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

Introduction

- 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!

Introduction

- **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.

捷奏动漫
All right, do your stuff!
好，一展身手吧!

2. Large databases of subtitles available online (3.6 million on 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, ≈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)

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

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):

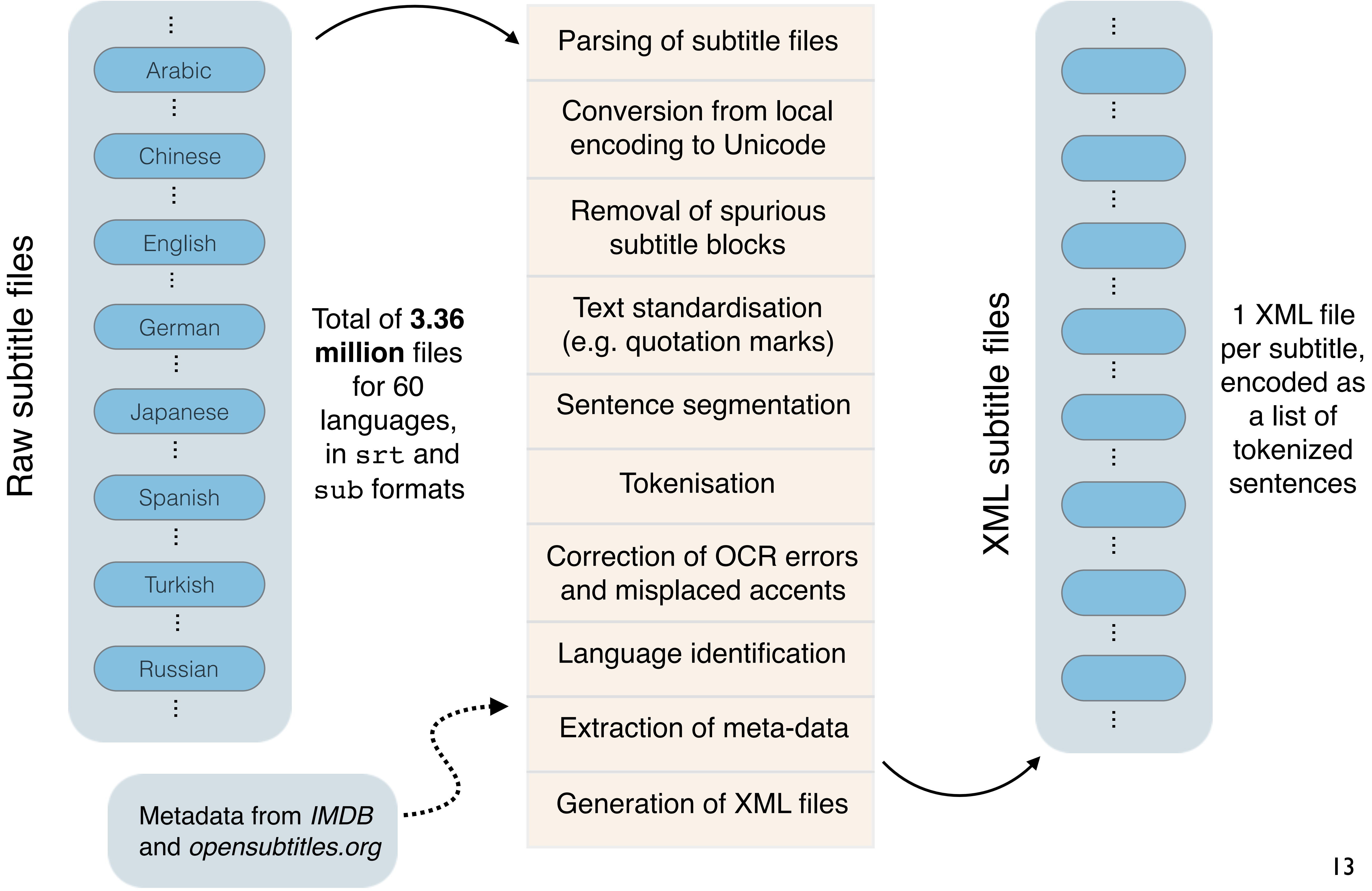
```
194
00:14:08,700 --> 00:14:09,620
I've got a portfolio.
我说 我已经有作品集了

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



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*

- 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

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

```
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!
```

- 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

```
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!
```

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

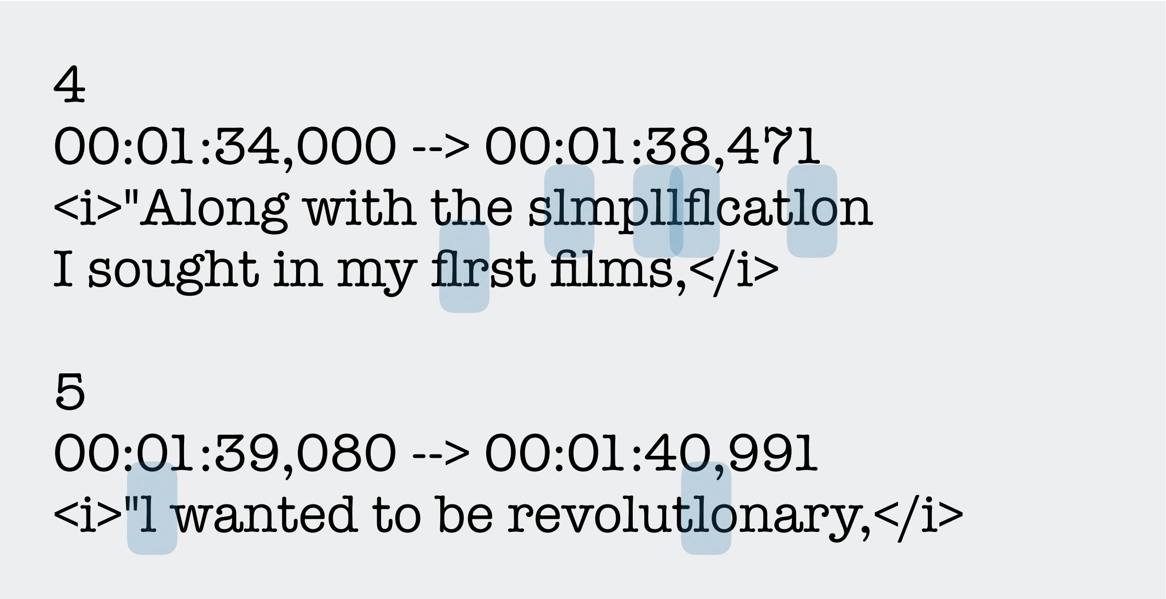
Tokenisation

- 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

[Neubig et al, "Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis", *ACL 2011*]

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



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

5
00:01:39,080 --> 00:01:40,991
<i>"l wanted to be revolutlonary,</i>
```

- 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?

Correction of OCR errors

- Developed a custom, bigram-based spellchecker based on the **Google Web 1T** bigrams for 11 European languages:
  - English, Swedish, Spanish, Romanian, Portuguese, Polish, Dutch, Italian, French, German, Czech
- Embarrassingly simple statistical model:

$$P(\underbrace{w_t}_{\text{Actual (corrected) token}}|\underbrace{w_t^o}_{\text{Observed (noisy) token}},\underbrace{w_{t-1}}_{\text{Previous token}}) = \alpha \underbrace{P(w_t^o|w_t)}_{\text{Error model (heuristics)}}\underbrace{P(w_t|w_{t-1})}_{\text{Bigram model}}$$

- The w_t tokens to consider for each w_t^o 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)

Output format

```
1243
02:36:31,524 --> 02:36:34,220
- ¿Quiénes lo envían?
- Bueno...

1244
02:36:34,294 --> 02:36:36,262
los miembros del club.
```

```
<s id="1396">
   <time id="T1243S" value="02:36:31,524" />
   <w id="1396.1">-</w>
   <w id="1396.2">¿</w>
   <w id="1396.3">Quiénes</w>
   <w id="1396.4">lo</w>
   <w id="1396.5">envían</w>
   <w id="1396.6">?</w>
</s>
<s id="1397">
   <w id="1397.1">-</w>
   <w id="1397.2">Bueno</w>
   <w id="1397.3">...</w>
   <time id="T1243E" value="02:36:34,220" />
   <time id="T1244S" value="02:36:34,294" />
   <w id="1397.4">los</w>
   <w id="1397.5">miembros</w>
   <w id="1397.6">del</w>
   <w id="1397.7">club</w>
   <w id="1397.8">.</w>
   <time id="T1244E" value="02:36:36,262" />
</s>
```

Not shown in this example:

- Markup used when token is corrected by spellchecker
- Markup for emphasised tokens/sentences

Inclusion of meta-data

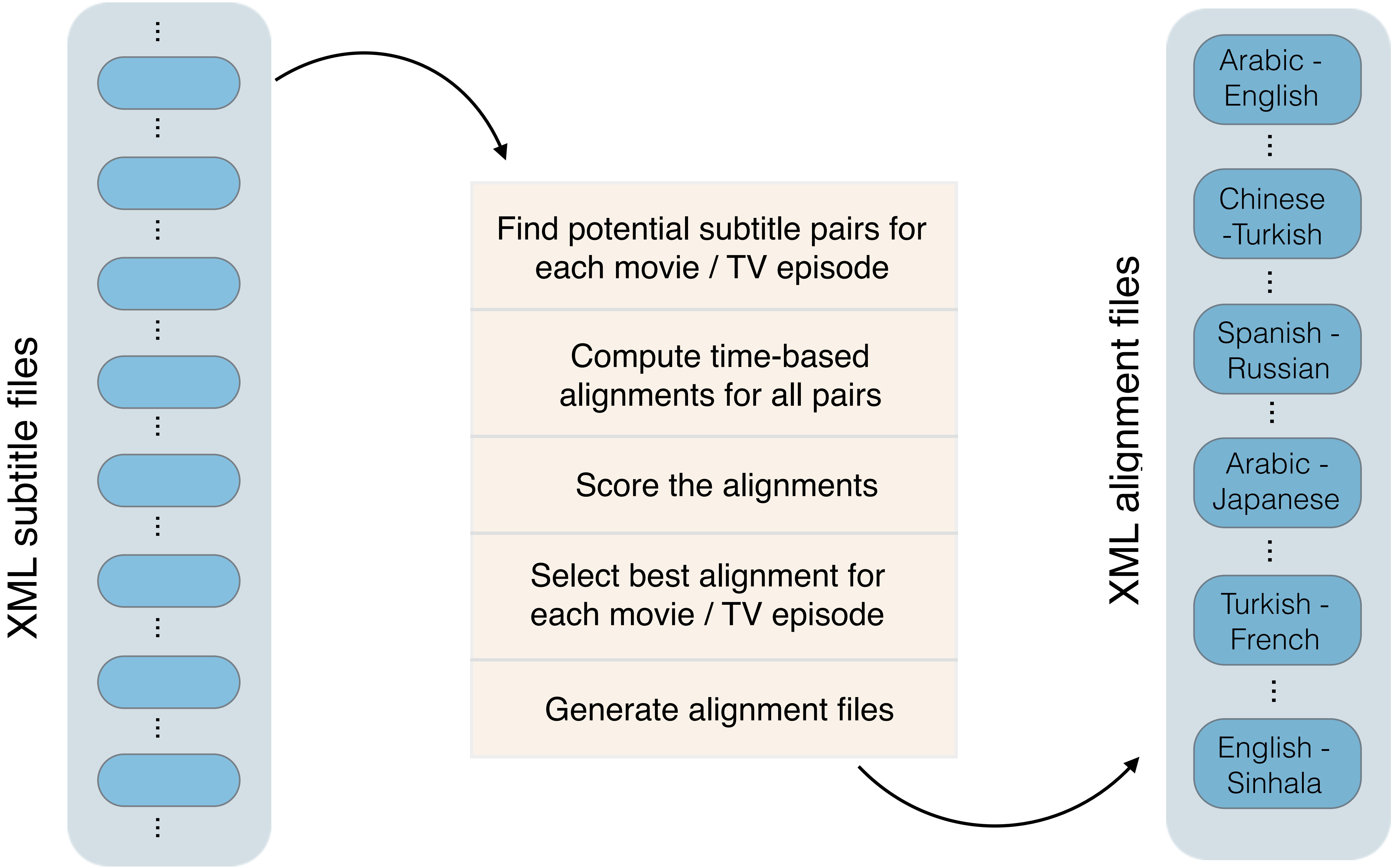
- Last preprocessing step: generate the meta-data
  - Information on the source material: release year, original language, duration and genre of the movie/episode (extracted from IMDB)
  - Subtitle characteristics: language, upload date, rating on opensubtitles.org and subtitle duration
  - Probability 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

[Lui & Baldwin (2012) langid.py: An Off-the-shelf Language Identification Tool, *ACL 2012*]

Outline of the talk

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

Alignment



XML subtitle files

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

Arabic - English

Chinese -Turkish

Spanish - Russian

Arabic - Japanese

Turkish - French

English - Sinhala

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

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

Document alignment

- 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:

Upload date	Newer subtitles are often corrections of previous ones
Language confidence	Confidence score from the langid identification tool
User rating	Average of user votes on the website, if they exist
Encoding	UTF-8 files are less prone to conversion errors
Spelling errors	Number of corrected and unknown tokens detected by the spellchecking tool
Subtitle coverage	Time difference between the duration of the source material and the duration of the subtitle
Subtitle overlap	Time overlap of subtitle frames between s_1 and s_2

Document alignment

- 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*

[W. Gale & K. Church (1993), "A program for aligning sentences in bilingual corpora", *Computational Linguistics*]

[J. Tiedemann (2007), "Improved sentence alignment for movie subtitles", *RANLP*]

Sentence alignment

- 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:

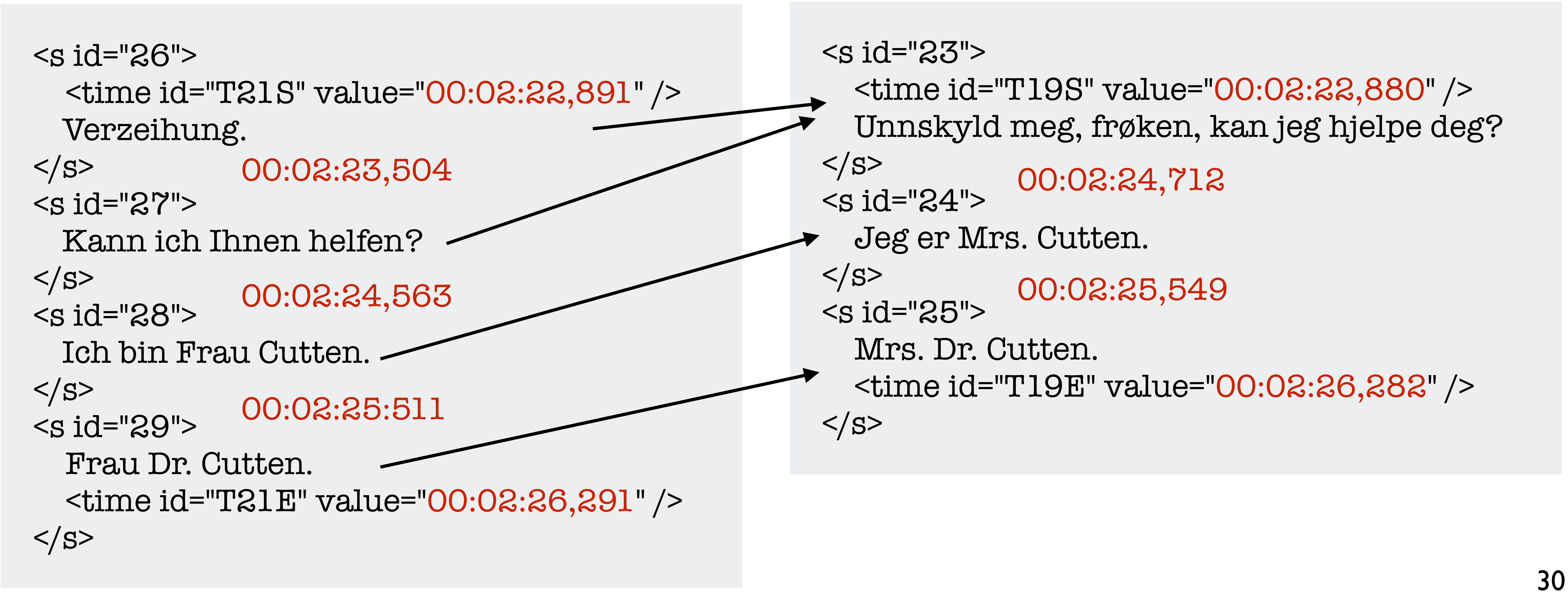
$$t_{interpol} = t_{\leftarrow} + c_{\leftarrow} \frac{t_{\rightarrow} - t_{\leftarrow}}{c_{\leftarrow} + c_{\rightarrow}}$$

$t_{\leftarrow}$	Nearest time event before current position
$t_{\rightarrow}$	Nearest time event after current position
$c_{\leftarrow}$	Nb. characters from t← to current position
$c_{\rightarrow}$	Nb. characters from t← to current position

```
<s id="168">
   <time id="T181S" value="00:10:15,705" />
   <w id="168.1">Ну</w>
   <w id="168.2">,</w>
   <w id="168.3">голова</w>
   <w id="168.4">и</w>
   <w id="168.5">конечности</w>
   <w id="168.6">разлетелись</w>
   <w id="168.7">,</w>
   <time id="T181E" value="00:10:17,566" />
   <time id="T182S" value="00:10:17,686" />
   <w id="168.8">но</w>
   <w id="168.9">мы</w>
   <w id="168.10">смогли</w>
   <w id="168.11">восстановить</w>
   <w id="168.12">половину</w>
   <w id="168.13">тела</w>
   <w id="168.14">.</w>
 </s>                                  00:10:19,912
 <s id="169">
   <w id="169.1">Удачный</w>
   <w id="169.2">день</w>
   <w id="169.3">)</w>
   <time id="T182E" value="00:10:20,675" />
 </s>
```

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

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

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

$t_{src(1)}$ and $t_{trg(1)}$: time stamps for the first anchor point (source & target)
$t_{src(2)}$ and $t_{trg(2)}$: 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">
  Déu meu!
  <time id="T3E" value="00:01:23,693" />
 </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.
 <time id="T1550E" value="02:02:37,755" />
</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.
  <time id="T1356E" value="02:02:37,758" />
 </s>

</document>
```

Alignment format

```
<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>
```

- 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

Some statistics (20 biggest bitexts)

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

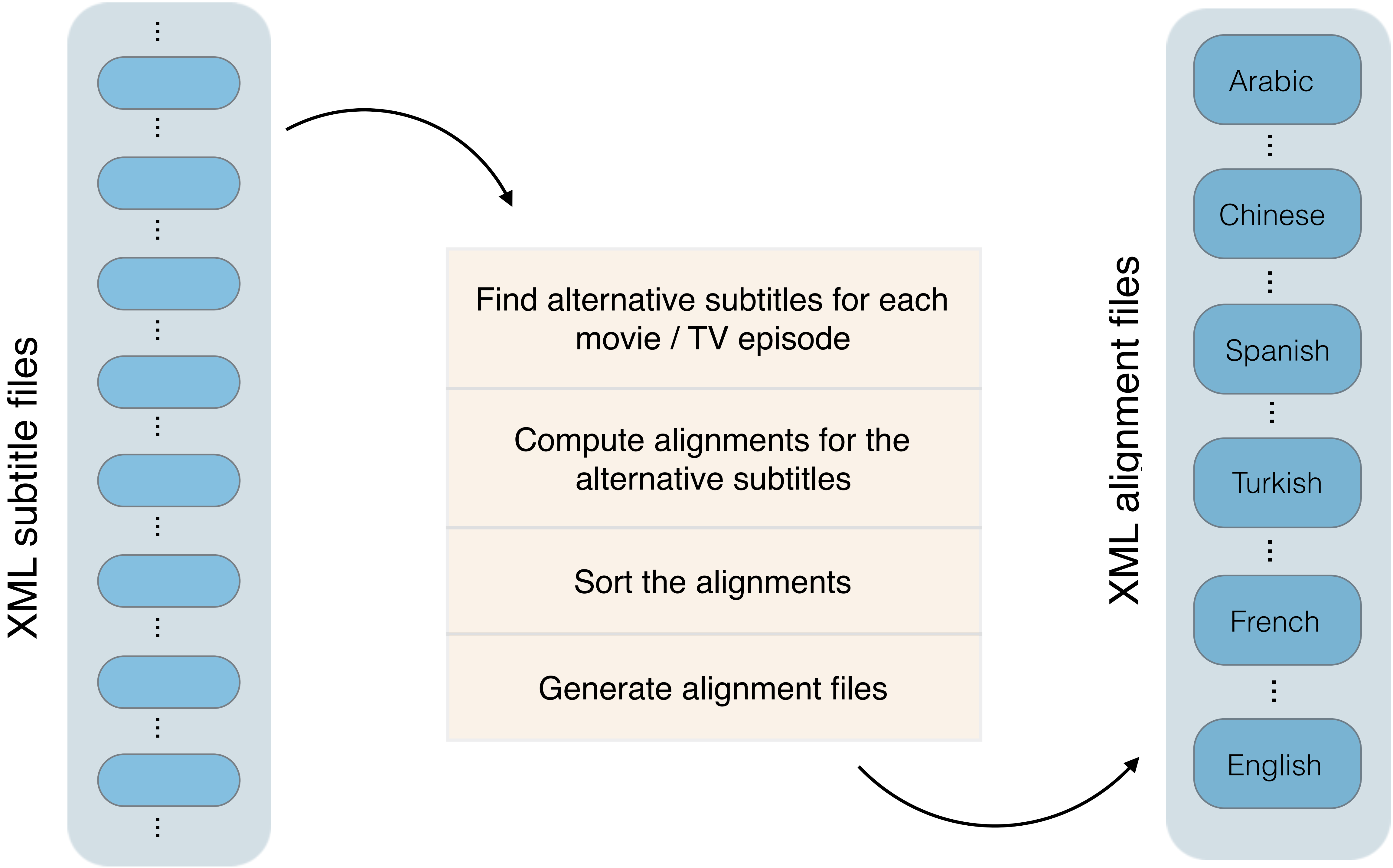
BLEU score improvements

	2012+213	2016
Spanish-English	28.57	**31.22**
English-Spanish	26.34	**29.83**
English-French	22.68	**25.34**
French-English	24.09	**26.31**
Polish-English	21.78	**24.50**
English-Polish	18.07	**20.96**
Russian-English	21.62	**25.11**
English-Russian	14.13	**15.91**
Arabic-English	24.39	**25.80**
Arabic-French	11.14	**17.28**
English-Arabic	8.38	**9.87**
Portuguese (BR)-English	31.10	**32.90**
Chinese-English	11.40	**17.18**
French-Czech	12.59	**17.44**
Czech-English	27.24	**28.68**
Greek-English	25.06	**28.10**
German-English	23.33	**24.35**
German-Norwegian	20.93	**31.69**

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



XML subtitle files

Find alternative subtitles for each movie / TV episode

Compute alignments for the alternative subtitles

Sort the alignments

Generate alignment files

XML alignment files

Arabic

Chinese

Spanish

Turkish

French

English

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

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

Intra-lingual alignments

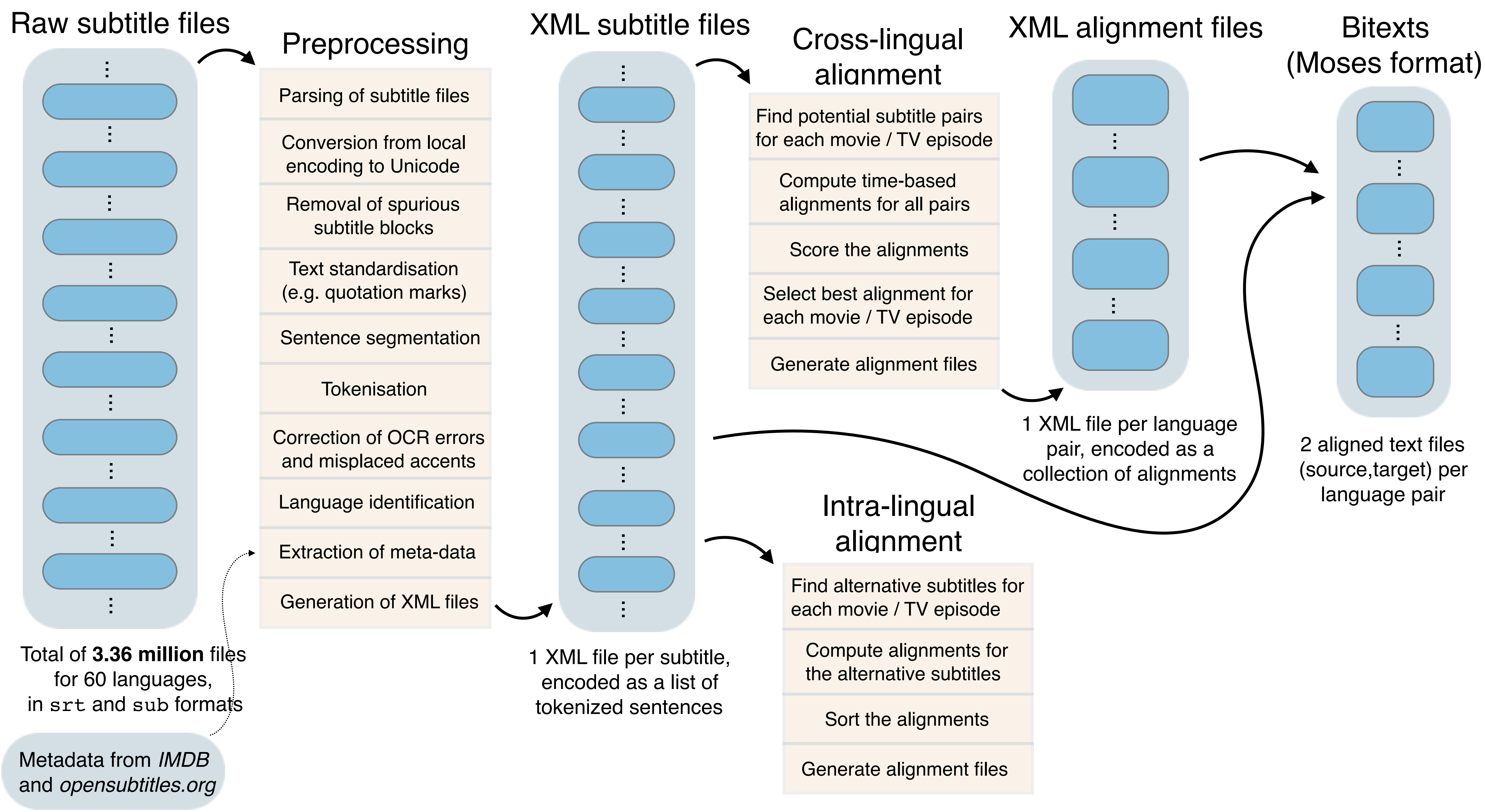
- Alignment also based on time overlaps
  - with a BLEU filter and search heuristics to improve the alignment quality
- The alignments can be classified into 4 categories
  - *Insertions*: Some sentences are identical except for some inserted text (words or phrases).
  - *Punctuation*: Sentence pairs 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

[J. Tiedemann (2016), "Finding Alternative Translations in a Large Corpus of Movie Subtitles ", *LREC*]

Outline of the talk

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

Full pipeline



Raw subtitle files

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

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

Bitexts (Moses format)

2 aligned text files (source,target) per language pair

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