OpenSubtitles 2016: 
Extracting Large Parallel Corpora from Movie/TV Subtitles

Pierre Lison  
University of Oslo

Jörg Tiedemann  
University of Helsinki

LTG Seminar, February 2016
Outline of the talk

- Introduction
- Source Data
- Preprocessing
- Alignment
- Conclusion
Outline of the talk

• Introduction
• Source Data
• Preprocessing
• Alignment
• Conclusion
Parallel corpora (or bitexts) are collections of texts available in two languages.

- Alignment levels: documents, sentences, words
- Crucial for machine translation, but also important for many other language technology tasks (lexicon extraction, multilingual information retrieval, etc.)
- A scarce resource for the vast majority of language pairs and domains!
• **Movie and TV subtitles** are a great resource for compiling parallel corpora:

1. Wide breadth of *linguistic genres*, from colloquial language to narrative and expository discourse.

2. Large databases of subtitles available online (3.6 million on [www.opensubtitles.org](http://www.opensubtitles.org))

3. Tight coupling between subtitles and their "source material" (usually a movie or TV episode)
Introduction

• Earlier versions (2011, 2012, 2013) of the OpenSubtitles collection compiled by Jörg and integrated into OPUS

• Jörg and I worked for the last months on a new, major release of the collection:
  
  • **2.5 billion sentences** (16.2 billion tokens, \( \approx 39\% \) bigger than last version) in **60** languages
  
  • Total of **1,689** bitexts aligned at the sentence level
  
  • Multiple enhancements in the preprocessing and alignment of the subtitles

Link to the OpenSubtitles2016 page
Outline of the talk

- Introduction

- Source Data

- Preprocessing

- Alignment

- Conclusion
The initial dataset

• The administrators of www.opensubtitles.org kindly provided us with a full dump of their database

• 3.36 million subtitle files

• Filtered out languages with < 10 subtitles, resulting in 60 languages

• Each subtitle is associated with:

  • A list of files (may be >1 if multiple CDs),
  • A language code and subtitle format (specified by the uploader)
  • Various information on the source material (movie or TV episode): title, release year, IMDB identifier
  • Details such as the subtitle rating, upload date, nb. of downloads, etc.
## Some statistics (20 biggest languages)

<table>
<thead>
<tr>
<th>Language</th>
<th>Number of files</th>
<th>Number of blocks</th>
<th>Covered IMDBs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>70.1K</td>
<td>53.2M</td>
<td>34.1K</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>95.8K</td>
<td>68.1M</td>
<td>49.3K</td>
</tr>
<tr>
<td>Czech</td>
<td>134K</td>
<td>93.4M</td>
<td>51.3K</td>
</tr>
<tr>
<td>Greek</td>
<td>118K</td>
<td>216M</td>
<td>49.9K</td>
</tr>
<tr>
<td>English</td>
<td>344K</td>
<td>347M</td>
<td>106K</td>
</tr>
<tr>
<td>Spanish</td>
<td>205K</td>
<td>167M</td>
<td>76.1K</td>
</tr>
<tr>
<td>Finnish</td>
<td>46.9K</td>
<td>27.9M</td>
<td>31.8K</td>
</tr>
<tr>
<td>French</td>
<td>110K</td>
<td>200M</td>
<td>56.4K</td>
</tr>
<tr>
<td>Hebrew</td>
<td>85.0K</td>
<td>60.6M</td>
<td>35.6K</td>
</tr>
<tr>
<td>Croatian</td>
<td>106K</td>
<td>64.8M</td>
<td>41.3K</td>
</tr>
<tr>
<td>Hungarian</td>
<td>103K</td>
<td>78.6M</td>
<td>52.7K</td>
</tr>
<tr>
<td>Italian</td>
<td>98.9K</td>
<td>70.5M</td>
<td>41.9K</td>
</tr>
<tr>
<td>Dutch</td>
<td>104K</td>
<td>68.7M</td>
<td>46.6K</td>
</tr>
<tr>
<td>Polish</td>
<td>169K</td>
<td>122M</td>
<td>44.0K</td>
</tr>
<tr>
<td>Portuguese</td>
<td>102K</td>
<td>94.9M</td>
<td>36.2K</td>
</tr>
<tr>
<td>Portuguese (BR)</td>
<td>228K</td>
<td>188M</td>
<td>77.0K</td>
</tr>
<tr>
<td>Romanian</td>
<td>170K</td>
<td>134M</td>
<td>58.1K</td>
</tr>
<tr>
<td>Slovenian</td>
<td>58.6K</td>
<td>37.8M</td>
<td>22.8K</td>
</tr>
<tr>
<td>Serbian</td>
<td>164K</td>
<td>226M</td>
<td>56.3K</td>
</tr>
<tr>
<td>Turkish</td>
<td>181K</td>
<td>115M</td>
<td>55.0K</td>
</tr>
</tbody>
</table>
Subtitle format

• The raw subtitles are structured in *blocks*, which are short text segments associated with a start and end time.

• Time/space constraints: at most 40-50 characters per line, max. of two lines and on-screen display between 1-6 seconds

---

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!

**Note:** No direct, one-to-one correspondence between subtitle blocks and sentences!
Subtitle format

• Many duplicate, spurious or misclassified subtitles

• The subtitle files may be in any character encoding!
  • Only a minority (subtitles uploaded in the last few years) are UTF-8
  • Widespread use of legacy encodings, such as Windows-codepages
  • Most likely encoding must be guessed (more on this later)

• Some subtitles are bilingual (one line per language):

194
00:14:08,700 --> 00:14:09,620
I've got a portfolio.
我说 我已经有作品集了

195
00:14:10,060 --> 00:14:11,100
how can you have a portfolio?
他很惊讶，你已经有作品集了?
Outline of the talk

- Introduction
- Source Data
- **Preprocessing**
- Alignment
- Conclusion
Preprocessing pipeline

Total of **3.36 million** files for 60 languages, in srt and sub formats

- Arabic
- Chinese
- English
- German
- Japanese
- Spanish
- Turkish
- Russian

Metadata from IMDB and opensubtitles.org

- 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

1 XML file per subtitle, encoded as a list of tokenized sentences
Conversion to Unicode

• Most likely encoding must be determined from heuristics.
  • Difficult and error-prone process (no "universal" method)
  • Some languages use more than one script (e.g. Serbian) or are associated with several mutually incompatible encodings (e.g. Russian, Chinese)

• Our solution:
  1. Specify a list of common encodings for each of the 60 languages:
     - Norwegian: UTF-8, Windows-1252, ISO-8859-1
     - Russian: UTF-8, KOI8-R, Windows-1251, MacCyrillic, ISO-8859-5, IBM85(5,6)
     - Chinese (traditional): UTF-8, Big5, GB2312, GB18030, HZ-GB-2312
       ...

  2. When several alternative encodings are possible, we use the chardet library to determine the most likely one

  3. The content is then converted to UTF-8

[Li & Momoi, "A composite approach to language/encoding detection", 19th International Unicode Conference]
Sentence segmentation

• ***Next step***: segment the subtitle blocks into sentences

• Recall the example:
  - First sentence spans blocks 5 & 6
  - Blocks 7 contains 3 sentences

• Main challenges:
  - Approach must scale to the 60 languages of our dataset, many of which do not use western punctuation symbols and conventions
  - User-generated data: cannot rely on strict adherence to spelling conventions (capitalisation, punctuation, etc.)!
Sentence segmentation

• Procedure

1. Each text line is tokenized and processed token-by-token

2. If a sentence-ending marker is detected, we record the tokens on the stack and start a new sentence

3. Upon processing a new block, we check whether it is a continuation of the previous sentence based on the timing and punctuation.

4. The process is repeated for each block in the subtitle

NB: The detection of sentence endings must be language-specific (distinct punctuation marks and conventions, unicameral vs. bicameral alphabets, etc.)!

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!
Tokenisation

• For most languages, we used the tokenizer.perl script from Moses to split the lines into tokens
  • Quite conservative (e.g. no hyphen splitting)

• Language-specific conventions:
  • Split contractions left (English) or right (French, Italian)
  • List of non-breaking prefixes for 22 languages

• For Japanese and Chinese, we relied on the KyTea word segmentation library
  • along with pre-trained models

Correction of OCR errors

- Many subtitles extracted from video streams via error-prone OCR
  - A common error is to confuse the i, I and l characters

- In addition to OCR-generated errors, misplaced accents are also very common
  - For instance, étè instead of été in French

- These errors undermine the quality of the corpora and can cause alignment problems
  - How can we correct these errors in a robust, scalable manner?

4
00:01:34,000 --> 00:01:38,471
<i>"Along with the simplification I sought in my first films,"</i>

5
00:01:39,080 --> 00:01:40,991
<i>"I wanted to be revolutionary,"</i>
Correction of OCR errors

• Developed a custom, bigram-based spellchecker based on the Google Web IT bigrams for 11 European languages:
  • English, Swedish, Spanish, Romanian, Portuguese, Polish, Dutch, Italian, French, German, Czech

• Embarrassingly simple statistical model:

\[
P(w_t | w_t^0, w_{t-1}) = \alpha P(w_t^0 | w_t) P(w_t | w_{t-1})
\]

• The \(w_t\) tokens to consider for each \(w_t^0\) is determined by

  • Enumerating possible confusions between the characters i, I and l
  • Storing all replacements resulting in a token present in the Google unigrams
Correction of OCR errors

• Total of 9.04 million corrected tokens
  • Average of 3.2 corrections per subtitle for the 11 languages
  • 0.5% of out-of-vocabulary tokens

• Current limitations
  • Error model is based on handcrafted heuristics
  • Only works for 11 of the 60 languages in the corpus (but which constitute more than half of the corpus size)
  • And for languages using the Latin script (we have no idea about which character confusions to expect in non-latin scripts)
¿Quiénes lo envían?
- Bueno...

los miembros del club.

Not shown in this example:

-Markup used when token is corrected by spellchecker

-Markup for emphasised tokens/sentences
Inclusion of meta-data

- Last preprocessing step: generate the meta-data
  - Information on the source material: release year, original language, duration and genre of the movie/episode (extracted from IMDB)
  - Subtitle characteristics: language, upload date, rating on opensubtitles.org and subtitle duration
  - Probability that the subtitle language (as specified by the uploader) is correct, based on the langid language identification tool
  - Features of the conversion process, e.g. number of extracted sentences, total number of tokens, number of corrections and file encoding.

- These meta-data are important for quality filtering and for generating alignments between subtitles

Outline of the talk

• Introduction
• Source Data
• Preprocessing
• Alignment
• Conclusion
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

XML subtitle files

... 1 XML file per subtitle, encoded as a list of tokenized sentences

XML alignment files

... 1 XML file per language pair, written as a collection of alignments

Arabic - English

Chinese - Turkish

Spanish - Russian

Arabic - Japanese

Turkish - French

English - Sinhala
First problem: which subtitles to align?

- **Goal**: align subtitles sharing a common *source material* (defined by their IMDB identifier)

- Often many alternative subtitles for a given movie / TV episode!

Given two languages $A$ and $B$, and an IMDB identifier $id$, let $subs_A(id)$ and $subs_B(id)$ denote the set of alternative subtitles available in the two languages

- Aligning each possible pair $(s_1, s_2) \in subs_A(id) \times subs_B(id)$ is not feasible, due to prohibitive computational and storage costs

- Rather, we want to score each subtitle pair $(s_1, s_2)$ according to various measures of the subtitle quality and likelihood of a good alignment
Document alignment

- We designed a handcrafted function to score each subtitle pair \((s_1, s_2)\) based on the following features extracted from the subtitles meta-data:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload date</td>
<td>Newer subtitles are often corrections of previous ones</td>
</tr>
<tr>
<td>Language confidence</td>
<td>Confidence score from the langid identification tool</td>
</tr>
<tr>
<td>User rating</td>
<td>Average of user votes on the website, if they exist</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8 files are less prone to conversion errors</td>
</tr>
<tr>
<td>Spelling errors</td>
<td>Number of corrected and unknown tokens detected by the spellchecking tool</td>
</tr>
<tr>
<td>Subtitle coverage</td>
<td>Time difference between the duration of the source material and the duration of the subtitle</td>
</tr>
<tr>
<td>Subtitle overlap</td>
<td>Time overlap of subtitle frames between (s_1) and (s_2)</td>
</tr>
</tbody>
</table>
Once the subtitle pairs are scored, the top 10 pairs are aligned at the sentence level.

- Alignment performed with a time-overlap algorithm (described in a few slides).

- Measure of synchronization quality: proportion of non-empty alignments relative to the total number of alignments.

Finally, the subtitle pair that maximizes the overall score is selected.

- Weighted sum of the meta-data features + relative proportion of non-empty alignments.
Sentence alignment

• Most approaches to sentence alignment are based on variants of Gale & Church "length-based approach"
  • Relies on a distance measure on the number of words/characters in the paired sentences and prior probabilities on alignment types

• Not working very well for subtitles, due to a larger number of insertions / deletions
  • Large variations in the degree of "compression" of the subtitles, depending on various linguistic and cultural factors

Alternative approach based on time overlaps

[J. Tiedemann (2007), "Improved sentence alignment for movie subtitles", RANLP]
The rich timing information in the subtitles should be exploited!

First problem: do not have time boundaries for each sentence

Must interpolate time values when not explicitly given:

\[
t_{\text{interp}} = t_\leftarrow + \frac{t_\rightarrow - t_\leftarrow}{c_\leftarrow + c_\rightarrow}
\]

- \(t_\leftarrow\): Nearest time event before current position
- \(t_\rightarrow\): Nearest time event after current position
- \(c_\leftarrow\): Nb. characters from \(t_\leftarrow\) to current position
- \(c_\rightarrow\): Nb. characters from \(t_\leftarrow\) to current position
Time-overlap approach

• One run through the subtitle pair (linear time!)
• At each step, find the alignment type with the highest time overlap
• And subsequently move the sliding window until the end
• 7 alignment types: 1:1, 2:1, 0:1, 1:0, 1:2, 1:3 and 3:1

<s id="26">
  <time id="T21S" value="00:02:22,891" />
  Verzeihung.
</s> 00:02:23,504
<s id="27">
  Kann ich Ihnen helfen?
</s> 00:02:24,563
<s id="28">
  Ich bin Frau Cutten.
</s> 00:02:25,511
<s id="29">
  Frau Dr. Cutten.
  <time id="T21E" value="00:02:26,291" />
</s>

<s id="23">
  <time id="T19S" value="00:02:22,880" />
  Unnskyld meg, frøken, kan jeg hjelpe deg?
</s> 00:02:24,712
<s id="24">
  Jeg er Mrs. Cutten.
</s> 00:02:25,549
<s id="25">
  Mrs. Dr. Cutten.
  <time id="T19E" value="00:02:26,282" />
</s>
Time-overlap approach

• Problem: small timing variations (due to differences in frame rate and starting time) across subtitles
  • Small deviations can lead to growing time gaps and poor alignments
• The speed ratio and starting time can be adjusted using anchor points
  • The anchor points correspond to true alignments

\[
\text{speed-ratio} = \frac{t_{\text{trg}(1)} - t_{\text{trg}(2)}}{t_{\text{src}(1)} - t_{\text{src}(2)}}
\]

\[
\text{offset} = t_{\text{trg}(2)} - t_{\text{src}(2)} \times \text{speed-ratio}
\]

\(t_{\text{src}(1)}\) and \(t_{\text{trg}(1)}\): time stamps for the first anchor point (source & target)
\(t_{\text{src}(2)}\) and \(t_{\text{trg}(2)}\): time stamps for the second anchor (source & target)
Time-overlap approach

• How to select the anchor points?
  • Should be far away from one another to get precise estimates for the speed ratio and offset

• Solution: use cognates (similar strings) to identify anchor points
  • Scan the source and target subtitles before alignment to find potential cognates (often proper names)
  • If available, bilingual dictionaries can also be used instead of cognates
  • Not always accurate for distant languages, due to e.g. transliterations
  • Can iterate over multiple candidates for anchor points until we reach a good synchronization (since alignment is quite fast)
Identification of cognates

<document>
....

<s id="2"
  <time id="T3S" value="00:01:21,524" />
  Oh, Déu meu!
</s>

<s id="3"
  <time id="T3E" value="00:01:23,693" />
  Déu meu!
</s>

<s id="4"
  <time id="T4S" value="00:01:29,366" />
  Frank, gràcies a Déu!
</s>

....

<s id="1896"
  <time id="T1550S" value="02:02:34,126" />
  (riu) Jo sé coses dels coloms, Lilly.
</s>

....

</document>

<document>
....

<s id="1"
  <time id="T1S" value="00:01:21,331" />
  Oh, mon Dieu.
  <time id="T1E" value="00:01:23,792" />
</s>

<s id="2"
  <time id="T2S" value="00:01:30,257" />
  Oh, Frank, merci.
  <time id="T2E" value="00:01:32,467" />
</s>

....

<s id="1528"
  <time id="T1356S" value="02:02:35,339" />
  Sur les pigeons, je me trompe rarement, Lilly.
</s>

....

</document>
Alignment format

Alignment = sequence of alignment units X;Y where

- X is a (possibly empty) list of contiguous sentence indices in the source
- Y is a (possibly empty) list of contiguous sentences indices in the target

```
<linkGrp targType="s"
fromDoc="de/1925/16641/3174032.xml.gz" toDoc="no/1925/16641/3260527.xml.gz">
  <link id="SL0" xtargets="1;1" overlap="0.980" />
  <link id="SL1" xtargets="2 3;2" overlap="0.937" />
  <link id="SL2" xtargets="4 5;3" overlap="0.930" />
  <link id="SL3" xtargets="6;4" overlap="0.956" />
  <link id="SL4" xtargets="7;5" overlap="0.957" />
  <link id="SL5" xtargets="8;6 7" overlap="0.893" />
  <link id="SL6" xtargets="9 10;8" overlap="0.957" />
  <link id="SL7" xtargets="11;9" overlap="0.866" />
  ....
</linkGrp>
```
Some statistics (20 biggest bitexts)

<table>
<thead>
<tr>
<th>Language pair</th>
<th>Aligned documents</th>
<th>Sentence pairs</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>English - Spanish</td>
<td>62.2K</td>
<td>50.1M</td>
<td>620.0M</td>
</tr>
<tr>
<td>English - Portuguese (BR)</td>
<td>61.1K</td>
<td>47.6M</td>
<td>587.4M</td>
</tr>
<tr>
<td>Spanish - Portuguese (BR)</td>
<td>56.3K</td>
<td>43.0M</td>
<td>521.1M</td>
</tr>
<tr>
<td>English - Romanian</td>
<td>48.8K</td>
<td>39.5M</td>
<td>484.5M</td>
</tr>
<tr>
<td>English - Turkish</td>
<td>47.4K</td>
<td>37.3M</td>
<td>404.9M</td>
</tr>
<tr>
<td>Spanish - Romanian</td>
<td>45.5K</td>
<td>35.1M</td>
<td>431.1M</td>
</tr>
<tr>
<td>Portuguese (BR) - Romanian</td>
<td>45.5K</td>
<td>34.7M</td>
<td>422.2M</td>
</tr>
<tr>
<td>English - Serbian</td>
<td>44.1K</td>
<td>34.6M</td>
<td>411.6M</td>
</tr>
<tr>
<td>English - Hungarian</td>
<td>44.7K</td>
<td>33.6M</td>
<td>381.4M</td>
</tr>
<tr>
<td>English - French</td>
<td>42.6K</td>
<td>33.5M</td>
<td>432.1M</td>
</tr>
<tr>
<td>Spanish - Turkish</td>
<td>44.1K</td>
<td>33.1M</td>
<td>358.1M</td>
</tr>
<tr>
<td>Portuguese (BR) - Turkish</td>
<td>43.9K</td>
<td>32.7M</td>
<td>348.9M</td>
</tr>
<tr>
<td>Spanish - Serbian</td>
<td>40.6K</td>
<td>30.5M</td>
<td>362.1M</td>
</tr>
<tr>
<td>Greek - English</td>
<td>36.6K</td>
<td>30.4M</td>
<td>376.8M</td>
</tr>
<tr>
<td>Portuguese (BR) - Serbian</td>
<td>40.6K</td>
<td>30.2M</td>
<td>353.6M</td>
</tr>
<tr>
<td>Czech - English</td>
<td>44.3K</td>
<td>27.5M</td>
<td>367.5M</td>
</tr>
<tr>
<td>Bulgarian - English</td>
<td>41.5K</td>
<td>26.4M</td>
<td>362.4M</td>
</tr>
<tr>
<td>Czech - Portuguese (BR)</td>
<td>41.9K</td>
<td>26.0M</td>
<td>331.4M</td>
</tr>
<tr>
<td>Czech - Spanish</td>
<td>41.7K</td>
<td>25.9M</td>
<td>338.0M</td>
</tr>
<tr>
<td>Romanian - Turkish</td>
<td>39.9K</td>
<td>25.5M</td>
<td>305.9M</td>
</tr>
</tbody>
</table>
## BLEU score improvements

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>2012+213</th>
<th>2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish-English</td>
<td>28.57</td>
<td>31.22</td>
</tr>
<tr>
<td>English-Spanish</td>
<td>26.34</td>
<td>29.83</td>
</tr>
<tr>
<td>English-French</td>
<td>22.68</td>
<td>25.34</td>
</tr>
<tr>
<td>French-English</td>
<td>24.09</td>
<td>26.31</td>
</tr>
<tr>
<td>Polish-English</td>
<td>21.78</td>
<td>24.50</td>
</tr>
<tr>
<td>English-Polish</td>
<td>18.07</td>
<td>20.96</td>
</tr>
<tr>
<td>Russian-English</td>
<td>21.62</td>
<td>25.11</td>
</tr>
<tr>
<td>English-Russian</td>
<td>14.13</td>
<td>15.91</td>
</tr>
<tr>
<td>Arabic-English</td>
<td>24.39</td>
<td>25.80</td>
</tr>
<tr>
<td>Arabic-French</td>
<td>11.14</td>
<td>17.28</td>
</tr>
<tr>
<td>English-Arabic</td>
<td>8.38</td>
<td>9.87</td>
</tr>
<tr>
<td>Portuguese (BR)-English</td>
<td>31.10</td>
<td>32.90</td>
</tr>
<tr>
<td>Chinese-English</td>
<td>11.40</td>
<td>17.18</td>
</tr>
<tr>
<td>French-Czech</td>
<td>12.59</td>
<td>17.44</td>
</tr>
<tr>
<td>Czech-English</td>
<td>27.24</td>
<td>28.68</td>
</tr>
<tr>
<td>Greek-English</td>
<td>25.06</td>
<td>28.10</td>
</tr>
<tr>
<td>German-English</td>
<td>23.33</td>
<td>24.35</td>
</tr>
<tr>
<td>German-Norwegian</td>
<td>20.93</td>
<td>31.69</td>
</tr>
</tbody>
</table>
Intra-lingual alignments

- In addition to inter-lingual alignments, we can also use the subtitles to produce intra-lingual alignments
  - Alignments between alternative subtitles of a given source material
  - Effectively create a *fully connected multilingual corpus*

- Intra-lingual alignments are useful for various purposes:
  - Detect errors in the subtitles (spelling and conversion errors, etc.)
  - Discover insertions and deletions (expressing e.g. extra-linguistic information)
  - Extract paraphrases and translation alternatives
Intra-lingual alignments

1 XML file per subtitle, encoded as a list of tokenized sentences

Find alternative subtitles for each movie / TV episode

Compute alignments for the alternative subtitles

Sort the alignments

Generate alignment files

1 XML file per language, written as a collection of alignments

XML subtitle files

XML alignment files

Arabic

Chinese

Spanish

Turkish

French

English
Intra-lingual alignments

• Alignment also based on time overlaps
  • with a BLEU filter and search heuristics to improve the alignment quality

• The alignments can be classified into 4 categories
  • Insertions: Some sentences are identical except for some inserted text (words or phrases).
  • Punctuation: Sentence pairs that only differ in their use of punctuation and/or white-spaces.
  • Spelling: Minor differences in a few words in otherwise identical sentences.
  • Paraphrases: Sentence pairs that use paraphrased expressions or are substantially different from each other

Outline of the talk

- Introduction
- Source Data
- Preprocessing
- Alignment

- Conclusion
Full pipeline

Raw subtitle files

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

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

XML 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

Bitexts (Moses format)

1 XML file per language pair, encoded as a collection of alignments

Total of 3.36 million files for 60 languages, in srt and sub formats

Metadata from IMDB and opensubtitles.org
Conclusion

• We presented a new major release of the OpenSubtitles collection of parallel corpora
  • 2.5 billion sentences (16.2 billion tokens) in 60 languages
  • The world's largest multilingual corpus currently available?

• Complex processing pipeline:
  • Preprocessing to convert the subtitle blocks into tokenised sentences.
  • Inter- and intra-lingual alignments based on time overlaps

• Freely available on the OPUS website:
  
  http://opus.lingfil.uu.se/OpenSubtitles2016.php