ParlaCLARIN Workshop: Creating and Using Parliamentary Corpora

Miyazaki, 7 May 2018

EuroParl-UdS: Preserving and Extending Metadata in Parliamentary Debates

Alina Karakanta, Mihaela Vela, Elke Teich

Department of Language Science and Technology, Saarland University

Outline

- Introduction
- Motivation
- EuroParl-Uds
- Corpus processing
  - Metadata – proceedings
  - Metadata – MEPs
  - Sorting and alignment
- Corpus structure
- Corpus statistics
- Possible applications

Introduction

Parliamentary corpora as a rich high-quality resource for linguistic applications

Metadata  Structure, organise, filter

Related projects:

- Europarl (Koehn, 2005)
- Corrected and Structured EuroParl corpus (Gräen et al., 2014)
- European Comparable and Parallel Corpora (Calzada Perez et al., 2006)
- Digital Corpus of the European Parliament (Hajlaouiet al., 2014)
- Talk of Europe – Travelling CLARIN Campus/LinkedEP (van Aggelen et al., 2017)

Motivation

- Machine translation
- Gender identification (Koppel et al., 2002)
- Topic detection (Yang et al., 2011; Blei, 2012)
- Translation studies (translationese)
  - Status: Original vs. Translation (Rabinovich et al., 2015)
  - Producer: Native vs. Non-native (Nisioi et al., 2016)

EuroParl-UdS

- **Parallel corpora** where the source language (SL) sentences come from native SL speakers and are aligned to sentences in the required target language (TL) (SL_{native} - TL) and
- **comparable monolingual corpora** of the target languages, where the sentences come from native TL speakers (TL_{native}).
- A complete **pipeline** to compile such a corpus from European Parliament debates.

Supported languages (so far) : EN, DE, ES

Corpus processing

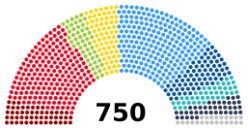
1. Download proceedings in HTML

xml

4. Model proceedings as XML

csv

3. Extract MEPs' information



2. Download MEPs' metadata in HTML

5. Filter out text not in the expected language

xml_metadata

6. Add MEPs' metadata to proceedings

7. Add sentence boundaries

8. Annotate token, lemma, PoS

EN_o

EN_t

EN_n

9. Separate originals from translations and filter by native speakers

10. Extract text into raw format

EN_n

DE

11. Sentence align

Corpus processing

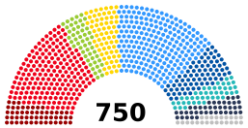
1. Download proceedings in HTML

xml

4. Model proceedings as XML

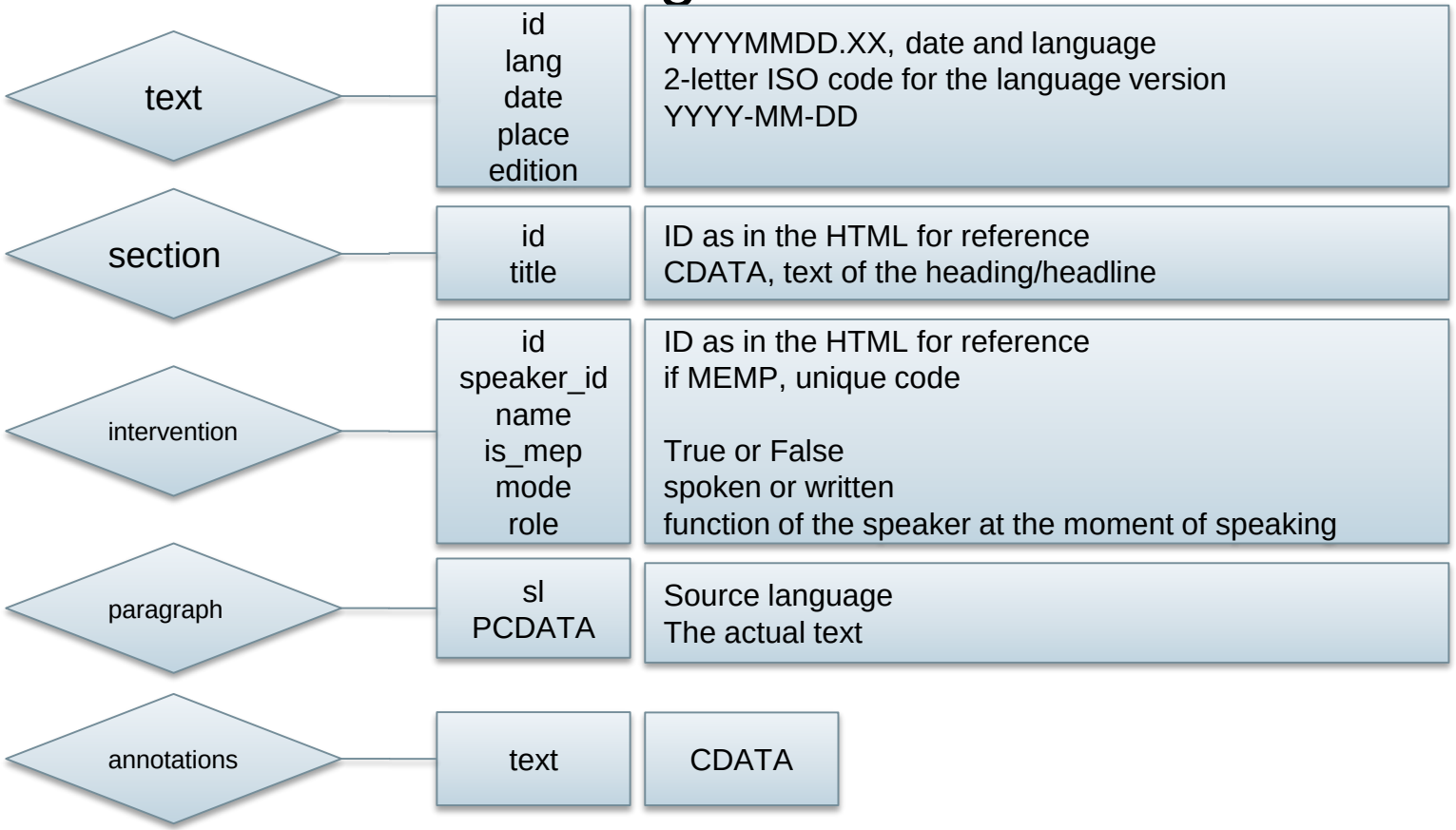
csv

3. Extract MEPs' information

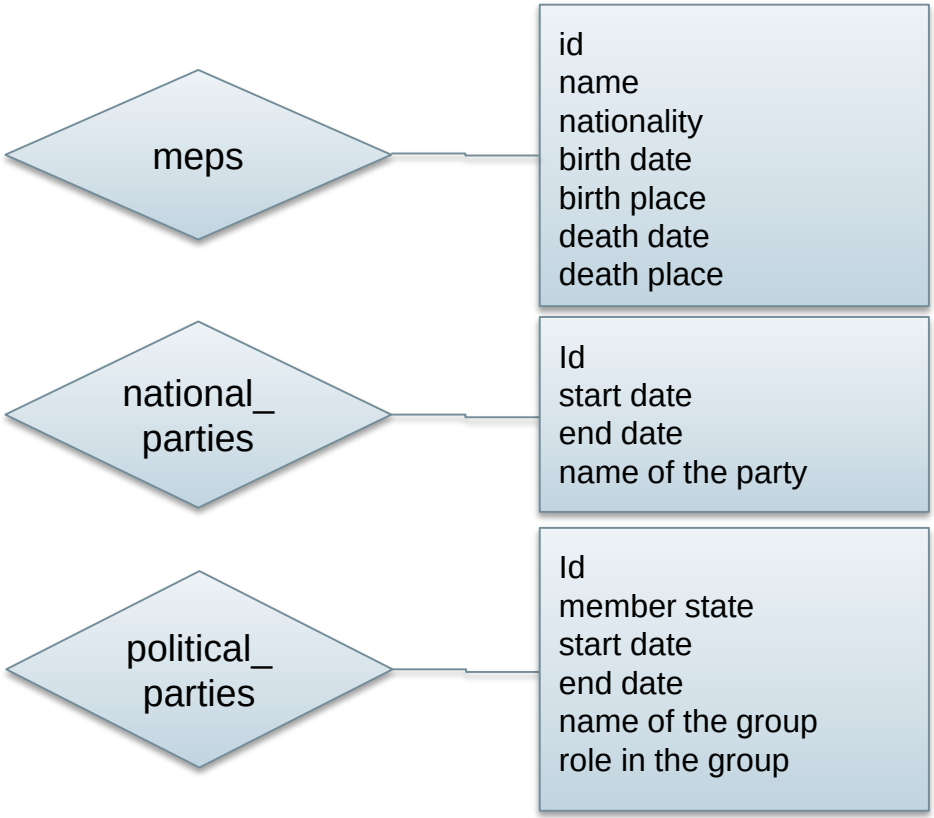


2. Download MEPs' metadata in HTML

Metadata - Proceedings



Metadata - MEPs



Corpus processing

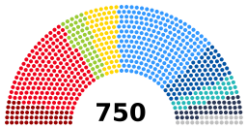
1. Download proceedings in HTML

xml

4. Model proceedings as XML

csv

3. Extract MEPs' information



2. Download MEPs' metadata in HTML

5. Filter out text not in the expected language

xml_metadata

6. Add MEPs' metadata to proceedings

7. Add sentence boundaries

8. Annotate token, lemma, PoS

Corpus processing

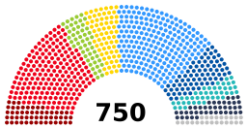
1. Download proceedings in HTML

xml

4. Model proceedings as XML

csv

3. Extract MEPs' information



2. Download MEPs' metadata in HTML

5. Filter out text not in the expected language

xml_metadata

6. Add MEPs' metadata to proceedings

7. Add sentence boundaries

8. Annotate token, lemma, PoS

EN_o

EN_t

EN_n

9. Separate originals from translations and filter by native speakers

10. Extract text into raw format

EN_n

DE

11. Sentence align

Sorting and alignment

- Language identifiers to filter text not in the expected language (Python: *langid*, *langdetect*)
- PoS-Tagging, lemmatisation (*TreeTagger*)
- Sentence split (*Punkt*, NLTK)
- Filter by:
  - Originals vs. Translations
  - Native vs. Non-native

For the parallel corpus:

- Automatically align text per intervention (*hunalign*)

Corpus structure

Directory	Description
html	The crawled proceedings and MEPs' information in HTML
metadata	MEPs' metadata in CSV
txt	Raw text of the proceedings
xml	Proceedings transformed from HTML to XML
xml_langid	Proceedings in XML where the text not in the expected language is filtered out
xml_metadata	Proceedings in XML with added MEPs' metadata
xml_sentences	Proceedings in XML where text is split into sentences
xml_translationese	Proceedings in XML filtered by factors relevant for translation – original, translation, native speaker For each language *a*, it contains · the originals in *a*, · the originals in *a* only by native speakers, · all translations from any language into *a* and · all translations into *a* from a specific *SL* where the speakers are native speakers of the *SL*
xml_ttg	PoS-tagged and lemmatized proceedings in XML
raw_parallel	For each language the corresponding parallel corpora
raw_comparable	For each language the comparable corpus of original texts by native speakers

Corpus statistics

	EN→DE		EN→ES	
	words	sents	words	sents
all	42.08 M/38.93 M	1.91 M	42.11 M/44.21 M	1.87 M
translationese_orig	6.43 M/6.22 M	296.7 K	5.75 M/6.18 M	249 K
translationese_native	3.18 M/3.10 M	137 K	2.93 M/3.15 M	125 K

Table 3: Statistics of the parallel corpora after every processing step

	EN		DE		ES	
	words	sents	words	sents	words	sents
html	95.21 M	5.11 M	91.48 M	5.25 M	97.08 M	5.19 M
xml	95.60 M	5.11 M	92.43 M	5.27 M	97.33 M	5.17 M
langidfilter	65.55 M	3.23 M	40.23 M	2.63 M	51.32 M	2.49 M
translationese_orig	19.69 M	0.84 M	11.74 M	0.68 M	10.75 M	0.37 M
translationese_native	8.67 M	0.37 M	7.86 M	0.42 M	5.66 M	0.18 M

Table 2: Statistics of the comparable corpora after every processing step

Possible applications

- Careful data selection for improving MT (Kurokawa et al., 2009; Lembersky et al., 2012a)
- Translationese features

Text (orig) vs. Text (trans)

trans vs. Rest

- Modelling human translation choice
  (Teich, E. and Martınez Martınez, J., in press)

$$\arg\max_t p(t|s) = \arg\max_t p(s|t)p(t)$$

Adequacy = Fidelity to SL + Conformity with TL

EuroParl-UdS

The corpus: http://fedora.clarin-d.uni-saarland.de/europarl-uds/

The code: https://github.com/hut-b7/europarl-uds

Thank you!

Questions: alina.karakanta@uni-saarland.de

Special thanks to our colleague Jose Martinez Martinez for by providing us his scripts (https://github.com/chozelinek/europarl) for crawling the data - while building this resource.