Report Group 21

Exercise 2

WS 2021 - 188.977 Grundlagen des Information Retrieval

Tobias Eidelpes, Arianna Pera, Michael Reder

Part1: Warmup

Word1	**Word2**	**Cosine Similarity**
cat	dog	0.7502
cat	Vienna	0.1706
Vienna	Austria	0.7769
Austria	dog	0.2198

As one could expect, *cat* is more similar to *dog* than it is to *Vienna*, as *Vienna* is more similar to *Austria* than it is to *dog*.

Word	**Top-1**	**Top-2**	**Top-3**
Vienna	Salzburg	Austria	Prague
Austria	Austria-	Vienna	German-Austria
cat	cats	housecat	-cat

The top-3 most similar words to *Vienna* are *Salzburg*, *Austria* and *Prague*. This result makes sense since it considers two other cities (one in Austria and the other in a closeby Country) and the Country of which Vienna is the capital of.

The top-3 most similar words to *Austria* are *Austria-*, *Vienna* and *German-Austria*. This results retrieves a token very similar to the provided one (except for the '-' punctuation), the capital of the provided Country and an historical definition of Austria as German-Austria, a Country created after WWI.

The top-3 most similar words to *cat* are *cats*, *housecat* and *-cat*. All of these results are very similar to the provided token, syntactically or semantically.

Part2: Short-Text Similarity

Method	**Preprocessing**	**Pearson Correlation**
Vector Space Model (from sklearn library)	Lowercasing + Stop word removal	0.734
Average Word Embedding	Lowercasing + Stop word removal	0.699
IDF Weighted Agg. Word Embedding	Lowercasing + Stop word removal	0.716
Vector Space Model (from sklearn library)	Lowercasing	0.730

Average Word Embedding	Lowercasing	0.653
IDF Weighted Agg. Word Embedding	Lowercasing	0.702

By observing the results, one can tell that the Vector Space Model seems to achieve the best results with both pre-processing architectures. The second best performing method is IDF weighted average of word embeddings, which achieves good results with both of the pre-processing functions.

Generally speaking, the full pre-processing (also considering stop words removal) outperforms the simple one (which only takes into account lowercasing).

Part3: Training new language models

Word (of your choice)	**Top-1**	**Top-2**	**Top-3**
Planeten	Himmelskörper	Mond	Kometen
Arzt	Zahnarzt	Psychiater	Augenarzt
Winter	Sommer	Frühsommer	Wintermonaten

The top-3 most similar words to *Planeten* are *Himmeslkörper*, *Mond* and *Kometen*. The result makes sense because the last two of them are celestial bodies, and the top 1 summarizes the token and the two others.

The top-3 most similar words to *Arzt* are *Zahnarzt*, *Psychiater* and *Augenarzt*. The token *Arzt* also summarizes the top 3 in their topic, furthermore it can be seen that the token is included in top 1 and top 3.

Ther top-3 most similar words to *Winter* are *Sommer*, *Frühsommer* and *Wintermonaten*. Here the top 3 are similar to the token as they describe seasons. Whereby with summer the exact opposite of winter is meant.

We used a dataset containing German Wikipedia entries for our training (GitHub). It contains a 6gb big .txt file which is cleaned and separated into sentences. One advantage of the dataset is that it also contains comments which are more written in sloppy languages. This is helpful when it comes to process chats or other sloppy tasks. To reduce the dataset we only retrieve the first 1073741824 bytes from the file and save them into a smaller file.

At first we tried using the recommended dataset from Twitter but it also contains other languages that were not marked correctly and could not be filtered properly.