Summer Semester 2021
Data & Knowledge Engineering
Heinrich-Heine-Universität
Düsseldorf

Prof. Dr. Stefan Dietze (lecturer)
Katarina Boland (assistant)

Optional Home Project

PLD Media Bias Classification

This home project exercise can be completed at home, using the expertise and skill sets acquired in previous lectures. All students are invited to present solutions in the exercises for grade improvement. In order to participate, please proceed as follows:

- Indicate your intention to submit and present by **24.05.2021**
- Use the folder structure in https://git.gesis.org/dke_2021_homeproject/home_project_students to prepare your solution at home.
- Provide the source code and generated output via your Git repository by **19.06., 3:00 pm** CEST time zone. To this end, it is sufficient to send a link to your **public** repository containing the source code and generated output via mail
- Present your solution (approach, results) to the class during the DKE 2021 lecture on **29.06.2021** (detailed arrangements about presentations will be announced later)

Presenting and submitting a working solution (on time) that fulfills requirements will result in a 0.3 grade improvement in the final exam.

The home project deals with the classification of media bias of PLDs linked in tweets. I.e. given a link to e.g. a news outlet in a tweet, is this news outlet left-biased or right-biased? For this project, we will use the tweets and their metadata for classification.

Prerequisites:

1. Clone/Fork the provided Gitlab repository into your own GitHub account to perform the tasks on your local machine.
2. The TweetsCOV19 knowledge base can be found here: https://data.gesis.org/tweetscov19/ Familiarize yourself with the data and the data model, i.e. the used schema for representing tweet data.

Detailed Task Description

Your task is to retrieve tweets from TweetsCOV19 and classify the PLDs of referenced URLs according to their political leaning / their bias: left or right, based on the content and

metadata of the tweets. As Ground Truth, you are given a list of PLDs with bias ratings that you may use for training your algorithm.

1. Use the SPARQL endpoint of TweetsCOV19 to retrieve all tweets along with all relevant metadata that reference at least one of the PLDs in the training dataset. For obtaining data, use the SPARQL endpoint (your code should show how data has been fetched). You may hydrate the tweets in order to use their texts (these are not included in TweetsCOV19). You may also retrieve full texts of cited PLDs or URLs.
2. You may use data from the TweetsCOV19 dataset exclusively or fetch related data from additional endpoints. For instance, tweet instances in TweetsCOV19 are linked with DBpedia entities about which additional data (e.g. categories, types), can be obtained from DBpedia. You are also free to add external resources such as pre-trained Word Embeddings or resources like WordNet.
3. Prepare a description of your algorithm: which features would you like to use for the classification? E.g. you may use entities, sentiment scores, hashtags, mentions but also other metadata and textual features. Use a machine learning algorithm of your choice.
4. Prepare the feature set(s) of your algorithm.
5. Train your algorithm on the Ground Truth data.
6. Prepare a script that outputs your results in the following format:
    Two CSV files: one file with all PLDs classified as left-leaning, one file with all PLDs classified as right-leaning.
    Each row must have two fields:
        The PLD
        Your confidence score
7. On 16.06 11am, we will add the test data and an evaluation script to the Gitlab repository. Fetch it and update your local repository.
8. Let your classification algorithm classify the test data with the model you trained on the training data, save the output and run the evaluation script.

We expect the git repository you submit to contain all code and dependencies, links to all resources you used or the resources themselves, the CSV files produced on the test data and the output of the evaluation script. Also supply a readme file that explains how your training data can be generated (including your SPARQL queries) and how your model can be trained and applied. You may use Java or Python with the libraries of your choice.

Presentation Guidelines

For your project presentation you have approximately 10 minutes (8 minutes for presentation and 2 minutes for discussion). A good presentation should cover the following aspects:

- The general idea of your approach
- An overview of the architecture and different modules you implemented
- SPARQL queries involved in the solution
- Your results: the performance of your algorithm and some interesting insights, e.g. for which cases does it work well, for which doesn't it?
- Observations, lessons learned and possible future improvements

Evaluation & Rewards

A grade improvement of **0.3** is given when the following criteria is met:

You provided a working solution to the problem and your results are reproducible. You submitted the solution in time and in the specified format. Your classification algorithm yields a reasonable performance on the classification task. You will find the precise score to beat in

the readme file of the git project. You presented your approach to the class during the DKE 2021 lecture on 29.06.

Further Reading and Documentations

- W3C SPARQL Query Language for RDF:
  https://www.w3.org/TR/rdf-sparql-query/

- SPARQL expressions and Function:
  https://en.wikibooks.org/wiki/SPARQL/Expressions_and_Functions

- Wikidata:SPARQL query examples
  https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries

- Interacting with SPARQL endpoints: Java/Jena
  (http://jena.sourceforge.net/)

- Interacting with SPARQL endpoints: Python
  (https://rdflib.dev/sparqlwrapper/).

- Hydrator for tweets:
  https://github.com/DocNow/hydrator

- Python library to extract TLDs from URLs:
  https://pypi.org/project/tldextract/

- Java libraries to extract TLDs from URLs:
  URL getHost()
  https://guava.dev/releases/snapshot/api/docs/com/google/common/net/InternetDomainName.html