Week 09 - Lab Session Results

February 26, 2025

Text Representation

Exercise 1

Load the metadata file from Absalon and discard any item that was not rated by our subset of users (not in training or test sets). You can refer to the original metadata file if you want to look up more explanations about the columns of the metada file. Apply preprocessing in this order: lowercasing, tokenizing, stemming, and stopwords removal (including punctuation) to clean up the text from the `title`. Report the vocabulary size before and after the preprocessing. You may have to specify the language for these steps.

```
Total number of items: 84
```

```
Vocabulary size before preprocessing: 545
Vocabulary size after preprocessing: 471
```

Exercise 2

Representation in vector spaces.

2.1

Represent all the items from Exercise 1 in a TF-IDF space. Interpret the meaning of the TF-IDF matrix dimensions. Be careful with multiple instances of preprocessing in the process, as default settings for creating the TF-IDF space may include some.

Tip: You may use the library scikit-learn

```
TF-IDF matrix shape:  (84, 471)
```

2.2

Using the TF-IDF representation, compute and the cosine similarity between products with asin `B000FI4S1E`, `B000LIBUBY` and `B000W0C07Y`. Take a look at their features to see whether results make sense with their characteristics. Round your final answer to 3 decimal places.

```
Similarity between 'B000FI4S1E' and 'B000LIBUBY':  0.031
Similarity between 'B000FI4S1E' and 'B000W0C07Y':  0.024
Similarity between 'B000LIBUBY' and 'B000W0C07Y':  0.501
```

Exercise 3

Representation in vector spaces with contextual Word Embeddings.

3.1.

Represent all the products from Exercise 1 in a vector space using embeddings from a pre-trained BERT model. The final embedding of a product should be the average of the word embeddings from all the words in the 'title'. Critically evaluate this procedure.

What is the vocabulary size of the model? What are the dimensions of the last hidden state?

Tip: you may install the transformers library and use their pretrained BERT model uncased.

```
Using device: cpu
Vocabulary size of 30522. Input dimension: 768.
```

```
last_hidden_states: torch.Size([84, 52, 768])
```

3.2.

Using the representation obtained from Exercise 3.1., compute the cosine similarity between items with asin `B000FI4S1E`, `B000LIBUBY` and `B000W0C07Y`. Round your final answer to 3 decimal places.

```
Similarity between 'B000FI4S1E' and 'B000LIBUBY':  0.734
Similarity between 'B000FI4S1E' and 'B000W0C07Y':  0.659
Similarity between 'B000LIBUBY' and 'B000W0C07Y':  0.748
```