OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles

Pierre Lison*, **Jörg Tiedemann**†

* Department of Informatics
University of Oslo
plison@ifi.uio.no

† Department of Modern Languages
University of Helsinki
jorg.tiedemann@helsinki.fi

Abstract

We present a new major release of the *OpenSubtitles* collection of parallel corpora. The release is compiled from a large database of movie and TV subtitles and includes a total of 1689 bitexts spanning 2.6 billion sentences across 60 languages. The release also incorporates a number of enhancements in the preprocessing and alignment of the subtitles, such as the automatic correction of OCR errors and the use of meta-data to estimate the quality of each subtitle and score subtitle pairs.

Keywords: Parallel corpora, Bitext alignment, Statistical Machine Translation

1 Introduction

Movie and TV subtitles constitute a prime resource for the compilation of parallel corpora. From a linguistic perspective, subtitles cover a wide and interesting breadth of genres, from colloquial language or slang to narrative and expository discourse (as in e.g. documentaries). Large databases of subtitles are also available and continue to grow rapidly – for instance, the *OpenSubtitles*[1] database contains more than 3 million subtitles in over 60 languages. Finally, and perhaps most importantly, the tight connection between subtitles and their corresponding source material – usually a movie or TV episode – makes it possible to efficiently align subtitles across languages based on time overlaps (Tiedemann, 2007).

This paper presents a new release of the *OpenSubtitles* collection of parallel corpora. The new release includes a total of 1689 bitexts extracted from a collection of subtitles of 2.6 billion sentences (17.2 billion tokens) distributed over 60 languages. In addition to increasing the global volume of the dataset by approximately 39 % compared to the previous version (Tiedemann, 2012), the release also includes several important improvements regarding how the subtitles are preprocessed and aligned with one another. Figure 1 illustrates the processing workflow, starting with the raw subtitle files and ending with the resulting bitexts.

Sections 2 and 3 present the dataset and the preprocessing techniques employed to convert the subtitle files into sentences and tokens encoded in XML. Section 4 describes the alignment process for cross-lingual alignments, and Section 5 extends it to alignments of alternative subtitles within the same language. Section 6 concludes this paper.

2 Source Data

The dataset consists of a database dump of the *OpenSubtitles.org* repository of subtitles, comprising a total of 3.36 million subtitle files covering more than 60 languages. We filtered out the languages associated with less than 10 subtitles, as they are typically the result of human classification errors. Most files are encoded in the `.srt` format. Subtitles that are encoded in an alternative format are first converted to `.srt` before processing. The following information is provided for each subtitle:

1. A unique identifier,
2. A list of files (there may be more than one file in the case of movies with multiple CDs),
3. A language code and subtitle format,
4. Generic information on the source material such as its title, release year, and IMDb[2] identifier,
5. Miscellaneous attributes such as the file's upload date, number of downloads, and user ratings.

Table 1 provides detailed statistics on the dataset, in terms of number of subtitles, number of sentences and number of tokens per language. The initial number of subtitles does not correspond one-to-one to the number of converted subtitles, as some files are discarded from the conversion due to e.g. unsupported subtitle formats, misclassified files, corrupt encodings or other conversion errors. Furthermore, the administrators of *OpenSubtitles* have introduced over the last years various mechanisms to sanitise their database and remove duplicate, spurious or misclassified subtitles. Compared to the previous release, the numbers of covered movies and TV episodes have therefore increased more rapidly than the raw number of subtitles. For instance, the number of English subtitles has only risen from 310K to 322K subtitles between 2013 and 2016, while the number of IMDbs went from 65K to 106K. Nevertheless, our collection increases in terms of numbers of subtitles for the vast majority of languages, and four new languages are introduced: Breton, Esperanto, Georgian and Tagalog.

The dataset covers a total of 152 939 movies or TV episodes (as determined by their IMDb identifier). 70% of the IMDb

[1] http://www.opensubtitles.org.

[2] Internet Movie Database, http://www.imdb.com.

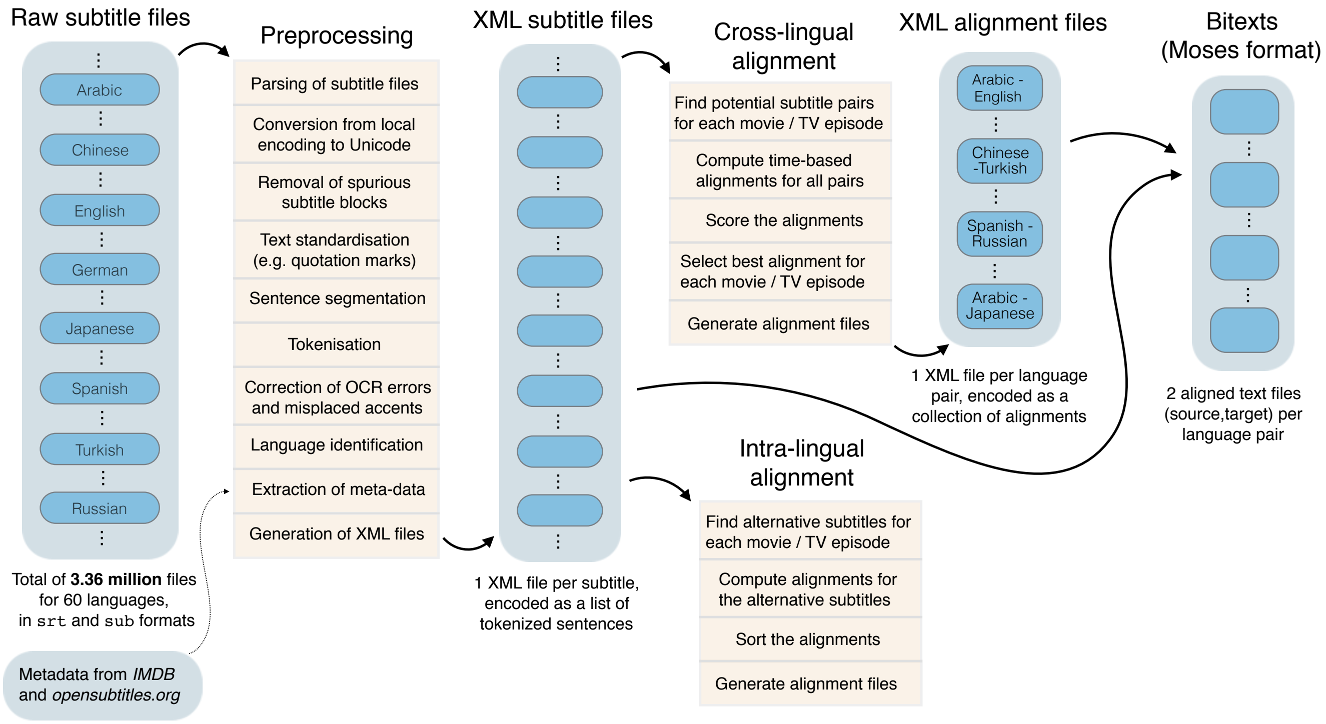
Raw subtitle files
Arabic
Chinese
English
German
Japanese
Spanish
Turkish
Russian
Total of 3.36 million files for 60 languages, in srt and sub formats
Metadata from IMDB and opensubtitles.org
Preprocessing
Parsing of subtitle files
Conversion from local encoding to Unicode
Removal of spurious subtitle blocks
Text standardisation (e.g. quotation marks)
Sentence segmentation
Tokenisation
Correction of OCR errors and misplaced accents
Language identification
Extraction of meta-data
Generation of XML files
XML subtitle files
1 XML file per subtitle, encoded as a list of tokenized sentences
Cross-lingual alignment
Find potential subtitle pairs for each movie / TV episode
Compute time-based alignments for all pairs
Score the alignments
Select best alignment for each movie / TV episode
Generate alignment files
Intra-lingual alignment
Find alternative subtitles for each movie / TV episode
Compute alignments for the alternative subtitles
Sort the alignments
Generate alignment files
XML alignment files
Arabic - English
Chinese -Turkish
Spanish - Russian
Arabic - Japanese
1 XML file per language pair, encoded as a collection of alignments
Bitexts (Moses format)
2 aligned text files (source,target) per language pair

Figure 1: Processing workflow for the creation of parallel corpora from the raw subtitle files.

identifiers are associated with subtitles in at least two languages, 44% with at least 5 languages, 28% with at least 10 languages, and 8% with at least 20 languages. The dataset also includes bilingual Chinese-English subtitles, which are subtitles displaying two languages at once, one per line (Zhang et al., 2014). These bilingual subtitles are split in their two constituent languages during the conversion.

3 Preprocessing

The subtitle files must undergo a number of preprocessing steps before they can be aligned with another. The output is a collection of XML files (one per subtitle), where each file is structured as a list of tokenised sentences.

Subtitle conversion

OpenSubtitles does not enforce any particular encoding format on the subtitles uploaded by its users. The most likely encoding for the file must therefore be determined based on various heuristics. This is a difficult and error-prone process, especially for older files which are more likely to rely on language-specific encodings (such as Windows code pages) instead of Unicode.
We addressed this problem by specifying a list of possible character encodings for each language in the dataset (for instance, common encodings for English subtitles are UTF-8, windows-1252 and ISO-8859-1). When several alternative encodings are admissible, the `chardet` library is applied to determine the most likely encoding given the file content (Li and Momoi, 2001).

Sentence segmentation and tokenisation

The raw subtitle files are structured in blocks, which are short text segments associated with a start and end time. These blocks are expected to obey specific time and space constraints: at most 40-50 characters per line, a maximum of two lines and an on-screen display between 1 and 6 seconds (Aziz et al., 2012). There is no direct, one-to-one correspondence between subtitle blocks and sentences, as illustrated in this sequence of subtitle blocks:

```
5
00:01:15,200 --> 00:01:20,764
Nehmt die Halme, schlagt sie oben ab,
entfernt die Blätter

6
00:01:21,120 --> 00:01:24,090
und werft alles auf einen Haufen
für den Pflanztrupp.

7
00:01:24,880 --> 00:01:30,489
Das Zuckerrohr beißt euch nicht.
Nicht so zaghaft! Na los, Burschen, los!
```

In the example above, the subtitle block 6 is used as a continuation of the sentence started in the block 5, while the last block contains 3 sentences, thereby summing up to 4 sentences spanning the 3 blocks.
Algorithm 1 illustrates the segmentation process, which is an adaptation of the approach used in our earlier work (Tiedemann, 2007). Each text line of the subtitle is first tokenised (line 10). When a sentence-ending marker is encountered, such as a period followed by an uppercase letter, the current tokens are recorded and a new sentence is started (lines 12-14). The detection of sentence endings obeys language-specific rules, as sentence-ending markers

	2012+13 release	**2016 release**			
Language	*Subtitle files*	**Subtitle files**	**Covered IMDbs**	**Sentences**	**Tokens**
Afrikaans	*2*	32	26	27.4K	205K
Albanian	*3.4K*	3.0K	1.9K	3.4M	23.5M
Arabic	*42.4K*	67.3K	34.1K	60.0M	327M
Armenian	*1*	1	1	1.1K	8.1K
Basque	*216*	188	167	230K	1.4M
Bengali	*8*	76	71	115K	0.6M
Bosnian	*7.1K*	30.5K	17.5K	28.4M	180M
Breton		32	28	23.1K	165K
Bulgarian	*73.0K*	90.4K	49.3K	80.2M	0.5G
Catalan	*305*	0.7K	0.7K	0.5M	4.0M
Chinese (simplified)	*3.1K*	22.4K	12.0K	24.8M	158M
Chinese (traditional)	*1.5K*	6.7K	4.5K	7.4M	51.4M
Bilingual Chinese-English		4.5K	3.0K	5.0M / 5.2M	34.1M / 35.0M
Croatian	*63.9K*	96.8K	41.3K	88.6M	0.6G
Czech	*105K*	125K	51.3K	113M	0.7G
Danish	*20.1K*	24.1K	14.6K	23.6M	164M
Dutch	*87.4K*	98.2K	46.6K	84.7M	0.6G
English	*310K*	322K	106K	337M	2.5G
Esperanto		89	81	79.3K	0.5M
Estonian	*18.3K*	23.5K	13.2K	22.9M	141M
Finnish	*51.2K*	44.6K	31.8K	38.7M	209M
French	*99.4K*	105K	56.4K	90.6M	0.7G
Galician	*3*	370	345	245K	1.9M
Georgian		271	245	262K	1.6M
German	*17.2K*	27.7K	20.1K	26.9M	187M
Greek	*85.4K*	114K	49.9K	102M	0.7G
Hebrew	*64.8K*	79.7K	35.6K	55.0M	406M
Hindi	*45*	57	51	81.6K	0.6M
Hungarian	*85.8K*	99.3K	52.7K	80.7M	490M
Icelandic	*1.3K*	1.3K	1.2K	1.7M	11.0M
Indonesian	*194*	11.0K	6.1K	12.4M	75.6M
Italian	*57.0K*	96.5K	41.9K	77.1M	0.6G
Japanese	*1.2K*	2.6K	2.2K	2.3M	17.3M
Korean	*49*	0.7K	0.5K	0.7M	3.3M
Latvian	*350*	392	369	499K	2.9M
Lithuanian	*1.0K*	1.5K	1.4K	1.8M	9.7M
Macedonian	*2.1K*	5.6K	3.6K	5.8M	37.2M
Malay	*6.7K*	1.0K	0.8K	1.3M	7.8M
Malayalam	*34*	251	225	308K	1.7M
Norwegian	*7.0K*	8.9K	7.2K	8.8M	58.9M
Persian	*4.7K*	6.5K	4.4K	7.4M	44.3M
Polish	*151K*	161K	44.0K	143M	0.9G
Portuguese	*94.2K*	96.3K	36.2K	91.8M	0.6G
Portuguese (BR)	*163K*	220K	77.0K	200M	1.3G
Romanian	*129K*	162K	58.1K	154M	1.0G
Russian	*23.7K*	38.7K	28.8K	32.7M	215M
Serbian	*53.4K*	148K	56.3K	140M	0.9G
Sinhalese	*166*	0.5K	476	0.6M	3.5M
Slovak	*11.6K*	14.7K	10.1K	13.3M	85.8M
Slovenian	*44.1K*	52.6K	22.8K	53.4M	322M
Spanish	*160K*	192K	76.1K	179M	1.3G
Swedish	*25.2K*	27.3K	16.9K	25.6M	173M
Tagalog		52	51	9.2K	67.4K
Tamil	*15*	17	17	24.4K	126K
Telugu	*17*	20	20	29.9K	157K
Thai	*9.2K*	10.2K	5.0K	8.3M	17.3M
Turkish	*85.0K*	159K	55.0K	149M	0.8G
Ukrainian	*372*	1.0K	0.9K	0.8M	5.4M
Urdu	*7*	14	14	17.7K	133K
Vietnamese	*1.3K*	3.1K	2.5K	3.3M	26.9M
Total	*2.2M*	2.8M		2.6G	17.2G

Table 1: Statistics for the 60 languages in the extracted corpus. The *subtitles files* corresponds to the number of converted subtitles (which may be lower than the number of raw subtitles in the database due to discarded files). The *covered IMDbs* represent the number of distinct movies or TV episodes (denoted by their IMDb identifier) covered by the subtitles. The first column stands for the total number of files in the 2012+2013 release.

Algorithm 1 : Sentence Segmentation

```
 1: processed_sentences ← {}
 2: tokens_stack ← {}
 3: for all block ∈ subtitle do
 4:   score = continuation_score(tokens_stack,block)
 5:   if score <threshold then
 6:     Add tokens_stack to processed_sentences
 7:     tokens_stack ← {}
 8:   end if
 9:   for all line ∈ block do
10:     for all token ∈ tokenise(line) do
11:       Add token to tokens_stack
12:       if token is sentence-ending marker then
13:         Add tokens_stack to processed_sentences
14:         tokens_stack ← {}
15:       end if
16:     end for
17:   end for
18: end for
19: return processed_sentences
```

vary from language to language, due for instance to distinct punctuation marks or unicameral vs. bicameral alphabets[3]. Upon processing a new subtitle block, the algorithm first determines the likelihood of the new block being a continuation of the previous sentence (line 4). This likelihood is determined from various heuristics such as the time gap between the two subtitles and the presence of punctuation markers between the two – for instance, three dots at the end of the previous subtitle is sometimes used as marker for an unfinished sentence. The process is repeated for each block in the subtitles, resulting in a sequence of tokenised sentences coupled with timing information.

For Japanese and Chinese, the KyTea[4] word segmentation library is used for the tokenisation (Neubig et al., 2011) along with pre-trained models. For other languages, the default tokeniser script from Moses is employed, along with language-specific non-breaking prefixes.

Correction of OCR and spelling errors

Many subtitles in our dataset are automatically extracted via Optical Character Recognition (OCR) from video streams, leading to a number of OCR errors. A particularly common error arise from misrecognising the characters `i`, `I` and `l`. We relied on a simple noisy-channel approach to automatically detect and correct such errors. The approach integrates a handcrafted error model together with a statistical language model compiled from the Google's Web 1T N-grams (Brants and Franz, 2006) for 11 European languages (English, Czech, Dutch, French, German, Italian, Polish, Portuguese, Romanian, Spanish and Swedish). Let w_t^o denote a (possibly erroneous) token observed at position t, w_t its actual (error-free) token, and w_{t-1} its preceding token. We can apply Bayes rule to calculate the probability of w_t given the error model $P(w_t^o|w_t)$, the bigram model $P(w_t|w_{t-1})$ and a normalisation factor α:

$$P(w_t|w_t^o, w_{t-1}) = \alpha P(w_t^o|w_t) P(w_t|w_{t-1}) \qquad (1)$$

The possible w_t tokens to consider for each observed w_t^o token are determined by enumerating the possible character confusions and recording all possible replacements that result in actual tokens in the language model. In addition to OCR errors, the method was also applied to correct misplaced accent marks (for instance “étè” instead of “été” in French) which are quite commonplace in subtitles.

A total of 9.04 million words were corrected with this correction technique, with an average of 3.2 corrections per subtitle for the 11 languages. In order to evaluate the correction performance, we extracted for each language a collection of 100 subtitles satisfying two criteria:

1. They were likely to contain OCR errors (based on the observed proportion of out-of-vocabulary tokens),

2. Their corresponding source material had at least another subtitle seemingly without OCR errors.

We used the OCR-free subtitles as gold standard for the evaluation. Based on the intra-lingual alignments described in Section 5, we examined all sentence pairs that were either identical or only differed in a few characters likely to be misspellings, and then calculated the precision and recall on their tokens. The results in Table 2 are provided for two correction methods: a “conservative” corrector which only corrects tokens that are highly likely to be mispellings, and an “aggressive” corrector that corrects all tokens that are more likely to be mispellings than not.

Language	Method	Precision	Recall	F_1 score
Czech	C	0.994	0.467	0.635
	A	0.992	0.503	0.668
Dutch	C	0.994	0.801	0.887
	A	0.993	0.830	0.904
English	C	0.992	0.783	0.875
	A	0.991	0.835	0.907
French	C	0.993	0.574	0.727
	A	0.988	0.592	0.740
German	C	0.995	0.656	0.791
	A	0.993	0.797	0.884
Italian	C	0.971	0.481	0.643
	A	0.964	0.509	0.667
Polish	C	0.991	0.415	0.585
	A	0.988	0.496	0.661
Portuguese	C	0.992	0.679	0.806
	A	0.991	0.708	0.826
Romanian	C	0.950	0.262	0.411
	A	0.948	0.294	0.448
Spanish	C	0.990	0.508	0.672
	A	0.989	0.528	0.689
Swedish	C	0.983	0.577	0.727
	A	0.982	0.621	0.761

Table 2: Evaluation results by language for the “conservative” (C) and “aggressive” (A) correction methods.

[3] Bicameral alphabets such as Latin or Cyrillic have two versions of each letter: one lowercase and one uppercase. Other scripts, such as Arabic, do not make such distinctions.

[4] http://www.phontron.com/kytea/

As evidenced by Table 2, the OCR correction has very high precision but a weaker recall. One explanation is that, for tractability reasons, the current method is currently limited to a maximum of one character confusion per token. This means that tokens including more than one misrecognised character cannot be corrected. Furthermore, there is a weak correlation (Pearson coefficient $R = 0.498$) between the size of the language model and the recall for that language. This can partly explain why Romanian (the language with the lowest number of bigrams) has a much weaker recall than e.g. English. Many proper nouns (persons, places, brands) were also ignored from the correction process, as many of these nouns are relatively rare and therefore not well captured by statistical language models.

Inclusion of meta-data

The last preprocessing step is to generate the meta-data associated with each subtitle. This meta-data includes the following information:

- Generic attributes of the source material, such as the release year, original language, duration, and genre of the movie or TV episode. These attributes are extracted from the IMDb database.
- Attributes of the subtitle, such as the subtitle language, upload date, subtitle rating on *OpenSubtitles* (online user votes), and subtitle duration.
- Probability that the specified language of the subtitle matches the actual language used in the subtitle text, based on the output of the `langid` language identification tool (Lui and Baldwin, 2012).
- Features of the conversion process, such as the number of extracted sentences, total number of tokens, number of detected OCR errors and file encoding.

4 Cross-lingual alignments

Once the subtitles files are processed, they can be aligned with one another to form a parallel corpus. To align subtitles across distinct languages, we first need to determine which subtitles to align, as many alternative subtitles may exist for a given movie / TV episode. Once the subtitle pairs are selected, the sentences are aligned one by one using a timing-based approach (Tiedemann, 2008).

Document alignment

Let A and B be two arbitrary languages and I be the IMDb identifier for a given source material (movie or TV episode). We define $S_{I,A}$ and $S_{I,B}$ to respectively represent the two sets of subtitles for I in the languages A and B. The first step in the alignment process is to score each pair of subtitles $(s_1, s_2) \in S_{I,A} \times S_{I,B}$ according to a handcrafted scoring function on the following features:

- Upload date of the subtitle (since more recent subtitles are often corrections of previous ones); we compute a *recency* feature based on the date relative to the first and the latest upload.
- Confidence score of the language identification tool.
- User rating of the subtitle, if they exist.
- File encoding (UTF-8 encodings being less prone to conversion errors than language-specific encodings); we use a binary feature to indicate whether the data was provided as UTF-8 or not.
- Number of corrections and unknown words detected during the file conversion.
- Distance between the duration of the source material and the duration of the subtitle.
- Relative time overlap of subtitle frames between source and target language subtitles.

The subtitle pairs are then ranked on the basis of this scoring function and the top 10 pairs are aligned.

Sentence alignment

A time-overlap algorithm is employed for the alignment (Tiedemann, 2007). The key idea behind this strategy is to exploit the rich timing information encoded in the subtitles to determine the most likely alignments. The alignment algorithm performs a single run through the subtitle pair by moving a sliding window through the subtitle pair and determines at each step the sentence alignment with the highest time overlap. Missing time stamps for the start or end of sentences[5] are interpolated based on the surrounding time stamps.

To account for small timing variations due to differences in frame rate and starting time across subtitles, the speed ratio and starting time are adjusted using anchor points, which can be extracted either from cognates or bilingual dictionaries (see Tiedemann (2008) for details). The original algorithm relied on the ratio between non-empty alignments and empty alignments to determine a good synchronisation between the movies. We replace this with the proportion of non-empty alignments relative to the overall number of alignment units in the subtitle pair, which is easier to combine with the scoring function described above.

Finally, we select the subtitle pair that maximises the overall score, defined as a weighted sum (with handcrafted weights) of the individual factors – except the time overlap, which can be misleading due to the synchronisation resulting from the alignment – and the relative proportion of non-empty alignments. We also use another threshold on the relative time overlap after synchronisation to improve the selection even more.

Resulting bitexts

The alignment process described above resulted in a total of 1689 bitexts, the largest bitext being for English-Spanish, with a total of about 50 million aligned sentences. Detailed statistics for the 20 largest bitexts is provided in Table 3.

[5] This may happen when a sentence finishes in the middle of a subtitle block, such as in the example "`Das Zuckerrohr beißt euch nicht.`" in the excerpt from Section 3.

Language pair	Aligned docs	Sentence pairs	Tokens
English-Spanish	62.2K	49.2M	760M
English-Portuguese	61.1K	46.6M	709M
Spanish-Portuguese	56.3K	42.3M	627M
English-Romanian	48.8K	38.8M	584M
English-Turkish	47.4K	36.6M	502M
Spanish-Romanian	45.5K	34.5M	514M
Portuguese-Romanian	45.5K	34.0M	496M
English-Serbian	44.1K	33.8M	499M
Czech-English	44.3K	33.2M	488M
English-Hungarian	44.7K	33.0M	473M
English-French	43.9K	32.6M	521M
Spanish-Turkish	44.1K	32.5M	439M
Portuguese-Turkish	43.9K	32.1M	421M
Bulgarian-English	41.5K	30.6M	465M
Czech-Spanish	41.7K	30.1M	438M
Czech-Portuguese	41.9K	30.0M	423M
Spanish-Serbian	40.6K	29.9M	436M
Romanian-Turkish	39.9K	29.8M	393M
Greek-English	39.6K	29.6M	452M
Portuguese-Serbian	40.6K	29.6M	420M

Table 3: Statistics for the 20 largest bitexts. Portuguese refers to Brazilian Portuguese in this table.

BLEU scores

In order to empirically evaluate the quality of the alignments, we performed an extrinsic evaluation using the bitexts as training material for a statistical machine translation system based on Moses (Koehn et al., 2007). For each language pair, we compiled a language model based on the monolingual data for the target language, extracted a phrase table and a lexicalised reordering table from the bitext, and tuned the model weights using MERT (Och, 2003) based on a small tuning set extracted from the same bitext.

The test sets used in the evaluation comprised 10 subtitles for each language pair, and relied on the intra-lingual alignments (described in the next section) to provide alternative reference translations. Table 4 details the BLEU scores for 18 language pairs. As we can observe from the table, the 2016 release is able to yield substantial improvements in terms of BLEU scores compared to the previous release. The BLEU scores remain nevertheless quite low for some language pairs. This is in no small part due to the fact that subtitle translations are often less literal than translations in other domains, and must also obey the time and space constraints specific to this media.

5 Intra-lingual alignments

Another novel extension of the *OpenSubtitles* collection of parallel corpora is the inclusion of intra-lingual alignments. Aligning alternative subtitles within each language is indeed interesting for several reasons. First of all, these alignments can be used to create a fully connected multilingual corpus across many languages. With the cross-lingual alignment strategy outlined in the previous section, it is not always possible to find links across more than two languages because different subtitle alternatives may be chosen for different language pairs.

Language pair	2012+2013 release	2016 release
Spanish-English	35.49	39.15
English-Spanish	31.22	34.05
Turkish-English	23.09	24.78
English-Turkish	14.70	16.14
English-French	21.13	21.26
French-English	23.69	24.01
Polish-English	25.25	26.62
English-Polish	18.90	21.21
Russian-English	22.47	25.49
English-Russian	15.75	17.05
Arabic-English	24.37	25.34
English-Arabic	9.37	9.28
Portuguese-English	33.10	33.34
English-Portuguese	27.37	27.49
Chinese-English	15.99	18.20
English-Chinese	11.60	11.85
Czech-English	28.65	29.67
English-Czech	20.51	22.12

Table 4: BLEU scores for the SMT systems based on the bitexts of the 2013 and 2016 release of OpenSubtitles.

Alternative subtitles can also be fruitfully exploited to:

- Detect errors (spelling mistakes, erroneous encodings, etc.) in the corpus;
- Discover insertions and deletions that may, among others, refer to extra-linguistic information;
- Extract paraphrases and translation alternatives.

The current procedure is based on the same time-based algorithm as for inter-lingual alignment, but includes a BLEU-filter and search heuristics over neighbouring links to improve the alignment quality. Additionally, we use string similarity metrics based on edit distance to distinguish between different alignment categories that refer to possible spelling errors, insertions or paraphrases. Details of the approach are presented in Tiedemann (2016) and will be omitted here.

6 Conclusion

This paper described the release of an extended and improved version of the *OpenSubtitles* collection of parallel corpora. The subtitles included in this release are first preprocessed to convert the subtitle blocks into tokenised sentences. The converted subtitles are then aligned with one another via a time-based approach. This alignment is performed both across languages, but also within alternative subtitles for the same language.

The corpora is made freely available to the research community on the OPUS website:

http://opus.lingfil.uu.se/OpenSubtitles2016.php

As future work, we wish to improve the OCR-correction method with a data-driven error-model and extend it beyond the current 11 languages. We would also like to investigate the integration of contextual information provided in movie transcripts into the generated XML files.

Acknowledgements

The authors would like to thank the administrators of www.opensubtitles.org for giving us access to their database of subtitles and helping us prepare this release.

7 References

Aziz, W., de Sousa, S. C. M., and Specia, L. (2012). Cross-lingual sentence compression for subtitles. In *16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, pages 103–110, Trento, Italy.

Brants, T. and Franz, A. (2006). Web 1T 5-gram corpus version 1. Technical report, Google Research.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 177–180, Prague, Czech Republic.

Li, S. and Momoi, K. (2001). A composite approach to language/encoding detection. In *19th International Unicode Conference (IUC 2001)*, San Jose, California.

Lui, M. and Baldwin, T. (2012). Langid.py: An off-the-shelf language identification tool. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012 System Demonstrations)*, pages 25–30, Jeju Island, Korea.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 529–533, Portland, Oregon, USA.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL 2003)*, pages 160–167, Sapporo, Japan.

Tiedemann, J. (2007). Improved sentence alignment for movie subtitles. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria.

Tiedemann, J. (2008). Synchronizing translated movie subtitles. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 1902–1906, Marrakesh, Marocco.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2214–2218, Istanbul, Turkey.

Tiedemann, J. (2016). Finding alternative translations in a large corpus of movie subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia.

Zhang, S., Ling, W., and Dyer, C. (2014). Dual subtitles as parallel corpora. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1869–1874, Reykjavik, Iceland.