Vectors & Text Classification

Exercise 2

Pranav, Carolin and Karina
pranav.agrawal@uni-hamburg.de, karina.vida@uni-hamburg.de

1 Vectors

You are planning to make a news aggregation website. Your platform wants to cluster together similar news and social media articles. For this, you plan to use a bit of NLP magic and vectorization.

Assume that you have the following texts:

d_1 = St. Pauli won against Bayern Munich by 3-0.
d_2 = omg stream dua lipa's leaked album in our house!
d_3 = ST. PAULI JUST SLAYED BAYERN MUNICH YAAASSS!!
d_4 = There is a pipe leak in our house.

Perform the following steps:

1. Process these texts. Convert them into lowercase, normalize noisy stuff, tokenize it etc.
2. Develop binary vectors of each text here.
3. Develop count-based vectors of each text here.
4. Develop TF-IDF based vectors of each text here.
5. Develop a cosine similarity based matrix of these documents.
6. You want to cluster similar documents together. For that you will use a cosine-similarity based matrix here. What's the threshold that you will use to cluster?
7. Suppose you have a new text that says, 'Bayern Munich won against St. Pauli by 3-0.' Using these similar algorithms above, where would it cluster? Why?
8. What disadvantages of bag-of-words algorithms have you observed in this exercise so far?

2 TF-IDF Search

One of the biggest advantages of TF-IDF algorithms is that they are FAST. They are commonly involved in searching databases and autocorrects on your phone.

We will look into spelling correction more in detail.

Firstly, we expand the words using n-grams models. Here we will go for unigrams and bigrams.

A unigram will look like this: $Hello = \{< s >, H, e, l, l, o, < /s >\}$. Here, <s> and $< /s >$ indicate the start token and the end token respectively.

A bigram will look like this: $Hello = \{< s >, He, el, ll, lo, < /s >\}$.

When combined, they will look like this: $Hello = \{< s >, H, e, l, l, o, He, el, ll, lo, < /s >\}$

1. Expand the following words: **apple** and **aple**. After expansion, compute the similarity score between the two sets.

2. For TF-IDF purposes, consider only bigrams. Assume that you have following words in the vocabulary:

- apple
- banana
- yoghurt

Execute the spelling correction algorithm for this and show your steps with an example. Your algorithm will consist of the following steps:

1. Firstly, you will expand every word in the vocabulary using bigrams.
2. Then, you will compute IDF scores of every term here.
3. Then you will take a misspelling and expand that using bigram.
4. After that you will compute similarities of that misspelling to every word in the vocabulary.
5. Return the match which has the highest similarity.

3 Classification: In-context Learning

In-context learning (ICL) is a technique where task demonstrations are integrated into the prompt in a natural language format. This approach allows pre-trained LLMs to address new tasks without fine-tuning the model.

In-context learning (ICL) is known as *few-shot learning* or *few-shot prompting.* Contrary to conventional models, the knowledge accumulated via this method is transient; post-inference, the LLM does not persistently store this information, ensuring the stability of model parameters.

Here's an example of ICL:

Prompt: Classify the sentiment of the following text as positive, negative, or neutral.

Text: The product is terrible. Sentiment: Negative

Text: Super helpful, worth it Sentiment: Positive

Text: It doesnt work! Sentiment:

GPT-4 Response: Negative

In the last exercise, you came across the data cleaning. While curating the dataset, we might want to filter out some unwanted data (like highly confidential data, data containing hate speech and so on). For this, you can use ICL to check whether a particular text would be a good candidate for the dataset or not.

1. Create a dataset with a few positive and negative examples regarding data cleaning.
2. Create a well-designed prompt with the dataset above to classify a text, whether it should be included in a dataset or not.
3. This is just an exercise! Please do not use ICL for purposes like this in actual practice. What harms do you think might happen if we use something like this for data filtering?