10

Improved Sentence Alignment for Building a Parallel Subtitle Corpus

Building a Multilingual Parallel Subtitle Corpus

Jörg Tiedemann
University of Groningen

Abstract

In this paper on-going work of creating an extensive multilingual parallel corpus of movie subtitles is presented. The corpus currently contains roughly 23,000 pairs of aligned subtitles covering about 2,700 movies in 29 languages. Subtitles mainly consist of transcribed speech, sometimes in a very condensed way. Insertions, deletions and paraphrases are very frequent which makes them a challenging data set to work with especially when applying automatic sentence alignment. Standard alignment approaches rely on translation consistency either in terms of length or term translations or a combination of both. In the paper, we show that these approaches are not applicable for subtitles and we propose a new alignment approach based on time overlaps specifically designed for subtitles. In our experiments we obtain a significant improvement of alignment accuracy compared to standard length-based approaches.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

10.1 Introduction

The value of parallel corpora has been shown in various NLP applications and research disciplines. Some of them are data-driven machine translation (Brown et al. 1993, Brown 1996), multilingual lexicon/terminology extraction (Gale and Church 1991, Smadja et al. 1996, Hiemstra 1998, Gaussier 1998, Tufis and Barbu 2001), word sense disambiguation (Ide 2000, Diab and Resnik 2002) and general translation studies (Johansson 2002) to mention just a few. However, in contrast to monolingual language corpora there are still only a few parallel corpora available especially ones containing more than two languages. Often they originate from specialized domains such as legislation and administration or technical documentation and cover only a few “high density” languages. On the other hand, the amount of translated documents is increasing on the Internet even for lower density languages. In the past years several projects working on the collection of multilingual material from the web have been reported (Resnik 1999, Tiedemann and Nygard 2004).

One of the fastest growing multilingual resources are on-line databases of movie subtitles. There is a huge demand for subtitles on the Internet and users provide them to others in various languages via download services on-line. They are available in form of plain text files for modern as well as for classical movies and they are usually tagged with extra information such as language, genre, release year, user ratings and download counts. Subtitles are different to other parallel resources in various aspects: Most of them are (at least close to) transcriptions of spontaneous speech. They include plenty of idiomatic expressions and slang. They can easily be divided into different genres and time periods. There are even different subtitles versions (in the same language) for the same movie. Translations are usually very free and show a lot of cultural differences. They are aligned to the original movie and can therefore be linked to the actual sound signals. However, subtitles often summarize spoken utterances instead of completely transcribing them. Hence, they can also be used to study text compression and summarization (Daelemans et al. 2004). To summarize the discussion, subtitle databases provide a unique multilingual resource with various kinds of valuable information encoded in the texts.

In the following we will concentrate on building a parallel subtitles corpus from one on-line resource. In particular, we obtained the entire database of about 308,000 files from `http://www.opensubtitles.org`, a free on-line collection of movie subtitles in many languages. We are very grateful for the support by the providers of this website.

The paper is focused on the alignment of sentences and sentence fragments which is an essential step for building a parallel corpus. However, in the next section we first discuss necessary pre-processing steps to clean up the original database and to convert subtitle files into XML-based corpus files. Thereafter, we describe the details of the sentence alignment approaches in detail applied to our data. Finally, we present evaluations of the automatic alignment and provide some conclusions and prospects for future work.

10.2 Pre-processing

Pre-processing the original subtitle files is necessary because of several reasons: First of all, the database entirely consists of user uploads and, therefore, the content is not as clean as we want it to be. For example, movies are sometimes not tagged with the correct language, they are encoded in various character encodings, and they come in various formats. In our corpus we require a consistent format in a uniform encoding. In particular, we decided to use a simple standalone XML format and Unicode UTF-8. A sample output after all steps including tokenization is shown on the right-hand side of figure 10.1. Pre-processing consists of the following steps:

Subtitle format detection & conversion: We accepted two popular subtitle formats: SubRip files (usually with extension '.srt') and microDVD subtitle files (usually with extension '.sub'). For the conversion to XML we relied on the SubRip format which is more frequently used in the database we got. An example is shown in the left part of figure 10.1. microDVD subtitle files were converted to SubRip format using a freely available script sub2srt (`http://www.robelix.com/sub2srt/`).

Removing doubles: The database contains a lot of repeated subtitles; i.e. created for the same movie in the same language. We simply took the first one in the database and dismissed all the others. In future, we like to investigate possible improvements by other selection principles, e.g. taking download counts or user ratings into account.

Character encoding: All files are converted to Unicode UTF-8 to have a uniform encoding throughout all data files. This is especially useful when working with aligned data where several languages have to be put together. Unfortunately, we are not aware of a reliable classifier for automatic detection of character encodings and, therefore, we manually defined an incomplete encoding conversion table after inspecting sample data in various languages (see table 10.1).

Certainly, using a fixed language encoding table is only an ad-hoc solution causing errors in the conversion. However, using the language filter described below we remove most of the subtitles for which the encoding conversion failed. This, at least ensures high quality in our data as a trade-off for some quantity. In the future we would like to use an automatic classifier for better encoding detection of individual files.

Language Checking While processing the data we realized that many uploads are not correct and, for instance, contain text in a language different to the one specified. In order to filter them out we used an automatic classifier to check the language before accepting a subtitle file. For this we used `textcat` a freely available and trainable classifier designed for language identification (van Noord 2006).

```
00:00:26,500 --> 00:00:28,434
Spend all day with us.
00:00:28,502 --> 00:00:30,436
There are two--
pardon me--
00:00:30,504 --> 00:00:34,440
two of everything in
every Noah’s arcade.
00:00:34,508 --> 00:00:36,361
That means
two of Zantar,
00:00:36,361 --> 00:00:36,884
That means
two of Zantar,
00:00:36,962 --> 00:00:40,454
Bay Wolf, Ninja Commando,
Snake-azon,
00:00:40,532 --> 00:00:41,464
Psycho Chopper...
00:00:41,533 --> 00:00:43,467
It’s really good
seeing you, Benjamin.
```

```
<?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>
```

Figure 10.1: A short segment of English subtitles of the movie “Wayne’s World” from 1993 in SubRip (.srt) format (left) and a tokenized XML version of the first two sentences (right).

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 UTF-8 by means of the free 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.

Tokenization and sentence splitting: We used simple regular expressions for tokenization and sentence splitting. Tokenization of subtitles is a challenging task. First of all, there are various languages in the corpus and both, tokenization and sentence splitting are highly language dependent. However, for most languages

Table 10.1: Character Encoding Table

encoding	languages (ISO-639 codes)
cp1250	alb, bos, cze, pol, rum, scc, scr, slv, hrv
cp1251	bul, mac, rus
cp1252	afr, bre, cat, dan, dut, epo, est, fin, fre, ger, hun, ita, nor, pob, pol, por, spa, swe
cp1253	ell, gre
cp1254	tur
cp1255	heb
cp1256	ara
cp1257	lat, lit
iso-8859-4	ice
big5-eten	chi
shiftjis	jpn
euc-kr	kor

tokenizers and sentence boundary detectors are not readily available. We opted for a general solution using patterns in terms of regular expressions. For this we used Unicode character classes hoping to cover various languages equally well. The following patterns where defined:

- split between a non-punctuation character and a punctuation that is followed by either space, another punctuation or end-of-string.
- split between a punctuation and a non-punctuation character if they are preceeded by either start-of-string, another punctuation symbol or a white-space character.
- split punctuation symbols if they are not identical (leaving, for example '...' intact)
- split on all white-space characters

Note, that subtitles may contain HTML-like tags for formatting issues (like <i> and <font ...>). These tags have to be treated in a special way to avoid their tokenization.

Sentence boundary detection is also done with general patterns due to the lack of available tools for all languages involved. An issue specific to our data is the fact that subtitles contain many sentence fragments instead of well-formed grammatical sentences. Hence, even more sophisticated sentence splitters available for some languages will fail in many cases.

Table 10.2 shows the basic patterns used for detecting sentence boundaries. For Chinese, Korean and Japanese we simply split at standard punctuation symbols ".!? :" which works to some extent but, of course, is by far not an optimal solution for these languages. Note that sentences may span over several screens and may also stop in the middle of a screen. Figure 10.1 shows an example sentence (with id=2) that spans over two screens. Hence, the patterns above are applied in a sliding window running through the data.

Table 10.2: Sentence splitting patterns using Perl regular expressions

pattern 1: split between “sentence end” and “sentence start”

sentence end:
`([ˆ.]\.|[!?:])[\′\"]?(\s*|\Z)`
(a dot following a non-dot OR one of the following punctuation symbols “!? :” possibly followed by single or double quotes and white-space characters)
sentence start:
`(\A|\s+)\-?\s*[\"\′]?[\p{N}\p{Ps}\p{Lu}]`
(one or more white-spaces possibly followed by a hyphen, possibly followed by white-space characters and quotes followed by either a number, an opening punctuation or an uppercase letter.)

pattern 2: split between “sentence end” and “sentence start”

sentence end:
`[.!?:][\"\′\]\}\)]?\-?(\s*|\Z)`
(one of “.!? :” possibly followed by quotes, a hyphen and white-spaces)
sentence start:
`(\A|\s+)[\"\′]?[\¿\¡\p{Lu}]`
(one or more spaces, possibly followed by quotes and either an inverted initial question/exclamation mark or an uppercase letter)

The reason for applying two separate patterns is to combine different types of evidence when making decisions about sentence boundaries. Pattern 1 has stronger end-of-sentence constraints (hyphens and multiple dots are not allowed) in combination with a slightly relaxed sentence-start constraint (allowing for example digits and opening brackets) whereas pattern 2 has stronger sentence-start constraints (no digits and opening brackets) but relaxed sentence-end constraints (allowing hyphens and some closing brackets).

Shortcomings of our simple pattern based approach are obvious but they work reasonably well for many subtitle files and languages. However, both, tokenization and sentence splitting have to be improved in future work. It is impossible to find a general solution especially if we want to keep the collection as open as possible in terms of languages and genres. One general cue for sentence splitting might come from the time tags. Long pauses between subtitles may help to determine sentence boundaries. This is (only) one reason why we like to keep the time information in our corpus[1].

[1]Note that we decided to encode time slots as two separate “time events”, one for the starting time and one for the end. In this way we can handle sentences and time slots that cross each other which would otherwise not be possible to encode in XML.

Selection for alignment: Of course, not all movies are covered in all languages. Furthermore, there are several versions of movies around (for instance various video encodings, movie splits, etc) and, hence, several versions of subtitles fitting specific movie files. In order to yield the highest alignment quality, we selected only those subtitles that have been produced for exactly the same physical movie file.

The original database we obtained contains 232,643 subtitles for 18,900 movies in 59 languages. After pre-processing and filtering as described above we were left with 38,825 subtitle files in 29 languages. From that we selected 22,794 pairs of subtitles for alignment covering 2,780 movies in 361 language pairs. Altogether, this corresponds to about 22 million sentence alignments.

10.3 Subtitle alignment

Essential for building a parallel corpus is the alignment at some segmentation level. At least two segmentation approaches are possible for our data: alignment of subtitle screens (time slot segmentation) and alignment of sentences (using the sentence boundaries detected in the pre-processing phase). We decided to use the latter for three reasons: First, sentence alignment is a well established task that usually yields high accuracy with language independent methods. Secondly, sentences are linguistically motivated units and, therefore, more suitable for further processing than subtitle fragments shown together on screen. Very often these fragments are not coherent units; for example they may come from various speakers in one scene. Finally, the format of subtitles is very different in various languages due to visibility constraints and cultural differences. There will be lots of partial overlaps when comparing the contents of subtitle screens across different languages. This makes it more difficult to align these units.

There are many challenges when aligning subtitle sentences as illustrated in figure 10.2.

Subtitles often contain summarized information instead of literal transcriptions or translations. Hence, we can observe a lot of insertions, deletions and paraphrases when comparing various translations. Furthermore, sentence splitting introduce errors that make it difficult to solve certain alignment problems. For example, in figure 10.2, the first three Dutch subtitle screens are marked as one sentence although the first one actually corresponds to the movie title that should not be connected to the following sentences (and which is not included in the English version of the subtitles). Furthermore, there are untranslated fragments such as the third and the sixth screen in English which are embedded in other sentences. However, sentences are treated as units and, therefore, the only solution is to align such fragments together with the surrounding ones even though they do not have corresponding fragments in the other language. From this little example it becomes obvious that we cannot expect the same quality of standard sentence alignment approaches as reported in the literature for other text types. Nevertheless, it is interesting to see how far we can get with standard approaches and how we can improve them for our purposes.

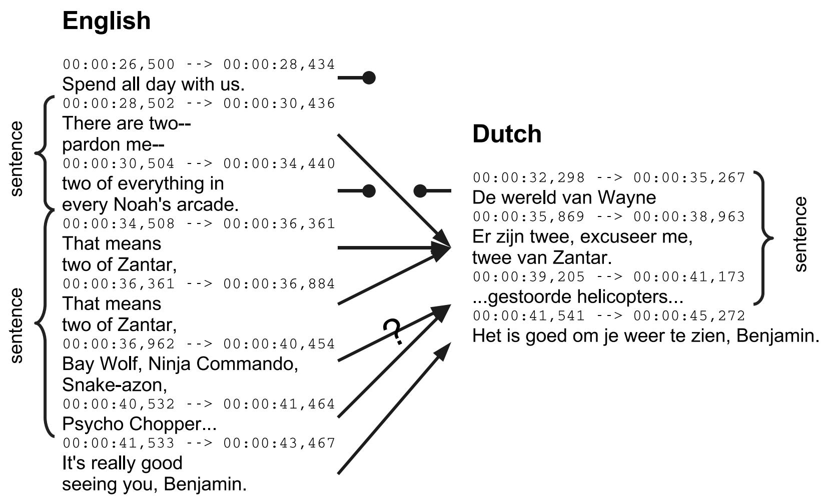

Figure 10.2: Alignment challenges: An example with English and Dutch subtitles.

10.3.1 Length-Based Sentence Alignment

There are several standard approaches to sentence alignment among them the popular length-based approach proposed by Gale and Church (1993). It is based on the assumption that translations tend to be of similar lengths (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. Another necessary assumption is that there are no crossing alignments (i.e. alignment is monotonic). Furthermore, alignments are restricted to certain types (usually 1:1, 1:0, 0:1, 2:1, 1:2 and 2:2) with prior probabilities attached to each type 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. These settings are based on empirical studies of some example data (Gale and Church 1993).

It has been shown that this algorithm is very flexible and robust even without changing its parameters (Tjong Kim Sang 1996, Tiedemann and Nygard 2004). However, looking at our data it is obvious that certain settings and assumptions of the algorithm are not appropriate. As discussed above, 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. Nevertheless, we applied this standard approach with its default settings to our data to study its performance. Figure 10.3 shows an example of the length-based alignment approach.

As the figure illustrates there are many erroneous alignments using the stan-

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 , Snakeazon , Psycho Chopper ...	*Het is goed om je weer te zien , Benjamin .*
It' s really good seeing you , Benjamin .	*Je bent al heel lang niet meer in Shakey' s geweest .*
You haven' t been into Shakey' s for so long .	*Ik heb het heel erg druk .*
Well , I' ve been real busy . **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 ...**
Ch- Ch- Chia Chia Pet – *the pottery that grows .*	*Er is een nieuw huisdier* **Het Chia huisdier .**
They are very fast . Simple .	*Het aardewerk dat groeit .*
Plug it in , and insert the plug from just about anything .	*Zij zijn erg snel .*
Simple .	*Simpel .*
Even for our customers in Waukegan , Elgin , and Aurora – We' ll be there right on time .	*Plug het in .*

Figure 10.3: Length-based sentence alignment - text in italics is wrongly aligned.

dard 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 in this example[2]. An obvious idea to improve the alignment quality is to optimize the parameters of the original alignment approach. For example, we might want to change the prior probabilities of alignment types supported by the algorithm. Furthermore, we might also want to include other alignment types that frequently occur in the data. However, such a tuning would have to be done for each language pair and even within one language pair it is questionable if these parameters will be consistent to a large degree.

A second idea is to use the time information given in the subtitle files. As we have discussed before, there are many insertions and deletions in various subtitles and therefore the correspondence between source and target language is often not so obvious in terms of sentence lengths. However, subtitles usually span the entire movie and, therefore, they cover more or less the same amount of time. Assuming that corresponding text segments are shown roughly at the same time we can use the *time length* of each slot (screen) instead of sentence length to match source and target language sentences. Here we have to interpolate between "time events" in cases where sentences do not start or end at time slot boundaries. For this, we used simple linear interpolation between the two nearest time events. Now, the same algorithm using dynamic programming can be used only with lengths in time instead of lengths in characters. Figure 10.4 shows the result of our example

[2]Note, that 'Simpel' in the end of the example is aligned to the wrong instance of 'Simple' in English.

movie when using time lengths for alignment.

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 , Snakeazon , Psycho Chopper ...	*Het is goed om je weer te zien , Benjamin .*
It' s really good seeing you , Benjamin .	*Je bent al heel lang niet meer in Shakey' s geweest .*
You haven' t been into Shakey' s for so long .	*Ik heb het heel erg druk .*
Well , I' ve been real busy .	*Het zijn er twee voor jou , want eentje zal het niet doen .*
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 .**

Figure 10.4: Sentence alignment based on time lengths - text in italics is wrongly aligned.

Unfortunately, the time length approach also produces a lot of errors. A striking difference is that in the end of the example the algorithm synchronizes well between source and target language which reduces the amount of follow-up errors from this point on. However, the accuracy is still unsatisfactory concluding from our first impressions. This will also be supported by our evaluations presented in section 10.4.

10.3.2 Alignment with Time Overlaps

As seen in the previous sections, length-based approaches cannot deal very well with our data collection. Let us now consider a different approach directly incorporating the time information given in the subtitles. The intuition in this approach is roughly the same as in the previous one based on time lengths: corresponding sentences are shown at roughly the same time because they should be synchronized with the movie. However, in the previous approach we only used the time length to match sentences but now we will directly use the absolute time values. Using start and end time for each sentence (using the same interpolation technique as before) we can measure the overlap between source and target language segments. We can now sequentially go through the subtitles and try to find segments with the highest overlap. This can be done efficiently in linear time without recursions because we use absolute times that cannot be shifted around. Again, we define a set of alignment types that we like to support in our alignment

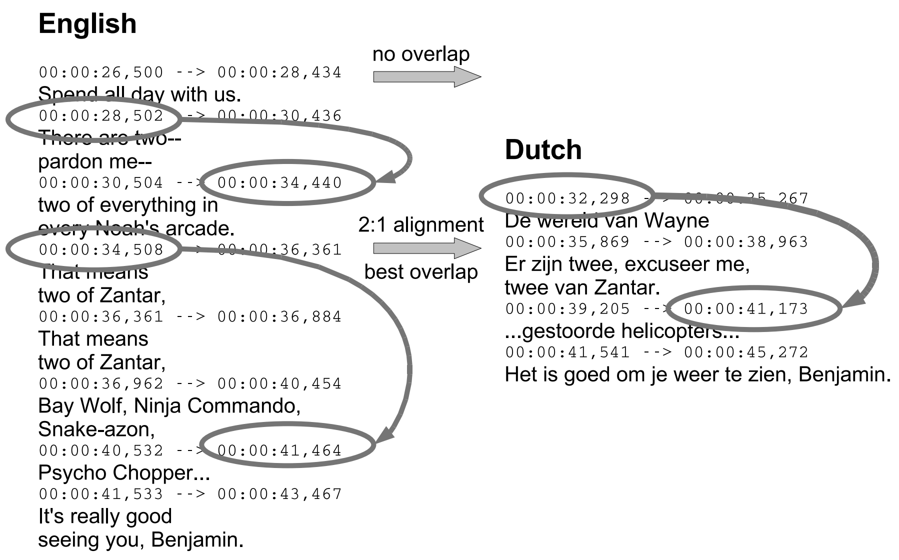

Figure 10.5: Sentence alignment with time overlaps

program. In particular, we use 1:1, 1:0, 0:1, 2:1, 1:2, 3:1, 1:3, 1:4, and 4:1 alignments[3]. Using these settings we can check possible alignments at each position and its surroundings and select the one with the highest overlap before moving to the next positions. The general principle of time overlap alignment is illustrated in figure 10.5.

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 10.5). Remaining errors mainly occur due to sentence splitting errors and timing differences. The latter will be discussed in the end of the following section. The result of the alignment with time overlaps for our example data is shown in figure 10.6.

10.4 Evaluation

In order to see the differences between the alignment approaches discussed above we manually evaluated a small sample of our aligned data. We selected two language pairs, Dutch-English and Dutch-German, and randomly selected five movies

[3]Note that the set of alignment types is different to the standard length-based sentence alignment approach. The impact of these additional types on the alignment quality has not been investigated.

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 !*

Figure 10.6: Sentence alignment based on time overlaps - text in italics is wrongly aligned.

for each of them[4]. In order to account for differences in alignment quality at different positions we selected 10 initial, 10 final sentence alignments, and 10 alignments in the middle of each document for each of the three alignment approaches. The evaluation was carried out by one human expert using the following three grades: correct, partially correct and wrong.

The overall result of our evaluation is shown in table 10.3[5].

As expected, the length-based approaches are much less accurate than the time-overlap approach. Surprisingly, the scores for alignments based on time lengths performs even worse than the standard sentence length based approach. The fact that Dutch-German performs much better for the time overlap approach should not be seen as a general tendency. The difference is due to the selection of movies which is different for the two language pairs. This is illustrated in table 10.4 showing the detailed scores per movie and language pair using the time overlap approach.

We can see that there are three movies that perform very poorly with our time-overlap approach, two in Dutch-English and one in Dutch-German. This explains the difference in total scores when comparing the two language pairs. The scores

[4]Each sub-corpus contains different movie pairs according to the subtitles available for the particular language pair. We did not want to restrict the selection to movies that have subtitles in all three languages. Hence, we obtained different sets of movies for both language pairs for our evaluation.

[5]We omit details over alignment positions. There are no striking differences in accuracies between initial, final and intermediate alignments with one exception: Time-length alignments performed much worse for the final sentence alignments than for the other ones. The reason for this is unclear to the author.

Table 10.3: Evaluation of alignment accuracy per alignment approach

alignment type	languages	correct	partially	wrong
sentence length	dut-eng	64.2%	9.2%	26.6%
sentence length	dut-ger	62.3%	12.3%	25.3%
time length	dut-eng	54.6%	6.9%	38.6%
time length	dut-ger	57.5%	9.8%	32.7%
time overlap	dut-eng	73.1%	8.7%	18.2%
time overlap	dut-ger	85.7%	6.8%	7.5%

Table 10.4: Evaluation of accuracy of time-overlap alignment per movie

languages	movie	correct	partially	wrong
dut-eng	Cube Zero	22.4%	14.3%	63.3%
dut-eng	Finding Nemo	36.8%	18.4%	44.7%
dut-eng	Grizzly Man	84.1%	12.7%	3.2%
dut-eng	Training Day	96.8%	3.2%	0.0%
dut-eng	Win a Date with Tad Hamilton	100.0%	0.0%	0.0%
dut-ger	Batman	92.6%	0.0%	7.4%
dut-ger	Cidade de deus	100.0%	0.0%	0.0%
dut-ger	Peggy Sue got married	82.8%	10.3%	6.9%
dut-ger	Rush Hour 2	93.5%	6.5%	0.0%
dut-ger	The Ring	33.3%	26.7%	40.0%

also show that the time-overlap approach either works very well (around or above 90% correct) or very poorly (below 40% correct). Here we see a clear effect of timing differences. If the timing is (only slightly) different for the two subtitle files to be aligned, the performance of the time-overlap approach drops dramatically. That means, if both subtitles are not synchronized very well with each other, almost everything goes wrong using the time-overlap approach whereas length-based approaches are not effected by this. These timing differences appear due to two factors: (1) the *subtitle speed* might be different, and, (2) the *time offset* for starting the subtitles might be different.

The solution to the problem mentioned above is to adjust the time values in one of the subtitles to synchronize it with the other one. In other words, we have to find the parameters for time offset and speed difference. In fact, appropriate values can easily be computed using two fixed anchor points at different positions in the movie, preferably far away from each other. Using the current time values at these fix-points it is a matter of simple maths to calculate offset and speed difference (or time ratio) assuming that the speed is constant in both subtitles.

The difficulty now is to find such reference points. One way is to add them by hand. We developed a simple tool for interactive sentence alignment, ISA (Tiedemann 2006), which can be used for this task. The tool allows to add break points at any place in subtitle pairs to be used for offset and speed calculation. Table 10.5 shows the results after manually adding such fix-points (one in the beginning and one at the end of the movie) to the three problematic movie pairs and re-aligning them after synchronization. The evaluation is done in the same way as before (using 10 initial, 10 final and 10 intermediate sentence alignments).

Table 10.5: Sentence alignment with speed and time offset adjustments using manual fix-points (*time ratio* refers to the speed difference and *offset* is the time offset in seconds).

movie	time ratio	offset	correct	partially	wrong
Cube Zero	0.9997	2.378	76.2%	9.5%	14.3%
Finding Nemo	0.9996	0.470	84.1%	4.5%	11.4%
The Ring	0.9589	0.302	100.0%	0.0%	0.0%

All alignments have been improved significantly with only very little human intervention. The speed and offset parameters fixed most of the alignment errors. Remaining errors are often due to little shifts in displaying subtitles and could easily be fixed using ISA as well. It is interesting to see that both parameters are very close to their default values (1 for time ratio and 0 for offset) but still have a significant impact on the alignment quality. It shows how brittle the time-overlap alignment approach is with respect to subtitle synchronization.

Although the manual intervention helped to improve the alignment quality significantly it is not reasonable to run the alignment in this way on the entire corpus with its more than 22,000 subtitle pairs. However, simple heuristics could be used to detect pairs for which an inspection would be desirable. For example, subtitle alignments with surprisingly many empty alignments (1:0 or 0:1) are likely to contain errors. They could be selected and presented to users via the ISA interface for validation.

Another technique to add break points for synchronization is to look for cognates in source and target language subtitles. Subtitles can be scanned from the beginning and from the end to find appropriate cognates. Simple string matching techniques and fixed thresholds can be used to spot candidate pairs. Assuming that they appear at the same position in the movie they can be used for calculating the two parameters necessary. However, experiments have shown that using such a technique in general decreases the overall alignment performance significantly. This is due to false hits where cognates are found but they do not refer to true reference positions. A reason for this is that names are often repeated in various contexts but not in the same way in all translations. Hence, correct matches but at non-corresponding positions frequently occur. However, cognate based techniques can again be combined with the same heuristics presented above: using this approach only in cases where the alignment seems to contain errors indicated by many empty alignments. This combined approach will be investigated in future work.

10.5 Conclusions

In this paper, a new multilingual parallel corpus consisting of movie subtitles in 29 languages has been presented. The corpus contains about 23,000 aligned subtitle pairs with altogether about 22 million sentence alignments. The data has been tokenized and sentences boundaries have been marked to be stored in a uniform XML format. We investigated three approaches to automatic sentence alignment, two based on length correspondence and a novel algorithm based on time overlaps. The latter yields significantly higher accuracies than traditional length-based alignment approaches. Remaining errors can be fixed to a large degree with minor human intervention. The corpus is available for research purposes and we will work on its extension in the future.

References

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D. and Mercer, R. L.(1993), The mathematics of statistcal machine translation: Parameter estimation, *Computational Linguistics* **19**(2), 263–311.

Brown, R. D.(1996), Example-based machine translation in the Pangloss system, *Proceedings of the 16th International Conference on Computational Linguistics, COLING-96*, Copenhagen, Denmark, pp. 169–174.

Daelemans, W., Höthker, A. and Tjong Kim Sang, E.(2004), Automatic sentence simplification for subtitling in Dutch and English, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal.

Diab, M. and Resnik, P.(2002), An unsupervised method for word sense tagging using parallel corpora, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia.

Gale, W. A. and Church, K. W.(1991), Identifying word correspondences in parallel texts, *Proceedings of the DARPA SNL Workshop*.

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

Gaussier, E.(1998), Flow network models for word alignment and terminology extraction from bilingual corpora, *Proceedings of COLING-ACL-98*, Montreal, pp. 444–450.

Hiemstra, D.(1998), Multilingual domain modeling in Twenty-One: Automatic creation of a bi-directional translation lexicon from a parallel corpus, *Proceedings of the eighth CLIN meeting*, pp. 41–58.

Ide, N.(2000), Cross-lingual sense determination: Can it work?, *Computers and the Humanities, Special Issue on the Proceedings of the*

SIGLEX/SENSEVAL Workshop, A. Kilgarriff and M. Palmer, eds. **34**(1-2), 223–34.

Johansson, S.(2002), Towards a multilingual corpus for contrastive analysis and translation studies, *in* L. Borin (ed.), *Parallel Corpora, Parallel Worlds*, number 43 in *Language and Computers: Studies in Practical Linguistics*, Rodopi, Amsterdam, New York, pp. 47–59.

Resnik, P.(1999), Mining the web for bilingual text, *37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, Maryland.

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

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

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

Tjong Kim Sang, E. F.(1996), Aligning the Scania Corpus, *Technical report*, Department of Linguistics, University of Uppsala.

Tufis, D. and Barbu, A.-M.(2001), Automatic construction of translation lexicons, *Proceedings of the WSES and IEEE International Conference on Multimedia, Internet, Video Technologies*, Malta, pp. 2181–2186.

van Noord, G.(2006), TextCat, http://www.let.rug.nl/∼vannoord/TextCat/.