TextMania: Intelligent Text Analytics

Bc. Dávid CSOMOR*, Bc. Adam ĎURIŠ*, Bc. Alan KOVÁČ*, Bc. Daniel KOVÁČ*,
Bc. Peter KRIŽAN*, Bc. Patrik MELICHERÍK*, Bc. Krištof ORLOVSKÝ*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
team5fiit@gmail.com

Nowadays, there is a huge amount of digitized written information, which surrounds us and is growing every minute. Our inability and lack of time to carefully read and organize every new article and categorize it, requires a tool for automated text analysis and tag recommendation. This tool should be able to import new texts, categorize them based on provided tags train the machine learning model and allow future texts to be categorized based on that pre-trained model. There are already several tools like this, but none of them is useful for Slovak language.

With this purpose in mind, we are developing a web-based environment with Node.js backend for analysis of text documents in Slovak language. We use technologies and advancements from the field of machine learning and automated text processing to achieve automated tagging and categorization.

In this paper, we focus on importing, analysing and automated categorization of new articles based on their content. Using our solution, we can easily import and categorize new articles based on our pretrained machine learning model.

Our proposed text analysis environment is designed with intent to be user friendly above all, so it can be used not only by data scientists, but also by regular users (without programming or scientific background). This should lead to improvements of our machine learning model, because we should have more input data. It is very important for us to make the process as easy and accessible as possible, so users can help us grow our datasets and improve our algorithms.

Our environment is focused on working with large collections of texts in Slovak language (e.g. various news articles, encyclopedia articles or any other text). For initial tracking, we firstly obtained articles from Slovak instance of Wikipedia[1] and Slovak popular news site Webnoviny[2]. Then we tagged them with keywords from the source page. Texts of these articles are stored in MongoDB, which is a flexible and scalable NoSQL database useful for textual data.

The first text processing task in our pipeline is word-by-word analysis. For this we use a third-party web-application NLP4SK[3], which is a set of tools for natural language processing. NLP4SK provides tools for text tokenization, sentence identification, lemmatization, part-of-speech (POS) tagging, named entity recognition (NER) and some other features.

After processing, all data are available for browsing and visualization in our web-application. Our main use case of text analysis is:

1. Import article(s) from several sources (*.txt or URL).
2. Store article(s) in MongoDB database.
3. Obtain basic information about imported text using the NLP4SK.
4. Insert analysis results into database.
5. Create and update inverted index.
6. Retrain machine learning model used for automatic categorization.
7. All articles and its attributes are available for browsing in our web-application.

Articles added during the development were manually tagged by our team, which provided a starting point for our machine learning process. We analyzed the inserted text and stored the retrieved data such as grammatical categories (*POS* – part-of-speech), identified entities (*NER* – named-entity-recognition) such as location and person, lemmas and N-Grams.

---

* Master degree study programme in field: Software Engineering
Supervisor: Dr. Miroslav Blšták, Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] https://sk.wikipedia.org

[2] https://www.webnoviny.sk

[3] Link to NLP4SK: http://arl6.library.sk/nlp4sk/

IIT.SRC 2019, Bratislava, April 17, 2019, pp. 1–2.

Our solution is based on two main parts. Figure 1 shows an overview of the system architecture. Client side of the application is built on Angular[4] framework, which provides flexibility for the frontend. User interface currently allows user to:

- Add new articles from supported sources.
- Show visualized statistical data based on stored articles.
- Search and browse articles based on specified criteria.

Inverted index section, shown on Figure 2, allows user to search our database based on:

- Category name
- Selected words
- Article names
- Part of speech specification
- Named entity specification
- Word's lemma
-

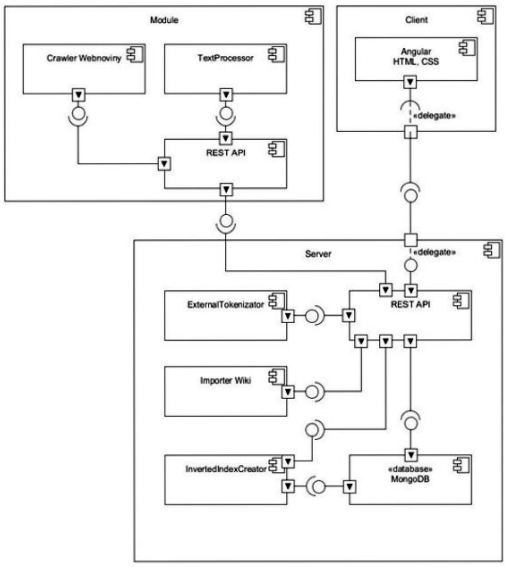

Figure 1.System architecture.

Server side of the application is built on Express[5] framework and running in Node.js environment. This component provides database access and routing. In this part are implemented "Wikipedia importing", "inverted index creation" and all other features mandatory for the client side.

There is also component powered by Django[6] framework which is the backbone of our machine learning processes. In the Python module, there is also implemented "articles importing from "Webnoviny".

All these components are communicating with each other through the HTTP and are independent. This allows you to deploy the Python component to another server and thus facilitate processing on the JavaScript component (the client-side searching through articles will be still smooth).

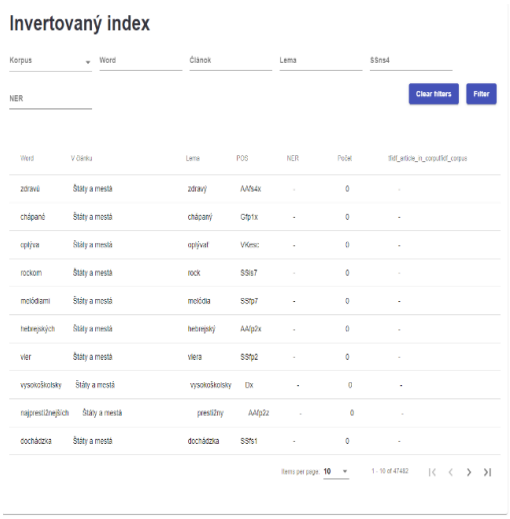

Figure 2. Application preview – inverted index view.

Final product will allow users to add new articles to our solution through our web page interface, where it will be processed, tagged and stored for later analysis by the user. Regular users could use it for categorization and researchers can use it for data analysis purposes, since they will have detailed information about this process and so they should be able improve their algorithms.

[4] Link to angular framework: https://angular.io
[5] Link to express framework: https://expressjs.com
[6] Link to Django: https://www.djangoproject.com