Semantic Annotation Module For Legal Acts Analysis System

Roman Salakhov
Faculty of Economics, Management, and Business Informatics
National Research University Higher School of Economics
Perm, Russia
rdsalakhov@edu.hse.ru

The modern law system became more and more complex, so there appears a need for a system that analyzes legal documents and extracts named entities and links to other documents. This system will help to introduce new laws and amendments without making controversy. The aim of this study is to develop a module for named entity recognition in legal documents. This app should extract entities from a text with basing on a given ontology and link every entity with a particular place in the text. Methods of study are based on common software engineering approaches such as collecting the requirements, developing system design and algorithms. Anticipated results are supposed to be an application that can create an annotation file for the given text. The proposed solution can be used for analytics and indexing of large text databases.

Keywords – NLP, legal reference system, semantic annotation

I. Introduction

A. Background

Today people have to deal with large amounts of textual information to find particular data or analyze and process texts. The most helpful tool that makes such work more convenient is search engines. Many of them are simple for example search tools in a text editor or in-page search in a web browser but there are also highly complicated systems such as web search engines (e.g. Google, Bing, Yandex). The main difference is that simple systems are just searching for particular symbol sequences in a text while complex systems can return results based not only on symbolic similarity but on the semantic meaning of the search request. Those features are based on approaches of NLP (natural text processing) in general and on semantic analysis and annotating in particular.

B. Problem Statement

Although complex search engines for common knowledge are well researched and used across the world, there is still a need for narrowly focused systems that will be used in a particular field of knowledge. Delimitation to a specific domain should result in more relevant search results because in that case polysemy of natural language could be omitted. The goal of the proposed project is to develop an annotation module for a search engine that will be specified on legal documents.

C. Delimitations of the Study

Delimitations of the study include the following: only module for semantic annotation will be developed, the ontology (list of terms of the legal domain) will not be developed thoroughly as a basis for annotation will be used a fictitious set of terms that are presumably related to law without any relations between them. There are also technical delimitations such as the size of the text that the application could process in a reasonable time and the simplicity of the semantic detecting algorithm.

D. Professional Significance

The proposed system will show how successful the semantic-based searching approach will be when applied to a domain-specific set of documents. In addition, enterprises can build more complex searching engines upon the proposed solution and use this project as a basis.

II. Literature Review

The main task of the project is related to computational analysis of texts written in natural, “human” language. This kind of task is usually studied and handled by a field of science called NLP. NLP or Natural Language Processing is a domain which unites sciences like linguistics, computer science and artificial intelligence. [1] introduces the history of that field and gives examples of common tasks and methods. M. Kurdi also describes basic concepts of corpus linguistics and morphology whose applications are also necessary in natural language processing.

The fundamental work that is needed in every NLP based analysis is data preprocessing. [2] reveals general approaches and techniques that help to prepare text for further processing and make work on next stages easier. The first approach is “tokenization” which is splitting text into lesser entities such as words or sentences. The second stage of text preprocessing is normalization. The main goal of this approach is to convert different forms of word to a single common one. There are two most popular options to normalize a text: stemming and lemmatization. Lemmatization removes word ending and converts word to the base dictionary form whereas stemming is a procedure to reduce all words with the same beginning to a most long common part. Comparison of these two methods in [3] shows that lemmatization is more precise, but the difference is insignificant. However, there is another comparison for Slovene language in [4] which is more related to the project because Slovene language is more similar to Russian than English, so the best stemming approach stated in [4] is preferable for the project.

Another text preprocessing procedure is stated in [5] and called noise removal. It includes removal of stop words, punctuation and markup tags. Stop words usually include articles, conjunctions, prepositions and so on. These words have a minor impact on the semantic aspect of text so without them data becomes cleaner and easier to work with. Now when the theory of text preprocessing has been reviewed, we need to find instruments to perform stated approaches.

One of the most popular NLP tools is NLTK [6]. It stands for natural language toolkit and provides various text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, along with a convenient interface to 50 linguistic resources and corpora. Its capabilities cover everything that might be needed for raw text

data preprocessing and basic natural language analysis. The drawbacks of using this tool are the steep learning curve, and insufficient performance and optimization for production usage. However, [7] states that NLTK is a safe choice for academic and educational purposes as well as for prototyping and testing enterprise solutions because of the rich tool spectrum and proven algorithms that lie in the basis of the library.

Another widely used tool is SpaCy [8]. As well as NLTK, SpaCy is a framework for the Python programming language but the difference is that SpaCy uses more advanced approaches which allows the libraries to work faster and demonstrate a better performance. As for other features, SpaCy provides a slightly more limited variety of language tools than NLTK, but it still includes libraries for text preprocessing, tagging, parsing and semantic reasoning. [9] shows how SpaCy could be used for tasks that are pretty like the tasks of our project. The research demonstrates how this framework can assist in part-of-speech tagging and named entity recognition for Greek text. Both approaches are necessary for a document analysis system that allows searching and cross-referencing entities on a set of documents.

After data processing there appears a need to find a convenient format that would be easy-to-use and at the same time have enough capabilities to represent all results of the analysis. One of the most popular approaches to represent complex data is Resource Description Framework or RDF [10]. RDF is a language specification that describes how to construct data models with a large variety of relationships between entities. RDF is based on a directed graph that is built out of three different elements: a node for the subject, an arc that represents a predicate and goes from subject to object and a node for the object. Each of these elements could be identified by a uniform resource identifier. As for the project, the whole amount of analyzed data could be represented through RDF. For example, a document can be a subject node, a named entity extracted from the document can be an object node, and an arc between these nodes would represent that the document contains this named entity.

Study [11] shows how an RDF-based approach could be used for semantic search tasks. The main idea of study is that both search query and the data on which the searching is performed should be represented as RDF graphs, and the searching algorithm would compare query and data graphs and return matching occurrences as a set of nodes and relationships. In this project that approach could be applied to find similarity between two documents with results of search representing references between these documents.

III. Methods

The project is about developing an application so the first stage will be collecting and analyzing requirements that the system should meet. For this stage, the main method would be the user story mapping [12] as it is an industry-standard in software engineering. User story mapping helps to split all system functions into separated user stories that tell why users need this feature and what value it will give to them.

For the stage of designing the architecture of the system will be used UML diagrams. They allow us to illustrate data models, how are they related to each other and sequences of separated system parts calling each other to perform a function. UML diagrams help achieve simple yet effective architecture that will be easy to maintain and scale.

The application will be developed on Python 3 programming language because it has a big variety of external tools and libraries for text processing and web application development. In particular, for web interfaces will be used Django framework as it provides full support for the MVC model and a convenient way to develop a REST API. The RDF standard will be used for semantic annotation representation inside the system and for data import and export from the application. The RDF provides a format for representing graphs and allows to store information about entities and relations between them. As for searching in semantic annotation, there will be used SPARQL query language. It is a special query standard introduced for RDF documents and it allows to catch subgraphs with particular entities and relations from the entire source graph.

Quality assurance of the system will be achieved by performing functional testing of the app that aims at the correct operation of the high-level functions (such as making a semantic annotation or searching in one) against specifications and requirements. In addition, unit testing will be involved because it helps to test the smallest modules of the system and make sure that low-level functions are working properly.

Quality assurance of system will be achieved by performing a functional testing of the app that aims on correct operation of the high-level functions (such as making a semantic annotation or searching in one) against specifications and requirements. In addition, unit testing will be involved because it helps to test the smallest modules of the system and make sure that low-level functions are working properly.

IV. Anticipated Results

The main anticipated result of the project is a working application with REST API that allows to create a semantic annotation for a given text, saving it and making search queries. Specification of the system will be represented with UML diagrams and a list of non-functional requirements. Besides, the study will contain a detailed technical task that will describe specifications, requirements, delimitations and stack of technologies.

The user requirements and supported functions will be documented in the use case diagram. The architecture of the system will be represented in ER (entity-relation) diagram, sequence diagram and class diagram. Technical documentation will also include users' guide and maintenance guide. All that documents is enough for further development and maintenance.

In addition to that, the paper will include an analysis of current NLP approaches, methods of semantic search and text processing tools. There also will be a comparison of existing natural language processing frameworks for Python programming language that will represent specific features that significant semantic search task.

V. Conclusion

The proposed project is aiming to analyze current situation in the field of semantic search and come up with a system that allows to create semantic annotation for any given text. The system can be used in creation of a complex search engine for a legal document database for people to analyze law cases and develop new legal acts.

The advantages of proposed system are web-based interface and architecture, modular structure and modern NLP methods that used inside the app. Those features allow to continue development and scale and grow the system even furtherfafd.

REFERENCES

[1] Mohamed Zakaria Kurdi, Natural language processing and computational linguistics. 1, Speech, morphology and syntax. London, Uk: Iste ; Hoboken, Nj, Usa, 2016.

[2] "A General Approach to Preprocessing Text Data" [Online]. Available: https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html (Accessed: Dec. 22, 2021).

[3] V. Balakrishnan and L.-Y. Ethel, "Stemming and Lemmatization: A Comparison of Retrieval Performances," Lecture Notes on Software Engineering, vol. 2, no. 3, pp. 262–267, 2014, doi: 10.7763/lnse.2014.v2.134.

[4] M. Popovič and P. Willett, "The effectiveness of stemming for natural-language access to Slovene textual data," *Journal of the American Society for Information Science*, vol. 43, no. 5, pp. 384–390, Jun. 1992.

[5] "Text Wrangling & Pre-processing: A Practitioner's Guide to NLP" [Online]. Available: https://www.kdnuggets.com/2018/08/practitioners-guide-processing-understanding-text-2.html (Accessed: Dec. 22, 2021).

[6] E. Loper and S. Bird, *NLTK : the Natural Language Toolkit*. S.L.: S.N.], [S.A.

[7] "Industrial-Strength Natural Language Processing" [Online]. Available: https://spacy.io/ [Accessed: 22-Dec-2021].

[8] Lobur, Mykhailo, Andriy Romanyuk, and Mariana Romanyshyn. "Using NLTK for educational and scientific purposes." "11th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM 2011)," *IEEE Transactions on Electron Devices*, vol. 57, no. 11, pp. 3192–3192, Nov. 2010, doi: 10.1109/ted.2010.2089250.

[9] E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis and K. I. Diamantaras, "Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy," *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 337-341, 2019.

[10] A. Tripathi, "Resource Description Framework (RDF) for Organised Searching on Internet," *DESIDOC Bulletin of Information Technology*, vol. 21, no. 4 & 5, pp. 3–7, Jul. 2001, doi: 10.14429/dbit.21.4.3545.

[11] Zhu, Haiping, Jiwei Zhong, Jianming Li, and Yong Yu. "An Approach for Semantic Search by Matching RDF Graphs." In *FLAIRS Conference*, pp. 450-454. 2002.

[12] "The New User Story Backlog is a Map" [Online]. Available: https://www.jpattonassociates.com/the-new-backlog/ [Accessed: 22-Dec-2021].