CLASSLA Wikipedia

This document provides a full account of the completed work and development approaches behind the project CLASSLA Wikipedia.

Outline:

- Introduction
- Approach
- Description of corpora
- Description of the Github repository

Introduction

The project has the task of generating Wikipedia corpora in seven south-Slavic languages, namely Macedonian, Bulgarian, Serbian, Croatian, Serbo-Croatian, Slovene, Bosnian. The corpora are generated using Wikipedia dumps which can be found on the Wikimedia Dumps website (links below). The Wikipedia dumps were downloaded on October 17th 2020 and contain all seven corpora downloaded from:

Serbian: https://dumps.wikimedia.org/srwiki/20201020/
Serbo-croatian: https://dumps.wikimedia.org/shwiki/20201020/
Slovene: https://dumps.wikimedia.org/slwiki/20201020/
Croatian: https://dumps.wikimedia.org/hrwiki/20201020/
Bulgarian: https://dumps.wikimedia.org/bgwiki/20201020/
Bosnian: https://dumps.wikimedia.org/bgwiki/20201020/
Macedonian: https://dumps.wikimedia.org/mkwiki/20201020/

Approach taken to generate the corpora

The seven downloaded files had a .gz extension, indicating they were zipped. After the files were unzipped, a specific tool was used with the ability to open the dumped files, and extract relevant parts of the dumped wikipedia corpus. The tool mentioned is titled WikiExtractor and is documented here (https://github.com/attardi/wikiextractor) Parts taken from the wikipedia corpus and maintained in the processed file are links, section titles, and lists. The output from the WikiExtractor tool yields several folders (titled AA, AB, AC… in alphabetical order), each containing files with html dumps from the wikipedia corpora, rich with HTML tags. This concludes the usage of the WikiExtractor tool.

Once the dumped files are stored in separate folders, a python module was developed for additional processing. The module was developed to be language-agnostic and it can be applied to all the seven corpora. The module itself contains four levels of processing of the dumped text, outlined as follows:

1. Usage of the scrapy python library to remove all relevant HTML tags from the corpora.

2. Capturing various relevant parts of the wikipedia article, storing them elsewhere temporarily while other processing is conducted, and afterwards reinjecting them into the corpus. This is relevant for cases like URLs, shortened URLs, ellipsis (...), dashed or numbered lists, intralinks within the articles themselves, etc. The various elements to be captured are defined using a regular expression.
3. Substitutions of text with other pieces of text based on regular expressions.
4. Substitutions of text with other pieces of text based on the .replace() method.

The python script was developed within a Jupyter notebook titled CLASSLA_Wikipedia.ipynb containing several parts:

- Imports
  A cell containing all relevant imports to the Python packages used for processing the wikipedia corpora.
- Terminal Section
  As the development was undertaken on a Linux system, the terminal section allows the developer to navigate the Linux system for easier exploration within the Jupyter notebook. This can (and should) be modified according to the needs of the user of the notebook.
- Python Section
  Definition of the EntityTemplate class (level 2 processing), which captures textual entities in a generic way thereby enabling their protection from other processing modules which are present in the text processing pipeline (level 3 and 4).
- For one File
  Definition of all EntityTemplate objects and subsequent ordering of the four levels of processing as described earlier. The cell defines several functions; the carrier of the majority of the functionality is the iterate_item() function.
- For all Files
  This cell calls the iterate_item() function found in the “For one File” cell and executes it for all relevant files in a parallelized fashion.

The output from the Jupyter notebook is generated within the For all Files cell, resulting in the same folder structure as the output from the wikiextractor. All files from this output have an “_m.txt” added to them, the “m” signifying “modified” and the “.txt” making the document more accessible across multiple operating systems.

Description of corpora

The corpora are kept in separate smaller files and deliberately not joined within one large flat file. This is done so that they can be more easily digested by users with a variety of resources at hand. All files are stored in .txt format, within the aforementioned folder structure. The files maintain three tags:

- <doc> signifying the ID of the document / wikipedia article.
- <formula> signifying a mathematical formula.
- <section> signifying the title of the section within the wikipedia article.

Description of the Github repository

The resulting code from this project has been added to two Github repos, https://github.com/clarinsi/classla-wikipedia and https://github.com/filipmarkoski/Phobos-1.

The project structure on the github repo is:

- `data`
  Contains the original files after their download and processing using WikiExtractor.
- `output`
  Contains all the output corpora, within seven folders, one for each language.
- `CLASSLA_Wikipedia.ipynb`
  The main notebook, containing all the code as described in section Approach.
- `Deprecated_Code.ipynb`
  A notebook containing all deprecated code, no longer used within CLASSLA_Wikipedia.ipynb, however may become relevant in the future.
- `definitions.py`
  A module defining variables pointing to relevant folders in the project directory.
- `README.md`
  README file containing all contextual information regarding the CLASSLA Wikipedia project.
- `requirements.txt`
  Documented requirements for running the CLASSLA_Wikipedia.ipynb notebook.