Report session 6-7 NLP

Cristina ORTEGA LÓPEZ

Assignment 7: Topic modelling

We wanted to retrieve, clean and analyse the tweets of French politicians using tools like tweepy, twikit, or scraping with snscrape. We visualized the top words with wordclouds.

First, we pre-processed the dataset, which comes from the group newsgroups, uploaded from a pickle file. We applied basic cleaning techniques like conversion to lower cases, eliminating emails, punctuation signs, and stopwords, and filtering tokens with length of over 4 characters. This resulted in a cleaned and vectorized text, prepared for modelling.

Then, we applied two main topic modelling techniques: LDA and NMF.

For LDA, we based the counting of words with CountVectorizer, we defined 10 topics and then visualized the top representative terms in graph plots.

For NMF, we used TF-IDF for representation. We also defined 10 topics and generated similar visualizations to analyse top terms by topic.

We generated word clouds for words in each topic, showing visually the relative importance of each term (see file in GitHub). We used library wordcloud.

Assignement 8.1: Sentiment analysis (VADER)

The aim of this first assignment is to apply the sentiment analysis VADER on tweets and comparing its results versus other modern approaches like the Transformers one.

The dataset used was tweet-data.csv.

The tool used was SentimentIntensityAnalyzer from nltk.sentiment.vader.

First, we cleaned our texts with the function clean_tweet() that we will also use in sentiment analysis with Transformers.

Then we applied VADER on column cleaned_tweet. We obtained a score dictionary: neg, neu, pos, compound.

It was assigned a final label: positive if compound was higher than or equal to 0.05, negative if compound was lower than or equal to -0.05, and neutral in the rest of the cases.

neutral 277

negative 118

positive 105

Name: sentiment_label, dtype: int64

VADER detected more neutrality in comparison with Transformers (see Assignment 8.1.). there is a high sensitivity to score and words with emotional charge.

VADER is quick, light and useful for informal language (as that of social media). It doesn't require previous training, but its performance can be limited versus contextual models like BERT. It can be confused with sarcasm, irony or lack of context.

Assignement 8.1: Sentiment analysis (ML - Transformers)

The objective was to apply a sentiment analysis model (distilbert-base-uncased-finetuned-sst-2-english) of Hugging Face on a sample of 500 tweets.

The dataset used for this sample was tweet-data.csv.

We cleaned the text by eliminating URLs, mentions, hashtags and special characters.

Then, we applied the pipeline of sentiment-analysis from Transformers.

We obtained two new columns called sentiment_label (positive or negative) and sentiment_score (confidence level of the model).

Most of the tweets were classified as positive. The sentiment_scores were mostly near to 1, which indicates a high confidence in our model.

Overall, the model proved a good performance. Scores near 1 reflect a high confidence in the model. Cleaning the texts helped to enhance the quality of this analysis.

Assignment 9.1: LLM Usage – Email classification

The aim was to use a LLM model to classify emails into spam or not spam, by applying prompt engineering techniques.

The dataset used was spam.csv. Columns in this dataset were "text" (content of the email) and "target" (real label).

We charged the dataset and explored the columns mentioned. We used the GPT2 model from Hugging Face with the pipeline text-generation.

We designed the following prompt: “Classify this email as “spam” or “not spam”. Only reply with “spam” or “not spam”. Email: {email}. Label:”

We extracted the generated prediction from the model and then compared it visually with some real emails.

Even though GPT-2 was not trained specifically for classification, thanks to the well designed and short prompt (max_new_tokens=10), we succeeded in generating clear answers. However, the model can generate unwanted additional text without a good prompt engineering technique.

Assignment 9.2: Chatbot with LLM

Objective was to create a simple app in Streamlit for introducing emails and classifying them into spam/not spam.

We generated an app.py file that asked the user to write their email, uses a pipeline with GPT2, and shows only the generated label as the answer.

We run streamlit run app.py. App is working and demonstrates how we can integrate NLP models in an accessible interface. It can be improved by adding processing by lots or uploading CSV files to classify multiple emails.