Griffith UNIVERSITY

3030/7130ICT

Data Analytics

Lab 05 – Data Analytics for Texts

Table of Contents

1. Text Processing

1.1. Preprocessing

The first NLP exercise is about preprocessing.

You will practice preprocessing using NLTK on raw data.
This is the first step in most of the NLP projects, so you have to master it.

Open the Preprocessing.ipynb notebook and follow the instructions.

1.2. Bag-of-words

Now that we master the preprocessing, let's make our first Bag Of Words (BOW).
We will reuse our dataset of Coldplay songs to make a BOW.

Open the BOW.ipynb notebook and follow the instructions.

2. Text Similarity

2.1. Similarity metrics

We will work on applying similarity: Jaccard and Cosine similarity. This exercise is a simple application.

Open Similarity.ipynb notebook and follow the instructions.

2.2. TF-IDF

We will compute the TF-IDF on a corpus of newspaper headlines.
Open TF-IDF.ipynb notebook and follow the instructions.

2.3. Plagiarism checker (OPTIONAL)

In the folder, you will find source texts (Asource.txt, Bsource.txt, Csource.txt and Dsource.txt) from which some texts were inspired (A1.txt was inspired from Asource.txt and so on). Some are plagiarism, some are regular inspiration.

Your job is to use text similarity to define a plagiarism detection algorithm based on those short examples.

In the examples, there are sources, direct plagiarism (A1, B1, C1, D1) and sometimes examples of so called mosaic plagiarism (D2). Can you make an algorithm that detects all kind of plagiarism without false positive?

Open Plagiarism.ipynb notebook and follow the instructions.

3. Text Classification

It's time to make our first real Machine Learning application of NLP: a spam classifier!

A spam classifier is a Machine Learning model that classifier texts (email or SMS) into two categories: Spam (1) or legitimate (0).
To do that, we will reuse our knowledge: we will apply preprocessing and BOW (Bag Of Words) on a dataset of texts.

Then we will use a classifier to predict to which class belong a new email/SMS, based on the BOW.

Open Spam-Classifier.ipynb notebook and follow the instructions.

4. Word Embedding (OPTIONAL)

Open quora.ipynb and follow the instructions.

Objectives
Quora is a popular website where anyone can ask and/or answer a question. There are more than 100 millions unique visitors per month.
Like any other forum, Quora is facing a problem: toxic questions and comments.
As you can imagine, Quora teams cannot check all of the Q&A by hand. So they decided to ask the data science community to help them to perform automatically insincere questions classification.

Guidelines
This challenge was launched on Kaggle : https://www.kaggle.com/c/quora-insincere-questions-classification
Read the overall information on Kaggle. Quora provided a dataset of questions with a label, and the features are the following:

- qid: a unique identifier for each question, a hexadecimal number
- question_text: the text of the question
- target: either 1 (for insincere question) or 0

The Kaggle dataset is quite heavy and it may be too difficult for your laptops to perform the computations. Therefore, we provide you with **the train dataset** (to be sampled) and also **light word embeddings**, which you can download here

Don't look at the published kernels, in order to keep your judgement unbiased.

Here are a few steps to follow:
1. First sample the dataset to 10.000 lines otherwise your laptop might die (you may need to use sklearn.utils.resample()).
2. As usual, begin with a proper EDA.
3. Perfom a nice text preprocessing.
4. Try to run a quick sentiment analysis using TextBlob.
5. Then, use a word embedding (Glove) to create your corpus and run your model.
6. Do some optimization (text preprocessing, model hyperparameters, other word embeddings if you trust your computer).
7. Optimize ++: Now, let's have a look at some published kernels and find some inspiration.
8. Bonus question : try to identify the most recurrent topics in toxic questions !

In this competition, the metric used for performance evaluation is the **F-score**.