Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6988–6994
Marseille, 11–16 May 2020

Natural Language Processing Pipeline to Annotate Bulgarian Legislative Data

Svetla Koeva, Nikola Obreshkov, Martin Yalamov
Institute for Bulgarian Language "Prof. Lyubomir Andreychin"
Bulgarian Academy of Sciences
52 Shipchenski prohod Blvd., Bldg. 17, Sofia 1113
{svetla, nikola, martin}@dcl.bas.bg

Abstract

The paper presents the Bulgarian MARCELL corpus, part of a recently developed multilingual corpus representing the national legislation in seven European countries and the NLP pipeline that turns the web crawled data into structured, linguistically annotated dataset. The Bulgarian data is web crawled, extracted from the original HTML format, filtered by document type, tokenised, sentence split, tagged and lemmatised with a fine-grained version of the Bulgarian Language Processing Chain, dependency parsed with NLP-Cube, annotated with named entities (persons, locations, organisations and others), noun phrases, IATE terms and EuroVoc descriptors. An orchestrator process has been developed to control the NLP pipeline performing an end-to-end data processing and annotation starting from the documents identification and ending in the generation of statistical reports. The Bulgarian MARCELL corpus consists of 25,283 documents (at the beginning of November 2019), which are classified into eleven types.

Keywords: NLP pipeline, legislative corpus, Bulgarian language

1. Introduction

The paper presents the Bulgarian MARCELL corpus, part of a recently developed multilingual corpus representing the national legislation in seven European countries and the NLP pipeline that turns the web crawled data into structured, linguistically annotated dataset. The presented work is an outcome of the CEF Telecom project Multilingual Resources for CEF.AT in the Legal Domain[1] (MARCELL) aiming to enhance the eTranslation system of the European Commission by supplying large-scale domain specific data in seven languages (Bulgarian, Croatian, Hungarian, Polish, Romanian, Slovak and Slovenian).

The NLP Pipeline for Bulgarian, developed specifically to answer the requirements of MARCEL project for autonomy and sustainability, is continuously feeding the Bulgarian MARCELL corpus with newly issued legislative documents. Data is extracted from a single web source and further transformed. The transformation phase makes changes to the data format, organises data in structures, accumulates data with linguistic information, analyses data and provides explicit links between different data segments. In particular, the Bulgarian data is web crawled, extracted from the original HTML format, filtered by document type, tokenised, sentence split, tagged and lemmatised with a fine-grained version of the Bulgarian Language Processing Chain[2] and dependency parsed with NLP-Cube[3]. Further, named entities (persons, locations, organisations and other), noun phrases, IATE terms[4] and EuroVoc descriptors[5] are recognised and annotated within the corpus.

The paper is focused on the NLP pipeline for Bulgarian to reveal some of its features:

- Standardisation: CoNLL-U Plus formatted output at any level of annotation.
- Domain-specific adaptation: the NLP pipeline modules are tuned to annotate legal domain documents.
- Multiword coverage: Multiwords are handled at the levels of annotation.
- Self-driving solutions: All modules are synchronised to work together and assembled in a self-driving architecture.

The paper is organised as follows: in Section 2, we present the data collection: crawling, extraction of raw text and metadata, grouping of topic-related documents. Section 3 presents the structure, format and current stage of the Bulgarian MARCELL corpus. The language processing modules of the NLP Pipeline for Bulgarian are presented in Section 4. Section 5 outlines how the pipeline is assembled. Finally, Section 6 presents conclusions for our results and explains how the pipeline will be further enriched.

2. Compilation of the Corpus

The primary data source for the compilation of the Bulgarian MARCELL corpus is the Bulgarian State Gazette[6], the Bulgarian government's official journal, publishing documents from the official institutions, such as the government, the National Assembly of Bulgaria, the Constitutional Court, etc. The online edition of the Bulgarian State Gazette maintains 113,427 legal documents in HTML (in November 2019) which can be accessed through the Register page. The documents are described in a searchable table amounting in 8,330 pages. The documents navigation is performed by selecting the page number from a dropdown menu, which calls a JavaScript function. The JavaScript code is injected into the HTML response and is processed at the client browser. The document metadata is stored in the Register tables, which requires content scraping and linking with the URLs of the documents. Some documents are registered but still not published and this results in scanning the Register for new available links on a regular basis.

In general, crawlers are suitable for large-volume low-complexity content extraction, while scrappers are best for

[1] https://marcell-project.eu
[2] https://dcl.bas.bg/en/webservices/
[3] https://opensource.adobe.com/NLP-Cube/index.html
[4] https://iate.europa.eu/home
[5] https://marcell-project.eu/links.html
[6] https://dv.parliament.bg

smaller amounts of data with high complexity (Shu 2020: 210). Some of the most commonly used crawlers are Heritrix[7], Nutch[8], Crub Next Generation[9]. There are also a number of widely available scrappers, such as Scrapy[10], Octoparse[11], Apifier[12], etc. The intended source data and metadata require a non-standard crawling algorithm, which will be able to cope with a specific navigational browsing and to integrate crawling and metadata extraction techniques. It is implemented in two separate tools: for universal web crawling and for dedicated scraping.

2.1 Crawling

A Universal web crawler has been developed to collect data from websites and Single-Page Applications that use AJAX technology, JavaScript and cookies. It is based on web automation testing technologies that can extract post rendering content. Puppeteer[13] v2.0.0 is used as a core framework – a Node library that provides a high-level API to control Chromium or Chrome over the DevTools Protocol. The main features in use are:

- Running a Google Chrome web browser in a headless mode.
- Performing a sequence of actions on web applications or web pages.
- Saving the rendered content for the extraction of hyperlinks for recursive crawling and further data processing.

2.2 Scraping

A Dedicated scrapper has been developed to scan, on a regular basis, the Register of the Online Bulgarian State Gazette for new entries and to extract the metadata: document type, registered date, title, status, URL, etc. The collected URLs are added to the seeds list of the Universal web crawler, which is set to work in a non-recursive mode. The LibreOffice Version: 6.3.2.2[14] is used in a headless mode to convert any HTML page without images to a raw text file

2.3 Data storage

A non-relational database server MongoDB[15] is used as sharing data point and for storing the following data:

- The crawler's seeds (the list of URLs for data scraping).
- The crawl frontier (the list of pending URLs that are extracted from web pages and recursively visited according to a set of policies).
- The visited URLs, for tracking the visits and looping preventions.
- The extracted metadata (title, author, publishing date and file content type).
- Other metadata (file storage location, file size and file type).

[7] https://github.com/internetarchive/heritrix3/wiki
[8] https://nutch.apache.org
[9] https://www.openhub.net/p/grubng
[10] https://scrapy.org
[11] https://www.octoparse.com
[12] https://apify.com/apify/web-scraper
[13] https://pptr.dev
[14] https://www.libreoffice.org/
[15] https://www.mongodb.com/

2.4 Metadata

Each document is linked to the respective metadata. The metadata is extracted from the page URL (1.), the document title (2.), the register tables (3., 5., 7. – 14.) or the document body (3., 6.). Some metadata is calculated or translated. The following metadata is obligatory for the MARCELL corpus.

1. ID: the unique identifier of the document.
2. DATE: the date (in ISO 8601 format) of creation of the legal act, or if not available, the date of going into effect of the legal act. If both dates are available, both are kept.
3. TYPE: the legislative type in Bulgarian to which a legal act is classified.
4. ENTYPE: the English translation of the legislative type.
5. TITLE: the name of the legal act.

Some other metadata is optional but recommended for the MARCELL corpus:

6. ISSUER: the organisation that issued the legal act.
7. URL: the original web address of the legal act.

At the end, there is metadata, which is used only internally:

8. DOCID: the unique identifier of the document within the Bulgarian State Gazette.
9. PLACE: the place where the legal act was created.
10. NUMBER: the outgoing document number.
11. STATUS: whether the document is already published or not.
12. LOCATION: the file storage location.
13. SIZE: the file size.
14. FILETYPE: the type of the file.
15. GROUP: the unique identifier of the primary document to which a document is related (i.e., a law and an instruction to this law). The relation between the documents is established if the specifying parts of their titles match (namely, the lemmas of the content words with the application of some exclusion rules and exclusion lemma lists).

3. Composition of the Corpus

3.1 Corpus structure

The Bulgarian MARCELL corpus consists of 25,283 documents (at the beginning of November 2019), which are classified into eleven types: Administrative court, Agreements, Amendments, Legislative acts, Conventions, Decrees, Decrees of the Council of Ministers, Guidelines, Instructions, Laws (Acts), Memorandums and Resolutions. The Bulgarian MARCELL corpus is a selection from a larger legislative domain dataset (113,427 documents containing 5,748,391 sentences, 81,651,084 words distributed in 52 legal types, with a total size of 1977.88M). The distribution of the number of documents, the number of sentences and the number of words, as well as the size of the documents in the selected legislative types is presented in Table 1.

Type	Docs	Sentences	Words	Size
Adm. court	5859	29066	686797	15.3866M
Agreements	593	96707	1731340	31.9039M
Amendments	253	2553	34934	0.80342M
Conventions	468	103009	1864710	31.6136M
Decrees	5318	1332830	17686526	377.193M
DecreeCM	5998	879524	11370176	328.213M
Guidelines	916	171807	2383244	41.1557M
Instructions	516	110646	1429302	33.3544M
Laws (Acts)	2551	467906	7516854	190.561M
Memorandums	209	25548	413923	8.26869M
Resolutions	2602	62070	826825	22.0285M
Total	25283	3281666	45944631	1080.48M

Table 1: The Bulgarian MARCELL corpus in numbers

The corpus is designed such as to represent universally binding legal acts. The selected documents are with different length: 2,931 documents contain less than 100 words, 12,013 – between 100 and 499 words, 2,390 documents – between 500 and 999 words and 7,949 documents – more than 1000 words. The time span of the documents in the Bulgarian MARCELL corpus is 1946 – 2019. The main part of the documents has been retrieved from the Bulgarian State Gazette. Only 33 documents were manually added to the corpus – legal acts that were issued before 2000 but are still in effect. The documents are distributed as follows: 9,928 documents issued before 2010 and 15,322 documents issued after 2010. The issuers are the Constitutional Court (6 documents), the Council of Ministers (6,120 documents), courts (5,844 documents), ministries and other institutions (8,127 documents), the National Assembly (5,167 documents), others (4 documents), the President of the Republic of Bulgaria (4 documents), state institutions and agencies (11 documents).

3.2 Corpus format

The Bulgarian MARCELL corpus is structured in the CoNLL-U Plus Format[16] enhanced with 4 additional columns added to answer the specific requirements of the MARCELL project (annotation of named entities, noun phrases, IATE terms and EuroVoc descriptors). The data is encoded in UTF-8 plain text files with three types of lines: word lines with the annotation in 14 fields separated by single tab characters, blank lines marking sentence boundaries and comment lines marked with a hash tag. Underscore (_) is used to denote unspecified values in all fields. The fields for text and annotation are as follows:

1. ID: a word index, starting at 1 for each new sentence.
2. FORM: a token, which might be a word form, an abbreviation, a date, a numerical expression, a punctuation mark or other symbols.
3. LEMMA: a lemma of a word form or a repetition of symbols in the filed FORM.
4. UPOS: a Universal part-of-speech tag[17]. The language specific part-of-speech tags are converted to the universal part-of-speech tags, i.e. NCF (noun, common, feminine) → NOUN.
5. XPOS: a language-specific part-of-speech tag. The language specific part-of-speech tags represent more extensive linguistic information compared to the Universal part-of-speech tags.
6. FEATS: a list of morphological features from the Universal feature inventory.
7. HEAD: the head of the current word, which is either the index of the HEAD word or zero.
8. DEPREL: the Universal dependency relation to the HEAD.
9. DEPS: an enhanced dependency graph (optional).
10. MISC: any other annotation.
11. MARCELL:NE: a BIO format annotation of the current token if it is part of a Named entity; each entity label (PER – for persons, LOC – for locations, ORG – for organisations, MISC – for any other named entity) is prefixed with either B or I, B for the beginning and I for the inside of an entity. Named entities are numbered within a given sentence (starting from 1).
12. MARCELL:NP: a BIO format annotation of the current token if it is part of a noun phrase. Noun phrases are numbered within a given sentence (starting from 1).
13. MARCELL:IATE: a MARCELL format annotation of the current token if it is (part of) an IATE term. IATE terms are numbered within a given sentence (starting from 1) and the number is repeated for each token belonging to the term. The IATE identification number for the term is listed followed by the numbers of the corresponding EuroVoc descriptor(s).
14. MARCELL:EUROVOC: a MARCELL format annotation of the current token if it is (part of) a EuroVoc descriptor. EuroVoc descriptors are numbered within a given sentence (starting from 1) and the number is repeated for each token belonging to the descriptor.

Each document in the Bulgarian MARCELL corpus has a unique name, constituted by the language code and identifier (i.e. bg-100317). The paragraphs and sentences are numbered (starting from 1) and distinguished by unique identifiers in comment lines. The text of the sentences is also presented in comment lines before the annotation lines as shown in Example 1.

newpar id = bg-100317-p1 (1st paragraph of the document bg-100317)

sent_id = bg-100317-p1s1 (1st sentence from the 1st paragraph of the document bg-100317)

text = Закон за Комисията по финансов надзор (the text of the first sentence: Zakon za Komisiyata po

[16] https://universaldependencies.org/format.html

[17] https://universaldependencies.org/u/pos/index.html

finansov nadzor 'Law on the Financial Supervision Commission')

Example 1: Comment lines for paragraph and sentence boundaries, and raw text of a sentence

4. NLP Pipeline for Bulgarian

There are several NLP libraries providing sets of linguistic annotations (tokenisation, sentence splitting, paragraph boundary detection, spell checking, lemmatisation, named entity recognition, chunking, dependency parsing, sentiment analysis, etc.). Some of the libraries are Java-based such as OpenNLP[18] and Stanford NLP (Manning et al., 2014)[19], others are Python-based such as NLTK (Bird and Loper, 2004) and spaCy[20], and some support multiple programming languages, i.e. Spark NLP[21]. A number of libraries provide deep learning techniques and knowledge graphs, and report good level of accuracy and speed (i.e. Spark NLP); however, they are not accomplished with pre-trained models for Bulgarian.

We implemented a C++ based pipeline for processing Bulgarian texts which integrates a sentence splitter, a tokeniser, a part-of-speech tagger, a lemmatiser, a UD parser, a named entity recogniser, a noun phrase parser, an IATE term annotator and a EuroVoc descriptor annotator. The sentence splitter, the tokeniser, the part-of-speech tagger and the lemmatiser are organised in a chain: Bulgarian Language Processing Chain – BGLPC (Koeva and Genov, 2011). The different modules: BGLPC, a named entity recognition, a noun phrase detection, an IATE term annotation and a EuroVoc descriptor annotation are implemented as stand-alone tools, which makes it easy to incorporate an existing or new tool in the pipeline. The syntactic dependencies are parsed with NLP-Cube (Boroș et al., 2018). The modules are represented in Figure 1.

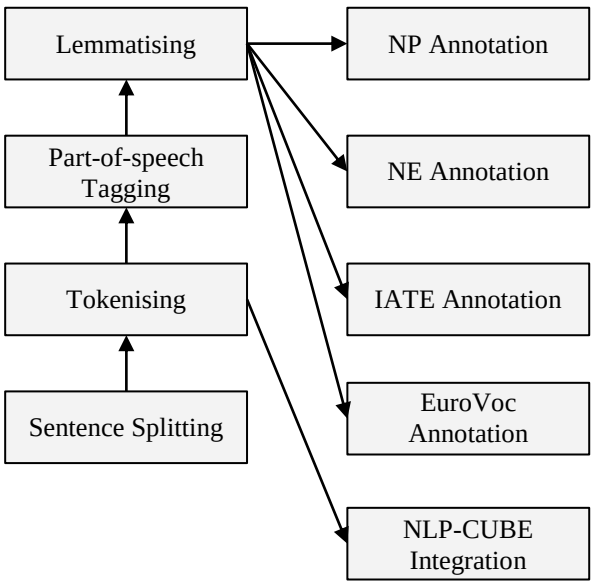

Figure 1. Modules of NLP Pipeline for Bulgarian

[18] https://opennlp.apache.org
[19] https://nlp.stanford.edu/software/
[20] https://spacy.io
[21] https://nlp.johnsnowlabs.com

Recently, two NLP pipelines (including a tokeniser, a sentence splitter, a tagger, a lemmatiser and a dependency parser) have become available: IDpipe and NLP-Cube (Straka et al., 2016; Boroș et al., 2018). Both pipelines are trained for languages with the UD Treebanks, including for Bulgarian (Nivre et al., 2018). The pipelines can be trained with external data in CoNLL-U format, but no major improvements could be expected without a larger UD treebank.

The presented NLP pipeline for Bulgarian uses the BGLPC for all levels of basic annotation because the BGLPC accuracy is higher than other methods. The BGLPC is successfully implemented in business solutions for real time data analyses[22] and corpus query systems[23].

4.1 Basic Annotation

The Bulgarian language processing chain created in 2011 consisted of a BG sentence splitter, a BG tokeniser, a BG Part-of-speech (POS) tagger, a BG lemmatiser, a BG Noun phrase (NP) extractor, a BG Named entity recogniser and a BG Stop word recogniser. All tools are self-contained and part of them are designed to work in a chain, i.e. the output of the previous component is the input for the next component, starting from the sentence splitter and following the strict order for the tokeniser, the POS tagger and the lemmatiser (Karagiozov et al., 2011). The other modules in the NLP pipeline for Bulgarian (except the UD parser) use the lemmatiser output and there are no dependencies in their execution order.

The enhanced versions of the BG sentence splitter, the BG tokeniser, the BG Part-of-speech tagger and the BG lemmatiser are used.

The BGLPC uses regular expressions for sentence and token segmentation supplemented with lists for different types of abbreviations to prevent splitting the abbreviations and punctuation marks at the level of tokenisation and splitting the sentences at false borders. Additional rules for sentence splitting have been developed to address cases specific to the legislative texts, i.e. single sentences divided into multiple raws; enumeration lists with atypical formatting, etc. At the tokenisation level, some additional rules are also implemented to recognise complex numbering, numbering of articles and paragraphs of laws, etc. as a single token.

Most of the applications for the identification and classification of Multiword expressions (MWEs) involve POS tagging and lemmatisation as a preliminary step. For example, the multilingual identification of MWEs is based on multiple sources of linguistic information, including POS tagging and lemmatisation (Tsvetkov and Wintner, 2011); the method for the identification of nested terminology uses a pre-processing for POS tagging and lemmatisation (Newman et al., 2012); the MWEs automatic classification with a SVM (Support-vector Machine) classifier is preceded by POS tagging (Kumar et al., 2017). The need for an integrated representation that identifies and labels single words and multiword noun and verb expressions has already been recognised (Schneider and Smith, 2015; Schneider et al., 2016). However, there are only a few attempts for simultaneous text

[22] https://www.tetracom.com/ilib/tetracom/products/graze
[23] https://www.sketchengine.eu/bulgarian-tagset/

segmentation into single words and MWEs at the level of tokenisation and for simultaneous grammatical and lexical annotation of single words and MWEs at the level of POS tagging and lemmatisation. A recent approach uses a POS tagger and a finite state transducer to extract and add multiword noun phrases to a unigram bag of words text representation (Handler et al., 2016) but it does not handle the lemmatisation of MWEs.

The Bulgarian Language Processing Chain provides: a) simultaneous segmentation of texts in single words and MWEs and b) simultaneous POS tagging and lemmatisation of single words and MWEs. To achieve these we: a) mixed three manually annotated corpora and enlarged them with a uniform annotation for contiguous MWEs (Koeva et al., 2006; Koeva et al., 2012) and b) enlarged the grammatical dictionary with forms, lemmas and grammatical categories of abbreviations and contiguous MWEs.

The list of contiguous MWEs (incorporated in the grammatical dictionary) is also implemented at the level of tokenisation to ensure merged segmentation of complex lexical units. The POS tagger is re-trained with the uniformly mixed re-annotated corpus for single words, abbreviations and MWEs. The accuracy of the POS tagger is not improved significantly, only by 0,033 points, compared to the accuracy before re-training. However, the most important result is that single words and MWEs are segmented and POS annotated simultaneously, which is also reflected in the enhancement of the existing lemmatiser.

A highly scalable web service based infrastructure was developed to provide easy access to the Bulgarian Language Processing Chain. There are different types of access[24]: access via browser[25] – suitable for users who need processing of relatively small amount of data occasionally; access via RESTful API – suitable for software developers who can integrate the processing tools in high level applications; asynchronous access – suitable for time-consuming tasks such as processing large corpora.

4.2 Annotation with UDs

Universal dependency parsing is performed with the NLP-Cube Framework used in API mode. A Python script was created to provide access to the NLP-Cube functionality and to automate the processing of the Bulgarian MARCELL corpus. For every document within the corpus, the NLP-Cube annotation and the BGLPC annotation are token-wise synchronised and a correspondence map is created between identical tokens in both documents. Based on this synchronisation, the Universal dependency relations are transferred to the BGLPC CoNLL-U Plus output and the relation index is recalculated.

4.3 Annotation with Noun Phrases

Manually crafted rules are used to identify noun phrases based on the syntactic information and the surrounding context. The rules are applied in a predefined order ensuring that a potential noun phrase will be captured as a whole instead of matching its subparts. The grammar is designed to recognise syntactically unambiguous phrases and to exclude pronouns and relative clauses as modifiers. The grammar operates on the part-of-speech and morphosyntactic tags and provides annotations for noun phrase boundaries and noun phrase heads. The processing of the Bulgarian MARCELL corpus has resulted in the recognition and annotation of 3,713,209 noun phrases. The output format is described in Subsection 3.2.

4.4 Annotator Tool

For IATE term and EuroVoc descriptor annotation, a dedicated instrument called TextAnnotator was developed and further used for named entity annotation. The TextAnnotator calls dictionaries of terms and finds occurrences of these terms in the documents. Both the documents and the dictionaries have to be in the CoNLL-U format. Each dictionary consists of a collection of single and multiword entries starting with an entry identifier and ending with a delimiter symbol # as shown in Example 2.

```
IATE-128861:1236,2841,4016      M       _       _
"       U       "       U
Лекари  N       лекар   NCMpo
без     R       без     R
граници N       граница NCFpo
"       U       "       U
#       M       #       M
```

Example 2: Dictionary entry for the multiword IATE term Lekari bez granitsa ‘Doctors without borders’ in Bulgarian

The annotation tool matches sequences of lemmas and part-of-speech tags of dictionary entries and lemmas and part-of-speech tags of document tokens. The matching procedure is based on a hash table indexing. For each dictionary entry, a hash key is generated concatenating lemmas and part-of-speech tags within it, as shown in Example 3:

```
"UлекарNбезRграницаN"U
```

Example 3: Concatenated information for a dictionary entry ‘Doctors without borders’ in Bulgarian

All hash keys for a given dictionary are grouped into length classes based on the number of words they contain. Each length class is implemented as a hash table. A single document is processed as follows (Example 4):

```
FOR EACH n IN documentLength
  FOR EACH lengthClass in MAX_SORT(lengthClases)
    FOR EACH length IN lengthClass
      documentKey =
        CreateDocumentKey(document, n, length)
      IF Exists(lengthClass, documentKey)
        Annotate(document, n, length)
        n += length - 1
```

Example 4: The matching algorithm

The algorithm gives a priority to the longest length classes, which ensures the selection of longest matches. When a match is found, the corresponding tokens in the document are annotated and the processing continues from the end index of the match.

For the improvement of the accuracy, several normalisation rules are applied priory to the annotation. For example, some guillemet symbols are replaced with ASCII symbols. The normalisation led to an improvement of up to 3% for the annotation of IATE terms and

[24] https://dcl.bas.bg/en/webservices/

[25] http://dcl.bas.bg/dclservices/

EuroVoc descriptors. This approach is appropriate for domains (such as the legislative domain) with unique or specialised terminology because we do not analyse the context and disambiguate the word senses.

4.5 Annotation with Named Entities

There are several attempts towards Named entity recognition for Bulgarian. Georgiev et al. (2009) reported a model based on Conditional Random Fields enhanced with hand-crafted features. Recently, Simeonova et al. (2019) proposed a model based on LSTM-CRF architecture and combined with word embeddings, Bi-LSTM character embeddings, part-of-speech tags and morphological information. Earlier models relied on manually crafted rules (Koeva and Genov, 2011) and lexicons (Koeva and Dimitrova, 2014). Taking into account the specifics of the legal domain (legislative texts are highly structured and formalised), we decided to use predefined lists with non-ambiguous named entities. Four lexicons with names (18,868), locations (461), organisations (2,095) and miscellaneous (296) Named entities were manually created (Koeva and Dimitrova, 2015).

The annotation is performed with the TextAnnotation tool and the output format is described in Subsection 3.2. Lexicon-based classification of Named entities was enhanced with context-sensitive rules of the form: <trigger> <candidate name>, which help to correctly disambiguate and annotate Named entities that belong to more than one category. For example, the rule combining the trigger улица (ulitsa) 'street' with a candidate name resulted in 2,188 correct annotations for a location name.

4.6 Annotation with IATE terms and EuroVoc descriptors

The Bulgarian MARCELL corpus has been annotated with terminology from two terminology repositories: IATE – 'Interactive Terminology for Europe', the EU's terminology database used in the EU institutions and agencies, and EuroVoc, a multilingual, multidisciplinary thesaurus covering the activities of the EU. The terminology annotation is performed with the TextAnnotator and the output format is described in Subsection 3.2.

The identification numbers (IDs) of the IATE terms point also to the relevant EuroVoc domains and sub-domains. There are 45,592 IATE terms for Bulgarian. The annotation takes into account that several terms can be related with one and the same IATE ID (synonyms) and one term can be related with different IDs (polysemy). There are also IATE terms in Bulgarian which describe concepts specific for other languages, i.e. община (obshtina) IATE ID: 3553038 'regions of Poland'. Such terms were excluded from the annotation (4,641 terms altogether).

5. Continuous updating of Bulgarian MARCELL corpus

An orchestrator process has been developed to control the NLP pipeline performing an end-to-end data processing and annotation starting from the documents identification and ending in the generation of statistical reports. It organises and ensures the permanent work of crawler and scraper for the new document acquisition and the raw text and metadata extraction; of tools for grouping of topic-related and type-related documents; of modules for linguistic annotation: sentence splitting, tokenisation, tagging, lemmatising, synchronisation with the NLP-Cube dependency parsing, named entity recognition, identification of noun phrases, annotation with IATE terms and EuroVoc descriptors; of the tool combining metadata and annotation output data in a common CoNLL-U Plus export, and of the tool for generation of statistical reports.

The downloaded files are stored in their original HTML format. Extracted texts are stored in TXT files for later annotation. Annotated files are stored in TSV files. A noSQL database MongoDB is used for synchronisation, status tracking, task queuing, metadata storing and generation of statistics.

6. Conclusions

The Bulgarian MARCELL corpus comprises national legislative documents in effect. Apart from the standard morphosyntactic analysis coupled with Named entity and dependency annotation, the corpus is enriched with the IATE and EUROVOC terminology set. The final dataset will be sent to the ELRC repository of language resources[26]. The NLP pipeline for Bulgarian is designed to ensure sustainable and autonomous feeding of the corpus with new legislative documents.

The pipeline will be enhanced with a text categorisation module. We have conducted some experiments with the JEX dataset (Steinberger et al., 2013) representing legal documents in 22 European languages annotated with EuroVoc descriptors. We are also experimenting with some calculations over the IATE and EuroVoc annotations.

The modularity of the presented pipeline allows inclusion of other NLP modules, such as syntax-driven clause segmentation, shallow parsing and others. We plan to build neural models for tagging, parsing and entity recognition to achieve a better balance of speed and accuracy.

7. Acknowledgements

The work reported here was supported by the European Commission in the CEF Telecom Programme (Action No: 2017-EU-IA-0136). We wish to thank the following colleagues for their valuable work in the project: Tsvetana Dimitrova, Valentina Stefanova, Dimitar Georgiev, Valeri Kostov, Tinko Tinchev.

8. Bibliographical References

Bird, S. and Loper, E. (2004). NLTK: The Natural Language Toolkit. In *Proceedings of ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, pp. 31–es.

Boroş, T., Dumitrescu, Ş.D., Burtica, R. (2018). NLP-Cube: End-to-End Raw Text Processing With Neural Networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* Association for Computational Linguistics. pp. 171–179.

Georgiev, G., Nakov, P., Ganchev, K., Osenova, P., and Simov, K. (2009). Feature-rich named entity

[26] https://elrc-share.eu/repository/search/

recognition for Bulgarian using conditional random fields. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Borovets, Bulgaria, RANLP '09, pp. 113–117.

Handler, A., Denny, M., Wallach, H. and O'Connor, B. (2016) Bag of What? Simple Noun Phrase Extraction for Text Analysis. In *Proceedings of the Workshop on Natural Language Processing and Computational Social Science*, pp. 114–124.

Karagiozov, D., Koeva, S., Ogrodniczuk, M., and Vertan, C. (2011). ATLAS – A Robust Multilingual Platform for the Web. In *Proceedings of the German Society for Computational Linguistics and Language Technology Conference* (GSCL 2011), Hamburg, pp. 223–226.

Koeva, S. and Genov, A. (2011). Bulgarian Language Processing Chain. In *Proceeding to The Integration of multilingual resources and tools in Web applications Workshop in conjunction with GSCL 2011*, University of Hamburg.

Koeva, S., Dimitrova. T. (2015). Rule-based Person Named Entity Recognition for Bulgarian. Slavic Languages in the Perspective of Formal Grammar. In *Proceedings of FDSL 10.5*, Brno 2014, *Series Linguistic International*, vol. 37, Peter Lang, pp. 121–139.

Kumar, Sh., Beherab, P., Jhac, G. (2017). A classification-based approach to the identification of Multiword Expressions (MWEs) in Magahi Applying SVM. In *Procedia Computer Science*, Volume 112, pp. 594–603.

Manning, Ch., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*: System Demonstrations, ACL Press, pp. 55–60.

Newman, D., Koilada, N., Lau, J., and Baldwin, T. (2012). Bayesian text segmentation for index term identification and keyphrase extraction. In *Proceedings of the 9th Workshop on Multiword Expressions*, pp. 139–144.

Nivre, J., Abrams, M., Agić Ž., and Ahrenberg. (2018). *Universal dependencies 2.3*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Schneider, N. and Smith, N. (2015). A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 1537–1547.

Schneider, N., Hovy, D., Johannsen, A., and Carpuat, M. (2016). SemEval-2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM). In *Proceedings of SemEval-2016*. San Diego, California, pp. 546–559.

Shu, X. (2020) *Knowledge Discovery in the Social Sciences: A Data Mining Approach*. University of California Press.

Simeonova, L., Simov, K., Osenova, P., Nakov, P. (2019) A Morpho-Syntactically Informed LSTM-CRF Model for Named Entity Recognition. In: *RANLP-2019*, 1104–1113.

Steinberger, R., Ebrahim, M., Turchi, M. (2012). JRC EuroVoc Indexer JEX-A freely available multi-label categorisation tool. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation* (LREC'12), pp. 798–805.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation* (LREC 2016). European Language Resources Association, Portoroz, Slovenia. pp. 4290–4297.

Tsvetkov, Y. and Wintner, Sh. (2011). Identification of multi-word expressions by combining multiple linguistic information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. pp. 836–845.

9. Language Resource References

Koeva, S., Leseva, S., Stoyanova, I., Tarpomanova, E., Todorova, M, (2006). Bulgarian Tagged Corpora. In *Proceedings of the Fifth International Conference Formal Approaches to South Slavic and Balkan Languages*, 18-20 October 2006, Sofia, Bulgaria, pp. 78–86.

Koeva, S., Rizov, B., Tarpomanova, E., Dimitrova, Ts., Dekova, R., Stoyanova, I., Leseva, S., Kukova, H., and Genov, A. (2012) Bulgarian-English Sentence- and Clause-Aligned Corpus. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities* (ACRH-2), Lisbon, Lisboa: Colibri, pp. 51–62.