Natural Language Processing, 2022/23

Group Assignment

(released on 22/04/2023, updated with additional dataset on 27/04/2023)

In this assignment you will analyse **publicly available text datasets**, applying the techniques you have learnt in class to train various **machine learning models** to perform various **NLP tasks**. Please read carefully the project description below:

Details of the assignment:

- The assignment must be completed in **groups of 3** (minimum) **to 4** (maximum) students.
- The involves writing a **Jupyter notebook** to analyse the data as discussed below.
- Your notebook should be **self-explanatory**, with **clear descriptions** of the analysis performed and the **conclusions** drawn.

Due date and presentations:

- The assignment is due **on Sunday the 28th of May** at 5 minutes to midnight (via **WeBeep**). Only **one member** from each group should hand-in the notebook, but the **names of all group members should be listed** at the start of the notebook.
- On Monday the 29th and Wednesday the 31st of May, each group will have around **10 minutes** to **present their notebook** during the practical session. (Please don't prepare slides, we just want to see your notebook.)

The assignment will be marked based on the:

(i) appropriateness of methods applied and depth of the **analysis**,
(ii) clarity of the description in the **notebook**, and
(iii) quality of the **presentation**.

THE TASKS

The aim of the assignment is to **apply the techniques you have learnt in class** to analyse **one** of the **text datasets** described below. The exact tasks performed may depend on the dataset chosen, but we would expect to see some of the following:

1. Preliminary analysis:

Briefly describe the dataset:

- what type of documents does it contain?
- how many documents are there?
- calculate and visualise some simple statistics for the collection, e.g. the average document length, the average vocabulary size, etc.

Play around with documents, using some of the code from the early parts of the course. You could, for example:

- cluster the documents and visualise the clusters to see what types of groups are present (or whether the known classes can be found);
- index the documents so that you can perform keyword search over them;
- train a Word2Vec embedding on the data and investigate the properties of the resulting embedding.

2. Training models:

Each of the datasets comes with a particular task that you need to perform, so:

- train a model to perform that task (by fine-tuning models on the training data);
- test pre-trained models on the task (if they already exist); and
- evaluate different models and compare their performance.

HINT: as a minimum here we would expect to see a linear classifier trained on the data (if an appropriate for the task) and compare it with deep learning model, such as BERT.

3. Possible extensions:

Depending on the dataset chosen there will be many additional investigations you can perform. For instance, oftentimes we can improve performance of a model on a particular task by simply including additional data that is related to the task in its training set. So see if you can find other data that helps with the task that you chose. Moreover, there are **many** NLP challenges out there, so if you can't find more data for the task you're working on, look for another interesting challenge to work on.

THE DATASETS

Each **group of 3 to 4 students** should choose **ONE** of the following **six** datasets to work on. To prevent that every group works on the same task, we limit the total number of groups working on each dataset to 10. So having chosen a dataset, please add the names of your group:

- https://docs.google.com/document/d/1_vCGy7X9NTNlj3zSs0jYlQyx5DKjVNiIfuBJUlOURkM/edit

If you are looking for partners to form a group, you can add your name and contact detail to the last table at the link above.

1. KIND (Kessler Italian Named-entities Dataset)

- **Website with data**: https://github.com/dhfbk/KIND
- **Task**: The task on this dataset is to perform Named-Entity-Recognition (NER) from Italian documents.
- The dataset consists of four different collections (degasperi, moro, fiction and wikinews), where each dataset has its own training and test split. (You might compare performance across the datasets.)
- **Hints and suggestions**: As well as testing the NER models you have seen in class, why not compare them with some transformer based NER models (multilingual and Italian-only or even English models) from HuggingFace (https://huggingface.co/models)?

2. EDOS: Emotional Dialogues in Open Subtitles

- **Link to paper**: https://aclanthology.org/2021.emnlp-main.96/
- **Link to data**: https://drive.google.com/file/d/1AqxpBMFVDX-hUGA-treA8ffnSdyu0ggJ/view?usp=sharing
- **Dataset**: The dataset consist of short conversation passages (of 3 to 5 utterances each), each labelled with an emotion.
- **Task**: The original task on this dataset was to generate responses in the dialogue that respect the target emotion, but the dataset could also be used for sentiment analysis: i.e. to identify the emotion associated with each utterance in the dialogue.
- **Hints and suggestions**: You could apply various classifiers for the sentiment analysis task including Transformer based models like BERT. Does knowing the previous (or subsequent) utterance improve the classifier performance? Note also that we will demonstrate text generation techniques in the latter practical sessions of the course.

3. STS: Semantic Textual Similarity

- **Website with data**: http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark
- **Dataset**: The dataset consists of pairs of sentences that have been judged by humans on how similar they are to one another. Here the similarity indicates the level of (semantic) agreement between the meaning of the sentences. The sentences have been sourced from three domains (news, caption and forum).
- **Task**: the original task here is to predict the similarity between the sentences. Note that this problem could be handled as either a regression (predict a real score) or classification (predict similar/non-similar label) task.
- **Hints and suggestions**: You could investigate various *unsupervised* techniques, such as computing the cosine similarity between TF-IDF bag-of-words vectors for the sentences, extending the BOW vector to include n-grams (bigrams and trigrams), and also consider the distance between average word embeddings. Moreover, *supervised* models could be trained, in particular transformer-based models are often fine-tuned to classify pairs of sentences.

4. AUTEXTIFICATION: Automatic Text Identification

- **Link to data:** https://polimi365-my.sharepoint.com/:u:/g/personal/10690519_polimi_it/EbjPT33UrF9GrxR_0KYeI2cBoloZDxQoCIi7M9tDoepHpg?e=IAiGRg
- **Link to Website**: https://sites.google.com/view/autextification
- **Dataset**: The dataset consists of short text passages that have either been written by a human or have been generated automatically by a Language Model.
- **Task**: The task on this dataset is to detect text which of the text has been automatically generated by a language model. There is also a second task in which one should also identify the particular Language Model that generated the text.
- **Hints and suggestions**: The text is very short so this may be a difficult problem. This is an authorship identification task. You could try to train a classifier on Part of Speech (POS) information, and/or look at syntactic information, such as spelling errors and the how complicated the vocabulary is (e.g. how long are the words on average). More sophisticated techniques might look to compute the log-likelihood of the text under various language models.

5. SQuAD2.0: The Stanford Question Answering Dataset

- **Website with data**: https://rajpurkar.github.io/SQuAD-explorer/
- **Dataset**: The dataset consists of a set of questions and Wikipedia articles containing the answers to the questions.
- **Task**: The original task for this dataset is to find answers to the question or to respond that the question is unanswerable given the information available.
- **Hints and suggestions**: Traditionally, question answering systems were trained either to locate the answer in the text, or to produce only a fixed set of possible answers by classifying the question into classes that corresponded to the most frequent answers to questions. You could try to train a model to work in that way. More recent QA systems simply generate the response text. You could test a GPT2 model or compare other models from HuggingFace (https://huggingface.co/models).

6. WizardOfWikipedia: Knowledge-powered Conversational Agents

- **Link to paper**: https://arxiv.org/pdf/1811.01241.pdf
- **Link to dataset**: https://parl.ai/downloads/wizard_of_wikipedia/wizard_of_wikipedia.tgz
- **Dataset**: The dataset consists of open-domain dialog containing utterances that were generated from content extracted from Wikipedia articles. Multiple candidate Wikipedia passages are associated with each turn in the dialog, where one of them is considered the “reference” (i.e. correct) passage and was used to generate the response text.
- **Task**: The original tasks for this dataset was to (i) choose the correct passage for each turn and (ii) generate the response in the dialog using the information contained in the reference passage.

- **Hints and suggestions**: You could investigate one or both of the tasks described above. Moreover, you could even try to tackle the third task of retrieving the candidate set of passages for each turn in the dialog.