And Who Is To Judge: On Automatic Judgment Analysis

Vera Davydova[†], Anastasia Kravtsova[†], Maria Podryadchikova[†], Arina Reshetnikova[†]

[†] National Research University Higher School of Economics, Moscow, Russia

Introduction

We present an approach to preprocess Russian court decision texts before the following procedure of discovering plagiarism.
The main difficulty is the lack of annotated data.
Besides, documents are extremely noisy.

Related work

There is a number of studies on legal texts ([7]), but none of them deals with Russian.Researchers carry out a variety of tasks: information retrieval [1], text summarization [3], similarity detection [6].

Data

About 1 million criminal cases of 2355 Russian district courts from 2002 to 2013.
From them **650** cases were already labeled.

1. Text preprocessing

To handle XML format Beautiful Soup package was used.
We used simple rules to divide each text into 3 parts:

- case participants
- case description
- case outcome

2. Metadata extraction

We extracted the following data about the case:

- the court name
- case result
- related criminal code articles
- participants' names (judge, prosecutor, advocate, secretary, accused).

Some of these entities was wrapped in tags and can be extracted with XML parsing. Another types of the entities (names of prosecutor, advocate, secretary and accused) were extracted with named entity recognition instruments.
Prosecutor's, advocate's and secretary's name shared the same strict format, so regular expressions showed solid results.

Evaluation (using metrics from [2]):

Rules and regexps		
entity	strict F1	soft f1
advocate	0.92	0.93
prosecutor	0.91	0.92
secretary	0.98	0.98

accused extraction		
method	strict F1	soft f1
rules + regexps	0.76	0.80
rules + regexps + natasha	**0.81**	**0.85**
rules + regexps + ner_rus	0.57	0.71

3. Sentence segmentation

Court decision texts contain are distinguished by a large number of abbreviations, => we could not resort to the rule-based solutions.
We used unsupervised multilingual sentence boundary detection from [5].
It uses collocational information for the detection of abbreviations, initials, and ordinal numbers.
Train set containing **9280** texts from different courts was selected randomly.
Then, some abbreviations of names and patronymic names that the model did not take into account were added to the list of exceptions.

Your suggestions for improvement

4. Parts detection

Legal judgment components are as follows:

- factual circumstances of a case (fabula)
- witness testimony (witness)
- physical evidences description and expert evidence (proof)
- judge's reasoning (meditation)

Parts detection was solved as a sentence multiclass classification task.
As a classifier, logistic regression was used.

Evaluation

F1-measure for sentence classification		
part	td–idf	sent_emb (avg(word_embs(fasttext)))
fabula	**0.56**	0.22
witness	**0.75**	0.65
proof	**0.46**	0.12
meditation	**0.76**	0.51
overall	**0.68**	0.52

5. Juridisms extraction

We get rid of highly frequent language cliches (juridisms), which create noise and should not be viewed as plagiarism.
To compose the list of juridisms, we:

- take the court with maximum number of documents (6000)
- lemmatize each text with pymorphy2
- extract 5000 most frequent ngrams from 4 to 20 words long using NLTK

Then, we take 20 random courts, repeat the procedure, and find an intersection between each of the 20 lists and the first one.
The whole algorithm was applied to top-5 biggest courts.
The final list consists of **500** phrases.
We lemmatize each text and search for juridisms using Boyer - Moore - Horspool algorithm [4]. Each juridism was replaced by tag JUR.

Plans for the future

1. Enriching data with actual legal judgments from sudact.ru
2. Creation of a legal judgments database
3. Making Python library and web application for legal texts processing

References

- de Araujo D. et al. Exploring the inference role in automatic information extraction from texts //Proceedings of the Joint Workshop on NLPLOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction. 2013. pp. 33-40.
- Chinchor N. MUC-4 evaluation metrics //Proceedings of the 4th conference on Message understanding. Association for Computational Linguistics, 1992. pp. 22-29.
- Farzindar A., Lapalme G. Letsum. An automatic legal text summarizing system // Legal knowledge and information systems, JURIX. 2004. pp. 11-18.
- Horspool R. N. Practical fast searching in strings // Software: Practice and Experience. 1980. T.10. 6. pp. 501-506.
- Kiss T., Strunk J. Unsupervised multilingual sentence boundary detection //Computational Linguistics. – 2006. – . 32. – . 4. – . 485-525.
- Kumar S. et al. Finding similar legal judgements under common law system // International Workshop on Databases in Networked Information Systems. Springer, Berlin, Heidelberg, 2013. pp. 103-116.
- Robaldo L. et al. D2.1 Collection of state-of-the-art NLP tools for processing of legal texts. - 2017.

Artificial Intelligence and Natural Language Conference (AINL), Tartu, Estonia

21 November 2019