Cinema at the service of Natural Language Processing

Luís Carlos Cachapela Rosado
luis.c.rosado@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa

Abstract. In the Natural Language Processing area, many systems require the existence of training data in order to improve their results. Typically, increasing the training data amount in a Natural Language Processing system leads to a great improvement on system performance. But, sometimes it can be difficult to find adequate corpora for training purposes. Movie subtitles are a very important resource that is available for free and in large amounts for almost every language and can be used to obtain parallel corpora. Using different movie subtitle files for the same movie, we can extract useful information. In this work, we explore the potential of movie subtitles to produce aligned parallel corpora and extract information from them. This is done by correctly aligning the subtitle files, through the combination of state-of-the-art techniques that use not only the timing information present in subtitle files, but also the textual similarity between sentences. This represents a way of producing aligned parallel corpora that can be used in Machine Translation systems, due to the characteristics of movie dialogs. Besides the subtitle aligner, we have contributed with the creation of a subtitle dataset and reference alignments that can be used to evaluate any subtitle aligner. Using the created dataset and reference alignments, we have observed that the developed subtitle aligner successfully improves state-of-the-art results.

Keywords: Movie subtitles; Subtitle alignment; Information extraction; Building parallel corpora; Training data; Machine Translation; Reference alignments

1 Introduction

In the Natural Language Processing area, many systems require the existence of training data in order to improve their results. Movie subtitles have been used in Natural Language Processing (NLP) systems for quite a long time due to their potential to produce parallel corpora [10]. As each movie has a large number of subtitles associated and those various subtitles are usually created by different people, this is a potentially useful source of parallel corpora – different corpora that contain the same information, whether it is in a different language, whether it is written using different words. There have been authors that studied how
parallel corpora could be used to improve NLP systems [6]. This is widely used, for example, in Statistical Machine Translation (SMT) systems. In this kind of systems, typically, the more data is used to estimate the parameters of the translation model, the better the translation [6].

1.1 Objectives and contributions

The main goal of this document is the creation of parallel corpora using movie subtitles. To accomplish this, we have studied the different approaches explored by multiple authors and we have built a module capable of correctly aligning movie subtitles. By combining the existing approaches we have developed an aligner that improves state-of-the-art results, as can be seen in Section 5. Besides the creation of a movie subtitle aligner, a movie subtitle dataset has been built and reference alignments have been created for each subtitle pair. With the created reference alignments, we have made it possible to evaluate any subtitle aligner. All the developed work was made available in the Internet.

1.2 Characterization of the problem

In order to correctly use movie subtitles as a source of parallel corpora, one needs to know how subtitle files are composed. The most common subtitle format is SubRip (.srt). A SubRip subtitle is composed by several subtitle screens. For each subtitle screen, it contains a block of plain text in the following format:

1 A unique identifier
00:00:57.057 --> 00:00:58.888 Start and end timestamps
Are you watching closely? One or two lines containing the subtitle text
Empty line indicating the end of the segment

During this document we will use “subtitle screen” to refer a single subtitle dialog and “subtitle” when referring the entire subtitle file.

Despite the huge potential of movie subtitles for NLP systems, multiple problems arise when trying to extract information from them [2]. As different subtitles for the same movie are usually created by different people, it is not simple to align them due to their variations. Some of the problems faced when working with subtitles are presented below.

Lack of synchronization Not every subtitle file for the same movie has the same timestamps for each dialog, which difficults the alignment process.

Scene description insertion Some subtitles are made for hearing-impaired people, thus containing additional information (e.g. [Laughing] or [Applause]) that other subtitles do not include. Moreover, if any informative text is displayed on the screen (e.g. signs), it may or may not appear in the subtitles. For example, if a wet floor sign appears on an English movie, it is not required to include it in English subtitles, whereas it might need translation in different language subtitles.
Segmentation: Alternative subtitles for the same movie may have different sentence breaks. Also, what is said in a few words in a language might need more words in another, thus requiring the use of more subtitle screens to say the same thing.

Subtitle omission and insertion: Some subtitles do not transcribe exactly what has been said in the movie. Different subtitles usually have small differences in their texts, which makes it harder to correctly align them.

The aforementioned variations found in subtitles represent serious problems when trying to extract informations and all of them must be dealt with in order to have decent results.

Furthermore, the process of aligning subtitle files has problems that also occur when aligning regular corpora. The correspondences between segments are not always one-to-one. Different types of alignments are possible, such as one-to-many or even many-to-many, which difficulties the process.

2 Related Work

2.1 Pre-processing

To address the problems described in the previous section, it is important to pre-process the subtitle files before they are used to produce parallel corpora. In the pre-processing phase, authors normalize the subtitle files by removing noise from the subtitles [2, 9, 10]. They also ensure that every file uses the same format and encoding before proceeding. This is a small step that facilitates the alignment process.

2.2 Subtitle alignment based on similarity of segments

Lavecchia et al. [2] use dynamic programming to align movie subtitles. The authors use the F-measure between source and target segments to find the best path between segments using Dynamic Time Warping (DTW).

Tsiartas et al. [10] calculate the minimum distance between each segment and compute the path that minimizes the global distance between segments, using DTW. A dictionary is used to translate the segments in target subtitle and compare the obtained translation with the source subtitle segments.

2.3 Subtitle alignment based on timing information

Tiedemann [9] considers that approaches popular in traditional bilingual corpora alignment are not the most appropriate when aligning subtitles, because subtitles contain many insertions and deletions. The author based his alignment solution in time overlaps. The overlaps in time are a good predictor for correspondences, even with unsynchronized subtitle files.

Itamar et al. [5] used an approach that uses the duration and length of sentences, in order to find the best matches.
Other authors have also done relevant work in the corpus alignment topic [1, 3, 4, 7, 8], which was useful to better understand the alignment process. By combining the studied time and similarity based techniques we have developed an aligner that outperforms state-of-the-art aligners, as can be seen in Section 5.

3 Alignerzilla

In order to achieve the aforementioned goals, Alignerzilla was created. The developed solution has been built having in mind the presented problems and taking into account the research done by other authors. The developed aligner aligns subtitles at the sentence level and considers not only one-to-one alignments, but also one-to-many.

Our work has been divided in two major phases: the pre-processing and the subtitle alignment phase. We have included a configuration file, where the user can specify some options before running the aligner. The following subsections describe what was done.

3.1 Architecture overview

As we want to create parallel corpora using movie subtitles, we can see the solution as a production line where the starting product (input) are movie subtitles in their raw state and the final product are the aligned subtitles (output). Figure 1 represents the different steps of our solution. The following subsections explain in detail how each of the components were developed.

3.2 Pre-processing

3.2.1 Encoding The first step in the pre-processing phase is the conversion of the character encodings. Following an approach similar to Tiedemman [9], a script was written to ensure the subtitle files are using UTF-8 encoding. The developed aligner expects all the files to use this encoding.

3.2.2 Normalization This is the core of the pre-processing phase. A script has been developed in order to remove noise from the subtitles and format the files in a way that makes it simpler to find alignments.

In this phase, we remove tags that appear in some subtitle files (e.g. <font color="#FFFF00">) and also all the scene descriptions found. We also mark with a distinctive symbol (in this case >>) the sentences that appear to be unfinished, so that this information can be used to complete sentences as explained in 3.2.3.
3.2.3 Sentence Concatenation If two consecutive subtitle screens complete each other, they are concatenated so that no sentences are left unfinished. When text from different subtitle screens is concatenated, the timestamps of the resulting sentence are recalculated. A concatenation example can be seen in Table 1.

<table>
<thead>
<tr>
<th>Before concatenation</th>
<th>After concatenation</th>
</tr>
</thead>
<tbody>
<tr>
<td>193 00:13:38,340 --&gt; 00:13:41,930 Gentlemen, might I remind you that my odds of success...</td>
<td>193 00:13:38,340 --&gt; 00:13:44,220 Gentlemen, might I remind you that my odds of success... dramatically improve with each attempt?</td>
</tr>
</tbody>
</table>

**Table 1. Sentence concatenation example taken from A Beautiful Mind movie.**

3.2.4 Sentence Splitting In order to align subtitles at the sentence level, not only do we need to concatenate sentences that appear in different subtitle screens, but also to split sentences that appear together in the same subtitle screen. In Table 2 we can see an example.

<table>
<thead>
<tr>
<th>Before splitting</th>
<th>After splitting</th>
</tr>
</thead>
</table>

**Table 2. Sentence splitting example taken from A Beautiful Mind movie.**

The sentence splitter looks for final punctuation but takes into account special cases that may appear, such as “Mr.,” “Dr.”, or “St.”. As this special cases are language dependent, new ones can be added using the configuration file. The start and end timestamps of split sentences are recalculated using Equation 1 [9].

\[
t_{new} = t_{before} + c_{before} \cdot \frac{t_{after} - t_{before}}{c_{before} + c_{after}}
\]  

In Equation 1, \( t_{before} \) and \( t_{after} \) are the nearest time events before and after the current position, respectively. Likewise, \( c_{before} \) and \( c_{after} \) are the lengths of the strings before and after the current position up to the nearest time events.
3.3 Aligner

As we have seen in the research done by other authors, alignment techniques based on the timing information available on the subtitle files represent a powerful and reliable way of aligning subtitles [5, 9]. This type of approach allows to have well aligned subtitles, but depends on the subtitle synchronization. On the contrary, subtitle alignment approaches based on the similarity of segments are more flexible regarding time, but are more susceptible to produce erroneous results [2, 10].

In order to obtain a module that aligns subtitles with a good performance and is flexible enough to align subtitle pairs that are not synchronized, we have combined the two techniques. The time based approach is the main provider of trustworthy alignments, while the similarity based technique is only used when the first cannot find alignments with a good confidence level. Figure 2 represents how we have combined time and similarity techniques. If neither method is able to obtain a valid alignment, Alignerzilla assumes there is no possible alignment for the given sentence.

![Fig. 2. Overview of the alignment process for each iteration.](image)

3.3.1 Time based alignment Alignerzilla starts by iterating over source subtitle looking for matches in target subtitle. Using the starting time of a source sentence, the aligner looks for the sentence in target that starts closer to that time. If those sentences have similar starting times and similar duration they are considered a match. However, if starting times match but the durations do not, it means one of the sentences is shorter than the other. In those cases we consider the possibility of a 1:2 or 2:1 alignment.

After using the time based approach, either the match is considered valid or the match is not valid – the sentences are too different duration-wise – and the textual similarity must be used in order to find a trustworthy alignment.

3.3.2 Similarity based alignment To align sentences that could not be aligned using timing information, a solution similar to the work done by Tsiarttas et al. [10] is used. We use a Distance Metric that we try to minimize. In
other words, we have defined the differences between two sentences as their distance. Alignerzilla chooses as an alignment the sentence pair that has the smaller distance. However, instead of using dynamic programming to do so (like Tsiartas et al. [10]), the aligner uses a sliding window approach, since we only want to use this technique on sentences that could not be aligned using the timing information. Thus, in order to find an alignment for a source sentence, the aligner chooses a target sentence based on the last alignments information and it looks to a window before and after that sentence, comparing the distances between sentences.

To calculate the distance between two sentences Alignerzilla uses two metrics: word and length similarity. The Distance ($D$) between two sentences ($S_i, T_j$) is calculated as Equation 2 and varies between 0 and 1. This equation can be seen as a variation of Itamar [5] cost function.

\[
D(S_i, T_i) = \lambda \left(1 - \frac{M_w}{N_w}\right)^2 + (1 - \lambda) \left(\frac{\min(L(S_i), L(T_j))}{\max(L(T_i), L(T_j))}\right)
\]  

(2)

Where $M_w$ is the number of words that match between the two sentences, $N_w$ the number of words in the biggest sentence and $L(s)$ the length of sentence $s$. The $\lambda$ parameter should vary between 0 and 1 and represents the relative importance of the word similarity. To decide if two words match, we chose not to use a direct comparison, because that would reject words with irrelevant differences (e.g. color vs colour). Hence, the Levenshtein distance – the difference between two character sequences – is calculated and used to decide if the words match. Obviously, if the two subtitles used are in different languages, we need to translate them before any comparison is made, as explained in subsection 3.3.3.

3.3.3 Translator To use the sentence similarity approach with subtitles of different languages, one of the sentences needs to be translated so that they can be compared.

In order to translate sentences, a bilingual dictionary has been built. A list of the top 5000 English words has been downloaded from the Word Frequency website\(^1\). Then, using an API for Google Translate\(^2\) all the words have been translated and stored as a bilingual lexicon – list of words that correspond. This process could be used to obtain dictionaries for other languages.

The developed pre-processing scripts and Alignerzilla are available at the repository: https://github.com/b2rosado/subtitle-alignment. Instructions on how to use the pre-processing scripts and the subtitle aligner are included in the repository read me file.

---

\(^1\) http://www.wordfrequency.info/

\(^2\) https://code.google.com/archive/p/java-google-translate-text-to-speech/
4 Building reference alignments

4.1 Dataset

In order to evaluate the developed solution, OpenSubtitles\(^3\) administrators provided us a great amount of SubRip (.srt) subtitles in English and Portuguese languages.

Inspired by the evaluation methodologies used by other authors \([2, 9, 10]\), we have selected 40 movies to evaluate our solution. The movie selection was manually done and we tried to pick different types of movies.

4.2 Reference creation

For the 40 subtitle pairs selected, reference alignments have been created between English and Portuguese languages, with the help of several annotators.

A web application was developed where the annotators would help us create reference alignments. Each annotator was given a different subtitle pair to align using the developed web application. In the end of the process, each annotator had manually selected 45 alignments. Afterwards, different annotators checked the alignments correctness. By using this process, we have obtained 1789 impartially chosen alignments that can pose as reference for any subtitle aligner.

The created subtitle dataset and reference alignments can be found at the aforementioned repository.

5 Evaluation

Using the dataset described in Section 4, we have obtained the alignments for those subtitles using the developed aligner. Afterwards, those alignments were manually analyzed and compared with the reference alignments.

To be able to properly evaluate the aligner performance, we have calculated the following values:

**True Positive** (TP) finds an alignment that matches the reference;  
**False Positive** (FP) finds an alignment but does not match the reference;  
**True Negative** (TN) does not find an alignment and neither does the reference;  
**False Negative** (FN) does not find an alignment but the reference does.

After defining those variables, we have calculated the Precision and Recall of each aligned subtitle pair with the formulas in 3.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN} \tag{3}
\]

In order to compare the results more easily, we have also calculated the traditional F-measure for each aligned subtitle pair, using the formula in Equation 4.

\[
F_M = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}
\]

\(^3\) [http://www.opensubtitles.org](http://www.opensubtitles.org)
5.1 Results

Following the described methodology, the average Precision, Recall and F-measure obtained for Alignerzilla alignments was 82.7%, 89.8% and 85.8%, respectively.

In order to see how the Alignerzilla performed relatively to a state-of-the-art subtitle aligner, we have decided to compare it with Tiedemann aligner [9]. The goal of this comparison was to see if the developed aligner outperformed a state-of-the-art aligner and if so, to quantify the improvements.

We have aligned the dataset subtitles with subalign 4 and the average Precision, Recall and F-measure obtained was 70.1%, 85.3% and 76.1%, respectively.

By comparing the results we can see that, in average, the developed aligner performs slightly better then subalign in every metric used. The obtained values are a good indicator that make us confident in the performance of Alignerzilla. While analyzing the results for each movie, we can see that Tiedemann solution still has better results in some of the subtitle pairs, as can be seen in Figure 3. This means that our aligner still has room for improvements that should be looked into in future work.

![Graphical representation of the F-measure calculated for each subtitle pair aligned using Tiedemann [9] and comparison with Alignerzilla.](https://bitbucket.org/tiedemann/subalign)

**Fig. 3.** Graphical representation of the F-measure calculated for each subtitle pair aligned using Tiedemann [9] and comparison with Alignerzilla.

However, these results are relative to our dataset, that only comprises alignments between English and Portuguese subtitles. We cannot predict how the results would be if a different dataset was used.

---

4 [https://bitbucket.org/tiedemann/subalign](https://bitbucket.org/tiedemann/subalign)
6 Conclusions

We believe the developed aligner achieves the proposed objectives. Alignerzilla is capable of creating parallel corpora that can be used in NLP systems as training data. Furthermore, the creation of the reference alignments was a useful contribution to this topic because it makes possible for everyone to test a subtitle aligner.

Nevertheless, the aligner still has some limitations. For instance, it only comprises 1:0, 1:1, 1:2 and 2:1 alignments. Also, the textual similarity alignment technique implemented uses a dictionary that contains only one translation per word, which is not reliable in every case because some words may have different meanings. No time synchronization is done. The aligner could be enhanced by fixing the current limitations.

References