Report Group 05

Exercise 2

WS 2021 - 188.977 Grundlagen des Information Retrieval

Part1: Warmup

Word1	Word2	Cosine Similarity
cat	dog	0.7502321600914001
cat	Vienna	0.17059049010276794
Vienna	Austria	0.776897668838501
Austria	dog	0.2197847068309784

Word	Top-1	Top-2	Top-3
Vienna	'Salzburg', 0.7848144769668579	'Austria', 0.7768975496292114	'Prague', 0.7675193548202515
Austria	'Austria-', 0.8106724619865417	'Vienna', 0.7768976092338562	'German-Austria', 0.7641868591308594
cat	'cats', 0.8368930220603943	'housecat', 0.7675315737724304	'-cat', 0.7603166103363037

The results show that the pretrained language model indeed delivers meaningful cosine similarities since two animals like “cat” and “dog” (~ 0.75) or a city and a country like “Vienna” and “Austria” (~ 0.78) are considered more similar than an animal and a city like “cat” and “Vienna” (~ 0.17) what intuitively seems to make sense.

However, among the Top-3 most similar words there are also words like “-cat” or “Austria-” that only have a limited semantic meaning but naturally have a high similarity to the word itself and therefore are not incorrect. The other top similar words intuitively make sense since cities like “Salzburg” or “Prague” or even the country where the city lies in are considered semantically similar.

Part2: Short-Text Similarity

Method	Preprocessing	Pearson Correlation
Vector Space Model (from sklearn library)	Lower-casing + Stopword	0.7286070352739519
Average Word Embedding	Lower-casing + Stopword	0.6900093941796671
IDF Weighted Agg. Word Embedding	Lower-casing + Stopword	0.7005491041476277
Vector Space Model (from sklearn library)	Lower-casing	0.6913475696585767
Average Word Embedding	Lower-casing	0.6358813486570845
IDF Weighted Agg. Word Embedding	Lower-casing	0.6814384786182335

According to the Pearson Correlation the following “ranking” of the different methods emerges: 1. Vector Space Model, 2. IDF Weighted Agg. Word Embedding, 3. Average Word Embedding. As expected, the Pearson correlation increases throughout all test cases when removing stop words. Comparing the *Average Word Embedding* with the *IDF Weighted Agg. Word Embedding*, it is noticeable that mainly the former method is affected by this difference. Since stop words appear quiet often in text, they have a low IDF value and consequently get weighted with a low weight when using *IDF Weighted Agg. Word Embedding*. By underweighting stop words in this way, one can almost achieve the values obtained by preprocessing the stop words beforehand.

Part3: Training new language models

Word (of your choice)	**Top-1**	**Top-2**	**Top-3**
Obst	Gemüse (0.789)	Fleisch (0.719)	Eiweiß (0.687)
Universität	Hochschule (0.727)	Fakultät (0.697)	Uni (0.681)
Tisch	Herd (0.721)	Teller (0.72)	Balkon (0.716)

TODO: Analyze results briefly with a few words.

The most similar words to "Obst" all describe other food categories and the first match "Gemüse" fits quite well. The term "University" yielded also very intuitive results. However, "Tisch" did yield somehow similar things (all related to eating and cooking, which is common activity on a table), but "Balkon" is not something usually related to a table.

Training data set:

As training data, we used the provided Twitter data- set of April 2019 consisting of 858 MB of compressed german tweets which matches the recommended data size from the exercise description.

Optional Section

We perceived that when using only the sentence pairs instead of the whole corpus of sentences for inferring IDF weights that the *IDF Weighted Agg. Word Embedding* underperformed the *Average Word Embedding*. We suspect the reason for this is that with a merely small data set (i. e., only two sentences instead of all sentences) the IDF of stop words might not be small, thus eliminating the advantage of using IDF weights. The difference between the two methods then lies only in the use of (a poorly weighted) mean and median, where the median evidently provides better values.

The table below outlines the difference between using only the sentence pairs and using the whole corpus.

Method	**Preprocessing**	**Pearson Correlation (sentence pairs)**	**Pearson Correlation (whole corpus)**
IDF Weighted Agg. Word Embedding	Lower-casing + Stopword	0.6707124864722416	0.7005491041476277
IDF Weighted Agg. Word Embedding	Lower-casing	0.6128573782172263	0.6814384786182335

Note: Install instructions can be found in the projects "README.md".