**Documentation:**

Extracting synonyms and similar labels with word embeddings

# 25-12-2021

****

This documentation includes methods and results on finding similar words of a terminology using word embedding models. We also describe how to run the Python scripts and get similar results.

**Requirements**

* Python (libraries: genism, nltk)
* Embedding models
* Terminologies

**Intro**

In order to understand the method use, we first make an introduction to the technologies exploited. The main method used here is word embeddings. Word embeddings are essentially vector representations that we use in order to detect similar words. Each word is mapped to a vector, creating a huge word embedding multidimensional space. These vectors can either be pre-trained or trained on specific corpora. In our case we train our own fasttext embedding model on medical terminologies.

**Methodology**

Having trained our own word embedding model, we can now search for similar labels. We remind that the embedding model is trained on single tokens. Thus we get a vector for a single word. In order to get a vector for a multiword label, we get the average vector of the combined vectors, extracted by the words. We first compute all the vectors for the CEPIDC terms and afterwards for the ICD11 terms. Then for each CEPIDC term, we find the ICD11 terms with the nearest average vector.

**Files required**

* CEPIDC terms
* ICD11 terms
* big\_model.fr.bin (can be found in Google Drive: icd11/mapping/embedding\_model/big\_model\_fr.bin)

**Running the script**

All relative code can be found under the directory embeddings. We use the embeddings/most\_similar.py file. We first run the run\_cepidc\_map function. The specific method use multiprocessing in order to run faster. The method also requires a powerful machine with many cores available if possible.

**Results**

The results can found in CEPIDC\_embeddings\_top10\_nearest\_ICD11\_labels.xlsx. The first column includes the CEPIDC label and the next columns include the top 10 nearest ICD11 labels in descending similarity (or increasing distance).