Automatic turn segmentation for movie and TV subtitles

Pierre Lison, NR
plison@nr.no
(joint work with Raveesh Meena, KTH)

LTG seminar, UiO
29/11/2016
Introduction

► Movie & TV subtitles are a great resource for NLP:
  - **Linguistic perspective:** Broad spectrum of linguistic genres & speaker styles (including colloquial language), non-sentential utterances, complex conversational structures, etc.
  - **Data-driven perspective:** Huge amounts of available data (and meta-data), covering many languages (2.8M subtitles in 60 languages in OpenSubtitles 2016)
Introduction

► Resources from movie and TV subtitles are already used for various NLP tasks:
  ▪ Language modelling
  ▪ Machine translation
  ▪ Multilingual and cross-lingual NLP
  ▪ Conversation modelling & dialogue systems

[0x0] [0x0] [0x0] [0x0]

[0x0] [0x0] [0x0] [0x0]

► However, they lack a crucial piece of information: the turn structure
  ▪ Who is speaking at a given time?
# Introduction

<table>
<thead>
<tr>
<th>ID</th>
<th>Utterance</th>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If we wanted to kill you, Mr Holmes, we would have done it by now.</td>
<td>01:17:34.76</td>
<td>01:17:37.75</td>
</tr>
<tr>
<td>2</td>
<td>We just wanted to make you inquisitive.</td>
<td>01:17:37.80</td>
<td>01:17:40.59</td>
</tr>
<tr>
<td>3</td>
<td>Do you have it?</td>
<td>01:17:42.40</td>
<td>01:17:43.91</td>
</tr>
<tr>
<td>4</td>
<td>Do I have what?</td>
<td>01:17:43.91</td>
<td>01:17:45.43</td>
</tr>
<tr>
<td>5</td>
<td>The treasure.</td>
<td>01:17:45.48</td>
<td>01:17:46.43</td>
</tr>
<tr>
<td>6</td>
<td>I don't know what you're talking about.</td>
<td>01:17:46.43</td>
<td>01:17:48.91</td>
</tr>
<tr>
<td>7</td>
<td>I would prefer to make certain.</td>
<td>01:17:48.96</td>
<td>01:17:52.03</td>
</tr>
<tr>
<td>8</td>
<td>Everything in the West has its price.</td>
<td>01:17:57.00</td>
<td>01:17:59.63</td>
</tr>
<tr>
<td>9</td>
<td>And the price for her life - information.</td>
<td>01:17:59.68</td>
<td>01:18:04.55</td>
</tr>
</tbody>
</table>

**Question**: can we automatically segment this dialogue into turns? (without having access to the audiovisual material)
Key idea

► Subtitles do not contain speaker information…

► But movie and TV scripts (screenplays, transcripts, etc.) do!

► Outline of approach:
  1. Align the subtitles with movie and TV scripts
  2. Use alignments to project speaker information on the subtitles
  3. Use the subtitles augmented with speaker information to train a classifier that detects turn boundaries
Step 1: Alignment with movie and TV scripts
OpenSubtitles 2016

- Earlier this year, Jörg Tiedemann and I released a new major version of the OpenSubtitles corpus

- **What is it?**
  - Collection of 2.8M subtitles from [www.opensubtitles.org](http://www.opensubtitles.org)
  - 2.6 G sentences, 17.2 G tokens
  - 60 languages aligned at sentence-level (1689 bitexts)

- **Preprocessing steps:**
  1. Conversion
  2. Sentence segmentation
  3. Tokenisation
  4. Correction of OCR and spelling errors
  5. Extraction of meta-data

![Diagram showing initial subtitle files (.srt format) converted to XML files (list of tokenised sentences), with IMDB and sentence alignments (inter, intra-linguál).]
Movie & TV scripts

► We crawled various websites hosting movie and TV scripts
  ▪ Scrapped them to extract the sequence of dialogue turns
  ▪ Result: total of 7,467 of dialogue transcripts

► NB: dialogues from screenplays can be very different from those found in the subtitles!
Alignment

- We can then align the subtitles with the movie scripts
  - One alignment for each <subtitle,script> pair
  - We used both hunalign and bleualign
Alignment results

- 3,864,058 sentence pairs
  - 34% of the sentences for movies, 60% for TV episodes

- Quality of the alignments?
  - hunalign and bleualign were quite consistent (only 0.3% of conflicting alignments)
  - Comparison with a small, manually annotated corpus of TV series: 97.6% of the projected speaker labels matched the manually labelled ones

- We also projected the speaker information onto 6 other languages (using the bitexts from [Lison and Tiedemann, 2016])
Step 2: Turn segmentation
Taking stock

► Where are we?
  ▪ Thanks to the alignments, we now have a subset of subtitles where a fraction of sentences are annotated with speaker information (speaker label + turn boundaries)

► What do we want?
  ▪ A classifier that detects turn boundaries, using only textual and timing features from the subtitles themselves

► Binary classifier: given two consecutive sentences, predicts the presence of a turn boundary between them
Training data

- We extracted all consecutive sentence pairs in the subtitles that were annotated with speaker information
  - Total of 1,521,382 sentence pairs
  - Divided in training (60%), dev (20%) and test (20%) sets

- Binary scheme:
  - If the sentence i and i+1 have the same speaker and are part of the same turn in the script, mark it as "same turn"
  - Otherwise, mark it as "new turn"

- Quite balanced dataset: 52.3 % of "new turn" pairs
Classifier

- **Goal**: train a binary classifier that, given a pair of two consecutive sentences, outputs the probability of a turn boundary between them

- We used a linear discriminative classifier for this task
  - Using *Vowpal Wabbit*, a high-performance linear classifier

- Which features to use?
  - Various linguistic markers can be useful
  - For instance, adjacency pairs (such as question/answer) often denote a turn change
  - Another example: reuse of same pronoun as subject in the two sentences often denote a turn continuation
## Features

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timing features:</strong></td>
<td><em>Time gaps and sentence durations</em></td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td><em>Nb. of characters/tokens in each sentence</em></td>
</tr>
<tr>
<td><strong>Lexical features:</strong></td>
<td><em>BoW, bigrams, occurrence of negation/question words, pronouns</em></td>
</tr>
<tr>
<td><strong>POS features</strong></td>
<td><em>POS tags and sequences, likely imperative mood (VB before NN or PRP and no question mark)</em></td>
</tr>
<tr>
<td><strong>Punctuation features:</strong></td>
<td><em>Marks at start/end of each sentence</em></td>
</tr>
<tr>
<td><strong>Edit distance features</strong></td>
<td><em>Token-level dist. between the two sentences</em></td>
</tr>
<tr>
<td><strong>Adjacency features</strong></td>
<td><em>Occurrence of specific patterns, such as</em></td>
</tr>
<tr>
<td></td>
<td>• Likely polar answer</td>
</tr>
<tr>
<td></td>
<td>• Likely clarification request</td>
</tr>
<tr>
<td></td>
<td>• Pronoun inversion</td>
</tr>
<tr>
<td><strong>Global features</strong></td>
<td><em>Occurrence of character names, movie genre, sentence/token density, sentence number</em></td>
</tr>
<tr>
<td><strong>Alignment features</strong></td>
<td><em>Proportion of inter- and intra-lingual alignments in the OpenSubtitles bitexts.</em></td>
</tr>
<tr>
<td><strong>&quot;Visual&quot; features</strong></td>
<td><em>Start/end of subtitle block</em></td>
</tr>
</tbody>
</table>

(Alignments of type 2:1 are much more likely to occur if the two sentences are from the same speaker.)
Extension 1: multilingual classifier

- We also have speaker annotations for non-English subtitles
  - Can we use them to further improve the classification?
  - Useful markers of turn change might be absent in a particular language but present in another one.

- We combine all classifiers in a weighted sum:

\[
P_{\text{multiling}}(\text{turn}|s_{i-1}, s_i) =
\alpha \left[ P_L(\text{turn}|s_{i-1}, s_i) + \sum_{L'} \sum_{L'} w_{L'} P_{L'}(\text{turn}|s_{j-1}, s_j) \right]
\]

- Probability of turn boundary for English sentence pair
- Probability of turn boundary for sentence pair in language L
Extension 2: speaker diarization

- When the corresponding audiovisual material is available, we can also exploit it to improve the segmentation.
- More precisely, we can apply speaker diarization techniques to segment the audio stream into clusters.
- Again, we integrate the result in a weighted sum:

\[
P_{\text{Classif+Dia}}(\text{turn=\text{same}}|s_{i-1}, s_i) = \\
\alpha \left[ P(\text{turn=\text{same}}|s_{i-1}, s_i) + w_{\text{Dia}} \mathbb{1}(C(s_{i-1}) = C(s_i)) \right]
\]

\[
P_{\text{Classif+Dia}}(\text{turn=\text{new}}|s_{i-1}, s_i) = \\
\alpha \left[ P(\text{turn=\text{new}}|s_{i-1}, s_i) + w_{\text{Dia}} \mathbb{1}(C(s_{i-1}) \neq C(s_i)) \right]
\]

Indicator function (1 if the \(s_{i-1}\) and \(s_i\) are part of the same diarization cluster, else 0)
Step 3: Experimental results
Experiments

► Baseline:
  ▪ If second sentence starts with a “-” dash → new turn
  ▪ Otherwise, if the time gap is exactly zero → same turn
  ▪ Else, → new turn (majority class in this context)

► And 3 alternative approaches:
  ▪ Basic discriminative classifier
  ▪ Ensemble of multilingual classifiers (extension 1)
  ▪ Classifier with speaker diarization (extension 2)

For the speaker diarization, we extracted the audio of one season (21 episodes of ~ 40 minutes each) of the “One Tree Hill” TV series, and applied the LIUM diarization toolkit on the data.
## Results

<table>
<thead>
<tr>
<th>Approach</th>
<th>Turn</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$P$</td>
<td>$R$</td>
</tr>
<tr>
<td>Baseline</td>
<td>Same</td>
<td>0.48</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.81</td>
<td>0.98</td>
</tr>
<tr>
<td>Classifier (basic)</td>
<td>Same</td>
<td>0.80</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.78</td>
<td>0.84</td>
</tr>
<tr>
<td>Classifier (multiling)</td>
<td>Same</td>
<td>0.80</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.79</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Accuracy, precision, recall and F1 scores based on the development set (197K sentence pairs) and test set (200K sentence pairs). The best results are written in bold and are all statistically significant using a bootstrap test (p-values < 0.0001)
Results

Accuracy, precision, recall and F1 scores on the small "Tree Hill" dataset. The best result is statistical significant with p-value = 0.013

<table>
<thead>
<tr>
<th>Approach</th>
<th>Turn</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Same</td>
<td>0.32</td>
<td>0.22</td>
<td>0.26</td>
<td>0.595</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.75</td>
<td>1.00</td>
<td>0.85</td>
<td></td>
</tr>
<tr>
<td>Classifier (basic)</td>
<td>Same</td>
<td>0.85</td>
<td>0.68</td>
<td>0.76</td>
<td>0.774</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.72</td>
<td>0.87</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>Classifier (multiling)</td>
<td>Same</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Diarization only</td>
<td>Same</td>
<td>0.75</td>
<td>0.39</td>
<td>0.51</td>
<td>0.617</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.57</td>
<td>0.86</td>
<td>0.69</td>
<td></td>
</tr>
<tr>
<td>Classifier+Diarization</td>
<td>Same</td>
<td>0.85</td>
<td>0.68</td>
<td>0.76</td>
<td>0.775*</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>0.72</td>
<td>0.87</td>
<td>0.79</td>
<td></td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Language</th>
<th>Baseline</th>
<th>Classifier (basic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>0.588</td>
<td>0.716</td>
</tr>
<tr>
<td>French</td>
<td>0.663</td>
<td>0.743</td>
</tr>
<tr>
<td>German</td>
<td>0.656</td>
<td>0.741</td>
</tr>
<tr>
<td>Czech</td>
<td>0.668</td>
<td>0.756</td>
</tr>
<tr>
<td>Turkish</td>
<td>0.662</td>
<td>0.758</td>
</tr>
<tr>
<td>Chinese</td>
<td>0.569</td>
<td>0.670</td>
</tr>
</tbody>
</table>

Table 3. Compared accuracies for the baseline and classifier for 6 non-English languages (test set).

NB: Some features (e.g. adjacency features) were not present for these languages.
Discussion

- The 3 approaches outperform the baseline, but the results are far from perfect
  - Is this the result of a bad classification model
  - ... or of the inherent difficulty of the task?
- Small-scale annotation experiment with 3 annotators
  - The annotators were shown 100 sentence pairs, together with their associated start and end times.
  - Fleiss' kappa of 0.35 ("fair" agreement) among the three annotators and the "gold standard" from the script
  - Classification accuracy not better than the baseline (68%, 72%, 65% respectively)

But they ignored the timing information, which is often crucial to detect turn boundaries
Step 4: Conclusion
Conclusion

► Two contributions:
  ▪ An extension of the OpenSubtitles dataset with speaker information extracted from movie & TV scripts
  ▪ A data-driven approach to the segmentation of subtitles into dialogue turns, based on linguistic and timing features

► Although the approach focused on subtitles, it can easily be adapted to other types of dialogue transcripts.

► Future work:
  ▪ More advanced segmentation approach? Neural architectures, structured prediction, etc.
  ▪ Use of the resulting turn structure for downstream tasks