End-to-end Acronym Expander System for Bulgarian language

Monthly Report 2 for MSc IS

Iliya Georgiev
University of Amsterdam
Amsterdam, The Netherlands
iliya.georgiev@student.uva.nl

1 DATA COLLECTION AND PREPARATION

In this section, the author will discuss the process of data collection and data preparation.

1.1 Data collection

The data will be collected from Wikipedia - Bulgaria[1], as well as different online newspapers and governmental websites. The extraction of the articles will be done within 10 generic categories - Psychology, BioMedical, Sports, Politics, Financial, Engineering, Computer science, Math, Education, and Biology. The author will manually extract at least 20 documents per topic and will save these documents as *.txt* files on the server. The articles will be randomly selected without taking any requirements into account. These documents will be later annotated by the author and external readers using the Acronym Expander Annotation System.

1.2 Data preparation - Acronym Expander Annotation System

To be able to use the gathered data for this research, it needs to be pre-processed. This process includes text pre-processing of the Wikipedia data and merging multiple documents to a data set. A custom annotation system was built by the author and other participators in the project in order to simplify the annotation process. The system is built upon Python[2] using Flask[3] as a web framework. It uses the Flask Templates to serve the frontend to the users. The frontend of the system is built upon HTML, CSS and JavaScript, using Bootstrap as a front-end framework.

This system allows external readers and the author to annotate the collected data. This is done by providing the user with an extracted text and asking him/her to match all acronyms within the text with the right expansion if there is one. In case the expansion is not found in the text, the user will have to annotate the acronym and its expansion according to the context of the text. Moreover, the annotators should provide the language of all of the acronyms that they found. After the annotation is performed, the annotated acronyms and their expansions will be stored in the database. The system keeps track of all of the articles that are annotated by the user in order to make sure that a user will not annotate the same article twice. Moreover, since each of the articles should be annotated by two different readers, the system prioritize articles that have not been annotated yet, or the ones that have been annotated only one time. Furthermore, the system uses a smart article lock process, which assigns a document to annotator for X hours. If the annotator do not finish the task in this time, then the system will assign this document to another user in order to prevent articles being not annotated due to lack of user activity.

After the annotation is performed, then the text of the annotated document with the annotated acronyms and their expansions will be stored on the server. This collection will be then merged and converted to a data set used to test the system. The data will be distributed into 3 different data sets to be used for the three different parts of the system - Acronym Identification(AI), Acronym Disambiguation(AD), and Acronym Expansion(AE). The AI data set will compose of more articles within text expansion, while the AD data set will mostly use articles where the expansion is not inside the text. For the AE, all of the articles will be used for a test data set.

[1] https://bg.wikipedia.org/wiki/%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D0%BD%D0%B0_%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0
[2] https://python.org
[3] https://flask.palletsprojects.com/