NLP3 Project

LAB01

December 2022

---

nelson.vicel-farah
antoine.zellmeyer
karen.kaspar
romain1.brand
maxence.plantard

Epita
SCIA - Promo 2023

Table des matières

1 Keyword Extraction

1.1 Question 1.3 (2 points)

What are some of the limits of raw counts ? How could we improve the approach through preprocessing ?

— Limits : we sometimes find words that are not relevant as keywords
— How to improve : Adding more stop words in the preprocessing

1.2 Question 1.4 (2 points)

How can you find an optimal max_df ? Why are we using a sparse matrix instead of a regular matrix ?

In order to find the optimal max_df for a particular problem, we need to test and try. Thus, we need to take an arbitrary max_df at the start (0.9 or 0.95). We can then evaluate the performance. If said performance does not yield the results we wanted, we can try to lower the max_df until we get a satisfactory performance.

Since we're representing the frequency of words in a corpus of text, we will encounter a considerable amounts of zeroes. For this reason, representing the text as a sparse matrix instead of a regular matrix is more advantageous storage wise and computing time wise, as the sparse matrix doesn't assign memory to the zeroes and can hence take less memory.

1.3 Question 1.5 (2 points)

Find an example where there is a noticeable difference between tf-idf and raw counts ? Justify which method you would choose yourself (there is no bad and good answer here)

A possible example where there is a noticeable difference between tf-idf and raw count could be a corpus of tweets from Twitter.

One reason is that tweets often contain hashtags, which are words or phrases preceded by the "#" symbol. Hashtags are used to label the content of a tweet and make it more discoverable to other users. Hashtags can be very informative in the context of a tweet, but they might not be as relevant when analyzing a larger corpus of tweets. Using tf-idf can help down-weight hashtags that are common across many tweets and give higher weight to hashtags that are specific to individual tweets or that occur less

frequently in the corpus.

In this case, we would choose to use tf-idf because it takes into account the relative importance of each term in the dataset and down-weights common stop words, which could be less informative or useful for analysis. However, it's worth noting that the choice between using raw counts and tf-idf depends on the specific goals of the analysis and the characteristics of the dataset.

1.4 Question 2.2 (4 points)

Comparison of multiple techniques

1.4.1 Question 2.2.1

Draw a table of the solution, the quality score that you defined and the time taken to find keywords across a sample of 1000 of the original dataset.

The quality score measures the number of appearance of the top-n words in the title and abstract of the papers. We attribute more points for the word when it appears in the title than when it's in the abstract. The quality score is then normalized to account for a few missing abstracts.

	Method	Quality Score	Time
0	Counter	0.290010	0.282272
1	TF-IDF	0.279554	4.339000
2	KeyBERT	0.184049	177.699784

1.4.2 Question 2.2.2

Can you think of tweaks to reduce time to compute ? If yes, add an additional column to the above table with your proposed tweaks.

To reduce the computation time we could directly remove the stopwords from the text so the model doesn't have to trim them each time.

1.4.3 Question 2.2.3

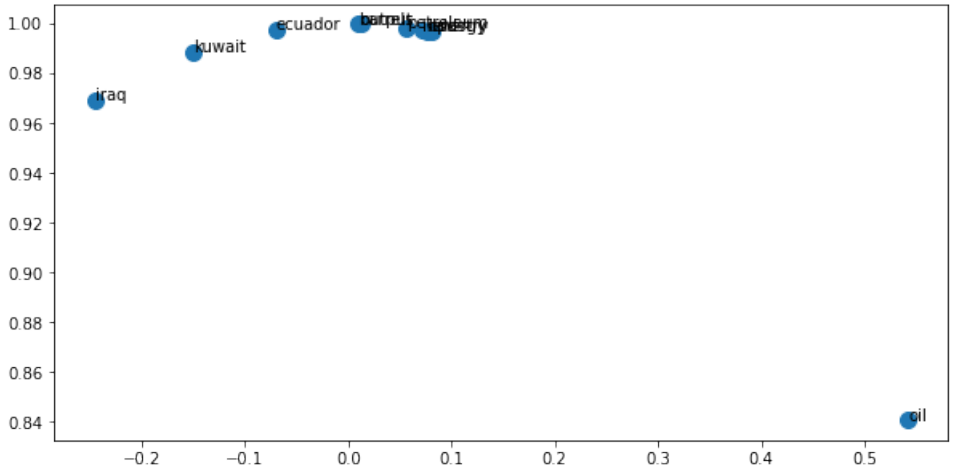
Based on the above table and lecture 1, what do you think is the most appropriate solution for keywords extraction ? Why ?

The best method depends on the specific needs of our application. Generally, a simple and fast solution would be the Counter method. A more sophisticated and effective approach would be Keybert or tf-idf. The best approach would be to try out multiple approaches and see which one works best for our needs. In our specific case, Counter seems to be the most appropriate solution for keywords extraction as it is the best in terms of time and quality performance according to our own measure of the quality score.

2 Word Vectors

2.1 Question 2.5 (3 points)

What clusters together in 2-dimensional embedding space ? What doesn't cluster together that you might think should have ? Note : "bpd" stands for "barrels per day" and is a commonly used abbreviation in crude oil topic articles.

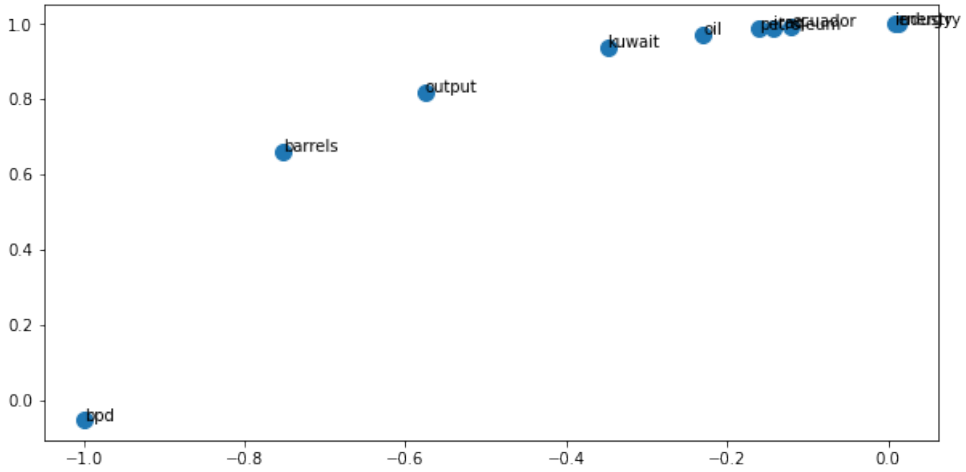


The words 'barrels', 'bpd', 'energy', 'industry', 'oil', 'output' and 'petroleum' cluster together in 2-dimensional embedding space. The word 'oil' and the word 'barrel' do not cluster together even though we might think they should have.

3 Prediction-Based Word Vectors

3.1 Question 3.1 (3 points)

What clusters together in 2-dimensional embedding space ? What doesn't cluster together that you think should have ? How is the plot different from the one generated earlier from the co-occurrence matrix ? What is a possible cause for the difference ?



The words 'energy' and 'industry' cluster together, while the words 'iraq', 'petromleum' and 'ecuador' cluster together. The words 'bpd' and 'barrels' are far from the rest of the clusters. The plot differs from the one generated earlier from the co-occurence matrix where all non-country words are overlapping. One possible cause of the difference is due to the difference in representation between GloVe and SVD.

3.2 Question 3.2 (1.5 points)

Find a word with at least two different meanings such that the top-10 most similar words (according to cosine similarity) contain related words from both meanings.

Please state the word you discover and the multiple meanings that occur in the top 10. Why do you think many of the polysemous or homonymic words you tried didn't work (i.e. the top-10 most similar words only contain one of the meanings of the words) ?

The word we discovered that has two different meanings in the top-10 most similar words is 'left'. One of the meanings appearing in the top-10 is the direction (i.e. as the opposite direction to right) and the second meaning is leaving.

Many of the polysemous or homonymic words we tried didn't work because one of the meanings of the words was used considerably more often than the other.

3.3 Question 3.3 (2 points)

When considering Cosine Similarity, it's often more convenient to think of Cosine Distance, which is simply 1 - Cosine Similarity.

Find three words (w1,w2,w3) where w1 and w2 are synonyms and w1 and w3 are antonyms, but Cosine Distance (w1,w3)< Cosine Distance (w1,w2) .

As an example, w1 ="happy" is closer to w3 ="sad" than to w2 ="cheerful". Please find a different example that satisfies the above. Once you have found your example, please give a possible explanation for why this counter-intuitive result may have happened.

The words we picked are :
w1 = "hot"
w2 = "boiling"
w3 = "cold"

The cosine distance between w1 and w3 is 0.41 and the cosine distance between w1 and w2 is 0.49. Therefore the cosine distance between the word 'hot' and its antonym 'cold' is shorter than the cosine distance between 'hot' and its synonym 'boiling'.

This counter-intuitive result may have happened because the word boiling isn't used as often and in the same context as hot, whereas cold is often used in the same context and sentences as hot.

3.4 Question 3.4 (1.5 points)

Let m , k , w , and x denote the word vectors for man, king, woman, and the answer, respectively. Using only vectors m , k , w , and the vector arithmetic operators + and - in your answer, what is the expression in which we are maximizing cosine similarity with x ?

The expression in which we are maximizing cosine similarity with x is

$$cos\_sim(k + w - m, x)$$

3.5 Question 3.5 (1.5 points)

Find an example of analogy that holds according to these vectors (i.e. the intended word is ranked top). In your solution please state the full analogy in the form x :y : : a :b.

An analogy that holds according to these vectors is : banana : yellow : : strawberry : red, where the y and b indicate the respective colors of x and a.

3.6 Question 3.6 (1.5 points)

Find an example of analogy that does not hold according to these vectors. In your solution, state the intended analogy in the form x :y : : a :b, and state the (incorrect) value of b according to the word vectors.

librarian :books : : florist :flowers

The incorrect value of b according to the word vectors is 'handmade'.

3.7 Question 3.7 (1 point)

Point out the difference between the list of female-associated words and the list of male-associated words, and explain how it is reflecting gender bias.

The difference between the list of female-associated words and the list of male-associated words :

The female list includes words such as : nurse, pregnant, mother, employer, teacher, child, homemaker, nurses.

The male list includes words such as : working, laborer, unemployed, job, work, mechanic, worked factory.

In the case of the female-associated list, most words include jobs stereotypically associated with women and womanhood such as caring job and stay at home jobs whereas the male-associated list includes words related to hard work and physical labor, job often associated with men. The list therefore showcases stereotypes and gender bias.

3.8 Question 3.8 (1 point)

Use the most_similar function to find another case where some bias is exhibited by the vectors. Please briefly explain the example of bias that you

discover.

We used the most_similar function to get a list of words associated with elderly and youthful.

The list associated with the word elderly contains words such as : infirm, homeless, needy, patients and sick. The list associated with the word youthful includes : enthusiasm, bravado, ambition, energetic and idealism. Both lists clearly showcase an example of age bias.

3.9 Question 3.9 (2 points)

Give one explanation of how bias gets into the word vectors. What is an experiment that you could do to test for or to measure this source of bias ?

One explanation of how bias gets into the word vectors could be that word embedding models learn the pattern they found in the data, which means that if the training corpus reproduce these biases on multiple of its texts, the model will restore these biases as well.

One experiment we could do to test for or measure bias in learning data for word embedding is to measure the association between a pair of target words and a pair of attribute words. For example, we could measure the cosine similarity between the target words that can indicate bias such as "man" and "woman" with the attribute words "work" and "home". If the model exhibits bias, we would expect to see a stronger similarity between the target words and the attribute words that are stereotypically associated with those targets.

4 Prediction-Based Sentence Vectors

4.1 Question 4.1 (3 points)

How would you represent a sentence with Glove ? What are the limits of your proposed implementation ?

One way to represent the sentence as a whole would be to take the average of the word vectors for each word in the sentence. This would give us a single vector that represents the overall meaning of the sentence.

There are a few limitations to this approach. One limitation is that it does not take into account the order of the words in the sentence, so the resulting vector may

not accurately capture the meaning of the sentence if the order of the words is important. Additionally, this approach does not capture relationships between words in the sentence, such as dependencies or syntactic structure.

4.2 Question 4.2 (3 points)

Evaluate clustering quality of SentenceBERT. What makes it good at clustering sentences ? Which method of the two below would you go for ?

The different clusters we found with SentenceBERT are relevant with the meaning of the sentences. The clustering quality of SentenceBERT when tested on a short corpus with a correct number of clusters seems appropriate.

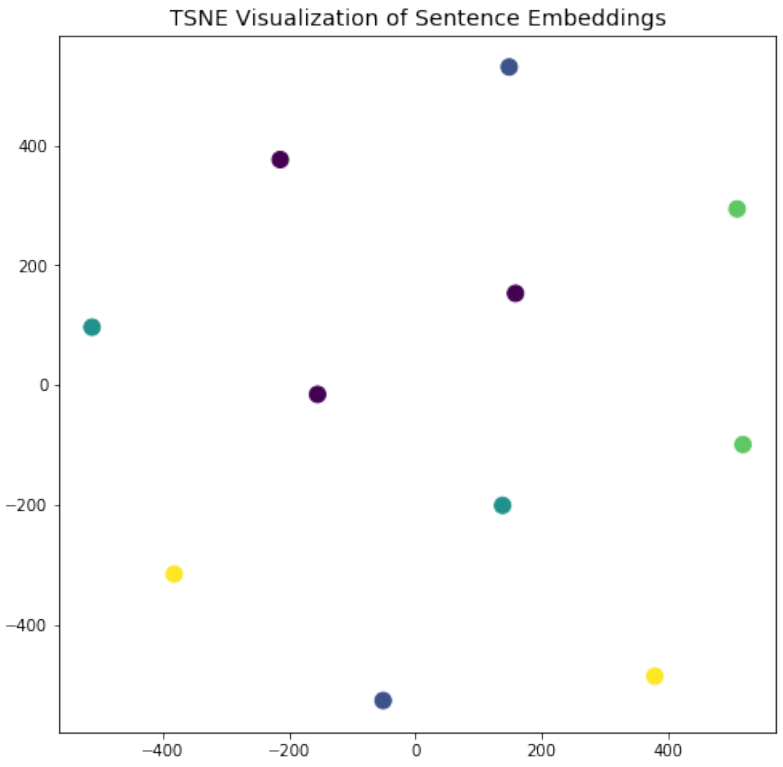
One of the main features that makes SentenceBERT good at clustering sentences is its ability to understand and capture the semantic meaning of a sentence. SentenceBERT is trained on a large dataset of sentence-level pairs, and it uses this training data to learn how to encode the meaning of a sentence into a fixed-length vector representation. These sentence embeddings capture the contextual information and meaning of the input sentence, and they can be used to compare and cluster similar sentences.

In general, SentenceBERT may be more effective for generating sentence embedding because it was specifically designed for that purpose. However, GLoVE may still be a good choice in certain situations, such as when you have limited resources.

4.3 Question 4.3 (3 points)

Plot the above corpus with your favorite method in a 2-dimensional space. Comment on the output.

The original dimensional space of the embeddings is of size 768. We project these vector using TSNE from the sckit-learn library which is a tool to visualize high-dimensional data. The result can be seen below.



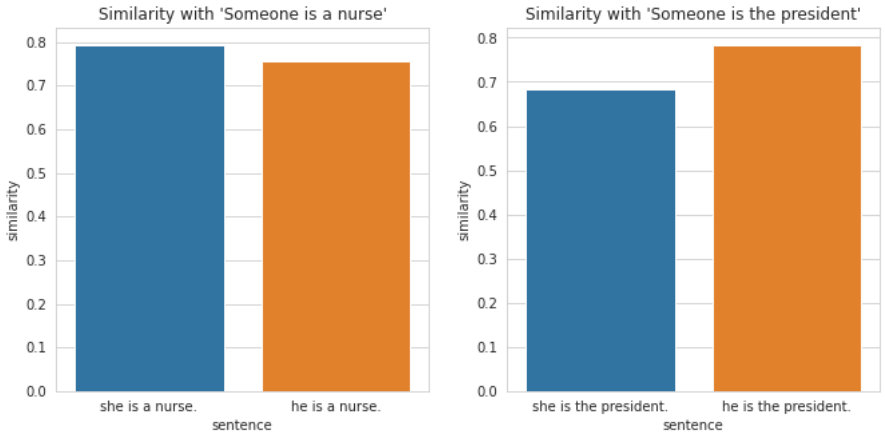
All the dots representing the sentences in the 2-dimensional space seem to be spaced by the same distance. However, the dots that have the same colors, and that therefore belong to the same cluster, aren't closer in the 2D space. The 2-dimensional space doesn't convey a good representation of the similarity and difference between sentences.

4.4 Question 4.4 (4 points)

Select a corpus of interest, or examples of interest and shed light on one source of bias from SentenceBERT.

One source of bias we could observe in such model might be the gender biases we mentioned previously.

To highlight this kind of bias from SentenceBERT, we can observe the distance between gendered sentences and gender neutral sentences. If the vector of a gender neutral sentence tends towards one of its gendered equivalent, we can consider that there is indeed a gender bias in the sentence embeddings.



In this example, we show that *"Someone is a nurse"* is slightly closer to its female equivalent than the male one. We also find a greater similarity for *"Someone is the president"* towards the male counterpart, which shows an even bigger bias.