**Measuring Text Similarity: TF-IDF and Cosine Similarity**

One established method for computing the similarity between two bodies of texts is to measure their cosine similarity. In order to measure the cosine similarity between two bodies of text, both bodies of text must first be represented in a quantitative form. By using a TF-IDF analysis, the bodies of text are vectorized so that their cosine similarity can be calculated.

**TF-IDF**

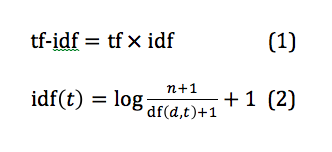
TF-IDF, short for Text Frequency – Inverse Document Frequency, is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. TF-IDF originates as a method for document search and information retrieval. This value of statistical significance is obtained by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

The TF-IDF weighting of a term in a collection of documents increases proportionally to the number of times a term appears in a document. However, this value is offset by the number of documents that contain that specific term. This is so that common words that appear across multiple documents, such as “this”, “what”, and “if”, are ranked lower despite their high frequency of appearance. Likewise, terms that are unique and appear sparsely in the collection of documents are weighted as more relevant.

The term frequency (TF) of a term in a document refers to the number of occurrences of that term inside a document. The simplest calculation of term frequency is a raw numerical count of the instances of that term. This frequency can be adjusted several different ways, one of which is by accounting for the length of a document.

The inverse document frequency (IDF) of a term in a document refers to the rarity of a word in the entire collection of documents. This is calculated by taking the total number of documents, dividing it by the number of documents that contain the term, calculating the logarithm of that value, and adding 1.

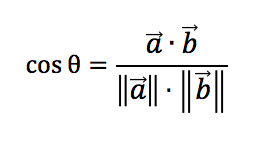
The TF-IDF score for a term is calculated by multiplying these two values (TF and IDF). The higher the score, the more relevant that term is in a particular document. The equations below are what is used in our calculations to calculate the TF-IDF.



\*In our calculations we add the 1 to prevent division by 0 and negative values, which are harder to interpret. Adding the 1 to the final logarithmic value prevents terms that appear in all documents to be given a 0 IDF score. This prevents a 0 score for a term that appears in all documents.

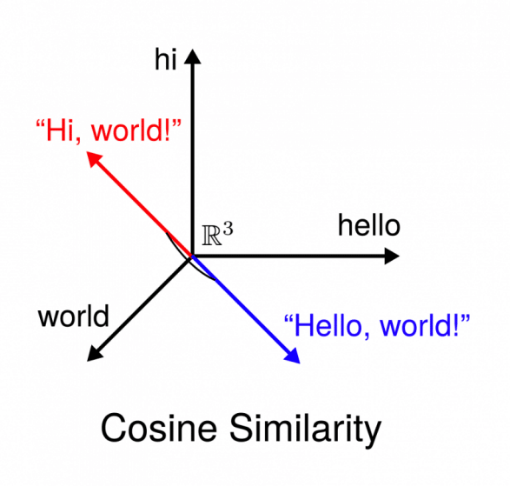
**Cosine Similarity**

With a TF-IDF analysis, we can determine how relevant a term is to a document. However, to determine the similarity between documents, we need additional metrics and a way to compare the similarity of the terms between documents as a whole. Cosine similarity is a metric used to measure how similar documents are by measuring the cosine of the angle between two vectors. In our case, the two vectors represent two bodies of text. Cosine similarity is more accurate than other methods of determining similarity (i.e. counting the number of similar terms) because even if two similar documents are far apart due to term frequency or length of text, calculating the cosine of the angle between the two will still give an accurate representation of similarity.



The formula above represents how the cosine similarity of the two vectors is calculated, where *a* and *b* are the vectors of the bodies of text. The cosine of the angle is the dot product of the two vectors divided by the product of their Euclidean norms. As the value of θ increases, the value of cos θ decreases. Two texts that are identical will have a cos θ value of 1. Two texts that are completely different will have a cos θ of 0. Thus, larger cos θ values will indicate that two texts are more similar.

When vectors are plotted on a multi-dimensional space, the cosine of the angle between vectors determines how far apart they are in similarity. The diagram below shows a graphical example of two sentences. There are a total of 3 terms across this collection of documents: “hello”, “hi”, and “world”.



**Examples**

Let’s take a look at some examples. We want to find out how similar these two sentences are:

1. Elise ate a salad for lunch on Friday.
2. John skipped lunch and went for a walk instead.

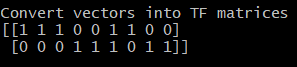
The first step in our process is to determine how significant each term is in the collection of documents. In this example, there are only two documents. Before we start vectorizing the terms through a TF-IDF analysis, we do some preprocessing on our documents.

In our preprocessing step we do several things. First, we remove punctuation from the sentences. Second, we remove stop-words from the sentences. Stop-words are commonly used words that provide no additional meaning to a sentence, such as the word “the”. Thirdly, we lemmatize all of the words in the sentences. Lemmatization is the process of converting a word into its base form. This is a necessary step as verbs and other words have multiple forms, yet their meaning will remain the same. For example, the terms ‘running’ and ‘ran’ would be treated as the same term for our purposes of text similarity.

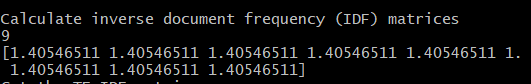
After our preprocessing, our indexed list of terms should look like this. The terms are indexed by alphabetical order. These terms will each be one element in our vector.



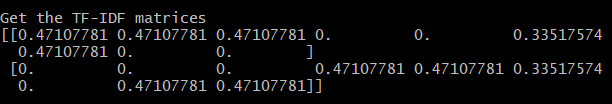
Our next step is to convert these word tokens into our TF matrix. Note that this is a 2x9 matrix. The number of rows in the TF matrix corresponds to the number of documents, while the number of columns corresponds to the number of tokens (words). Therefore row 1 represents sentence 1, and row 2 represents sentence 2. Each vector element corresponds to the number of the occurrences of the term for a particular document.



Now that we have calculated the TF, we can now calculate the IDF for each term. Calculating the IDF for each term in the matrix above gives us the following IDF matrix below.



The next step after computing both the TF and IDF matrices is to compute the TF-IDF matrix. This is done by multiplying the TF matrix by the diagonal IDF matrix, and then dividing the TF-IDF by the Euclidean norm. The final result is a 2x8 matrix with each vector element corresponding to the TF-IDF of a term.



Now that we have completed the TF-IDF analysis and vectorized the terms in the sentences, we can now use cosine similarity to compute the similarity between the two sentences. The similarity matrix can be found by multiplying the matrix obtained in the last step by its transpose. Doing so will give us the following result.

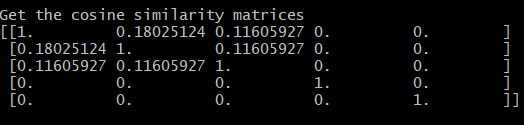


The final similarity matrix can be read by row or by column. An element of value 1 indicates that the element represents the principal review, i.e. the review that is being compared to the other reviews in the row or column. In the case of identical documents being compared, there may be several element values of 1. From our similarity matrix, we can see that the two sentences have a cosine similarity value of 0.11234728. However, on its own, this value is not so useful. Let’s take a look at an example with several documents.

Consider the following body of documents:

1. John and Alex decided that because it was a sunny day, they would go out and play some tennis.
2. Luis and Michael felt like playing tennis today.
3. Tennis is a sport that is loved by many.
4. How many licks does it take to get to the center of a tootsie pop?
5. Listen to this new song by Coldplay, it rocks!

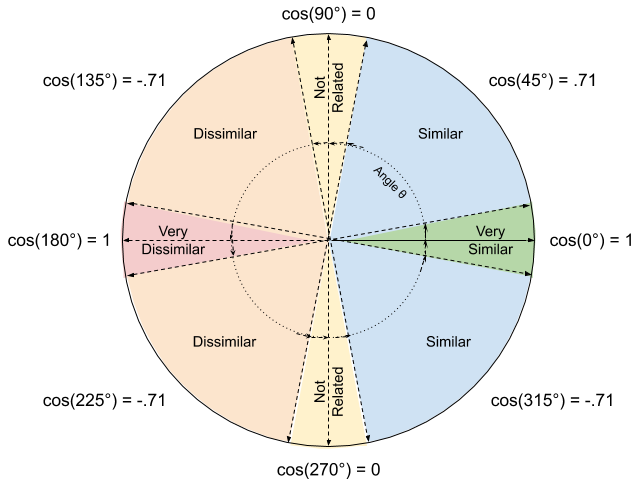
The final similarity matrix is as follows:



In this matrix, we can see a more apparent difference in similarity between the texts. If we take an initial observation of our documents, we would assume that sentence 1 and sentence 2 are closely related, sentence 1 and sentence 3 are slightly related, and sentence 4 and sentences 5 are not related to sentence 1 at all. The matrix corroborates our assumption. Sentence 1 and sentence 2 have a cosine similarity score of 0.18025124, the highest in the matrix. With the second highest similarity, Sentence 1 and sentence 3 have a similarity score of 0.11605927. Sentence 4 and sentence 5 are not related to sentence 1 at all, so we get a similarity score of 0 for both.

**Using the Cosine Similarity Values**

While cosine similarity values can help us tell how *relatively* similar documents are, they cannot give us an *absolute* similarity conclusion. We cannot determine that a document *A* is similar to a document *B*, but we can determine that document *A* is more similar to document *B* than a document *C*. So how can we use these values?



While our data does not work with negative cosine values, the above is a helpful diagram on how we can interpret cosine similarity scores. Cosine similarity values are heavily dependent on the vectors that are used to form the final resulting value. There are no standardized values for cosine similarity, or a set expected range that similar documents will fall in.

Because interpretation of cosine similarity is heavily dependent on the context of the data, it is necessary to create our own interpretation. One method researchers use is to calculate the cosine similarity across all of their data and determine an appropriate cutoff range based on the distribution. Similarity scores within the range are identified as the target group of the data. Calculating the average across a group of similarity scores is also another way of interpreting similarity. In all, interpreting the cosine similarity depends largely on the goal of the task at hand.

**Sources**

**Articles**

<https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

<https://monkeylearn.com/blog/what-is-tf-idf/>

<https://www.machinelearningplus.com/nlp/cosine-similarity/>

<https://sites.temple.edu/tudsc/2017/03/30/measuring-similarity-between-texts-in-python/>

<https://www.ml-science.com/cosine-similarity>

**Academic Papers**

<https://opencommons.uconn.edu/cgi/viewcontent.cgi?article=1009&context=nera-2018>