignments (never 0:x or x:0 with x>1) and are either correct or wrong but never *partial*.

movie (*dut-ger subtitles*)	length	time1	time2	time3
Location Production Footage: The Last Temptation of Christ	0.963	1.000	1.000	1.000
Finding Neverland	0.523	0.047	0.791	0.756
A Beautiful Mind	0.796	0.092	0.663	0.643
Under Fire	0.890	0.900	0.900	0.900
Batman Forever	0.608	0.706	0.706	0.706
The Last Samurai	0.787	0.915	0.915	0.915
Basic	0.500	0.154	0.590	0.487
Pulp Fiction	0.637	0.049	0.049	0.716
Return to Paradise	0.798	0.915	0.915	0.915
The Diary of Anne Frank	0.361	0.759	0.759	0.759
average (472 alignments)	0.683	0.559	0.722	0.781

Table 5: *Alignment accuracy per subtitle pair for Dutch-German*

proach which is much higher for Dutch-German than for Dutch-English. This could also be expected due to the differences in style. The English subtitles are much more detailed whereas German and Dutch subtitles are more compressed. It looks like that this compression is rather similar for the two languages which favors the length-based approach. The time-overlap alignment is actually a bit worse for Dutch-German than for Dutch-English but this might be rather incidental. Many errors are due to sentence splitting mistakes which can be quite different for the various subtitles. In future work, preprocessing should be improved to reduce errors originating in tokenization and sentence splitting. Fortunately, the alignment is done automatically and, therefore, can easily be re-run after any preprocessing improvement. Another task for future work is to check the alignment quality for other language pairs especially more distant ones for which the synchronization approach using cognates is not applicable.

5 Conclusions

In this paper a sentence alignment approach for movie subtitles based on time overlaps has been introduced. It has been used to align an extensive multilingual corpus of about 38,000 subtitles in 29 languages. Its accuracy outperforms standard length-based alignment approaches especially by improving non-1:1 alignments that frequently occur in this kind of data. Furthermore, we presented additional techniques to synchronize subtitles to improve the alignment even further.

References

[1] P. F. Brown, J. C. Lai, and R. L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29st Annual Meeting of the ACL*, pages 169–176, 1991.

[2] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The Mathematics of Statistcal Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993.

[3] S. F. Chen. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1993. Association for Computational Linguistics.

[4] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.

[5] M. Kay and M. Röscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, 1993.

[6] I. D. Melamed. Bitext Maps and Alignment via Pattern Recognition. *Computational Linguistics*, 25(1):107–130, 1999.

[7] R. C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144, London, UK, 2002. Springer-Verlag.

[8] M. Simard, G. F. Foster, and P. Isabelle. 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, 1992.

[9] F. Smadja, K. R. McKeown, and V. Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics, 22(1)*, 1996.

[10] J. Tiedemann. 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, 2006.

[11] J. Tiedemann and L. Nygard. The OPUS corpus - parallel and free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal, 2004.

[12] E. F. Tjong Kim Sang. Aligning the Scania Corpus. Technical report, Department of Linguistics, University of Uppsala, 1996.

[13] G. van Noord. Textcat – implementation of the algorithm presented in Cavnar, W. B. and J. M. Trenkle, "N-Gram-Based Text Categorization" In Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, UNLV Publications/Reprographics, pp. 161-175, 1994. http://www.let.rug.nl/∼vannoord/TextCat/, 2006.

[14] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing*, pages 590 – 596, 2005.