Similarity Index of two text documents

The following two main resources used for finding similarity index of two text documents:

1.Tensorflow:

- TensorFlow is a free and open source software python library for fast numerical computing. It is created and released by Google.
- It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

2.Google's universal Sentence encoder:

- The Universal Sentence Encoder encodes text into high-dimensional vectors that can be used for text classification, semantic similarity, clustering and other natural language tasks.The pre-trained Universal Sentence Encoder is publicly available in Tensorflow-hub.
- This encoder is a better tool than tf-idf.It provides most accurate results.
- GloVe, word2vec, fastText, etc these embeddings are only useful for word level operations, sometimes we would want to explore embeddings for sentences, or generally, greater-than-word length text, universal sentence encoder is best suited for it.
- It is trained on a combination of unsupervised data (in a skip-thought-like task) and supervised data (the SNLI corpus).

Following are steps:

1. Import tensorflow libraries,include google's universal encoder using the latest embedder URL available.Load the embedder using Tensorflow Hub.

2. I have used google colab as it supports many runtime libraries. Next load the documents to be tested from the local machine.

3. Create the placeholders for the documents. A placeholder is simply a variable that we will assign data to at a later date. It allows us to create our operations and build our computation graph, without needing the data. In TensorFlow terminology, we then feed data into the graph through these placeholders.

4. Placeholders return a tensor(multi-dimensional arrays with a uniform type (called a dtype) tensors are like np.arrays) that may be used as a handle for feeding a value, but not evaluated directly.

5. Create the message encodings.The tensors are sent to universal encoders which converts the messages into a 521 size vector. It is known as message embedding.

6. Create the session using tensorflow. Send the documents one by one to encoder using placeholders and apply embedding..

7. The universal encoder module does not require preprocessing the data before applying the module, it performs best effort text input preprocessing inside the graph. It is already trained on a combination of unsupervised data (in a skip-thought-like task) and supervised data (the SNLI corpus).

8. The output will be two vectors of 512 size each for every document.These have been represented in the output.

9. Similarity index between these vectors is calculated using the inner product method. This inner product is nothing but a sum product over the last axes.

10. The output is represented using the heatmap,also known as correlation matrix.The higher the similarity index more are the chances of the documents being the same.