TF-IDF

- TF-IDF stands for “Term Frequency — Inverse Document Frequency”.
- This is a technique to quantify a word in documents, we generally compute a weight to each word which signifies the importance of the word in the document and corpus.
- The computer can understand any data only in the form of numerical value. So, for this reason we vectorise all of the text so that the computer can understand the text better.
- By vectorizing the documents we can further perform multiple tasks such as finding the relevant documents, ranking, clustering and so on. This is the same thing that happens when you perform a google search. The web pages are called documents and the search text with which you search is called a query. google maintains a fixed representation for all of the documents. When you search with a query, google will find the relevance of the query with all of the documents, ranks them in the order of relevance and shows you the top k documents, all of this process is done using the vectorised form of query and documents.

TF-IDF = Term Frequency (TF) * Inverse Document Frequency (IDF)

Terminology : t — term (word), **d** — document (set of words), **N** — count of corpus and **corpus** — the total document set

Term Frequency

- This measures the frequency of a word in a document.
- This highly depends on the length of the document and the generality of word, for example a very common word such as “was” can appear multiple times in a document.
- But if we take two documents one which have 100 words and other which have 10,000 words. There is a high probability that the common word such as “was” can be present more in the 10,000 worded document.
- But we cannot say that the longer document is more important than the shorter document. For this exact reason, we perform a normalisation on the frequency value.
- We divide the the frequency with the total number of words in the document.

Recall that we need to finally vectorize the document, when we are planning to vectorize the documents, we cannot just consider the words that are present in that particular document. If we do that, then the vector length will be different for both the documents, and it will not be feasible to compute the similarity. So, what we do is that we vectorize the documents on the vocab. vocab is the list of all possible words in the corpus.

When we are vectorizing the documents, we check for each words count. In worst case if the term doesn’t exist in the document, then that particular TF value will be 0 and in other extreme case, if all the words in the document are same, then it will be 1. The final value of the normalised TF value will be in the range of [0 to 1]. 0, 1 inclusive.

TF is individual to each document and word, hence we can formulate TF as follows.

tf(t,d) = count of t in d / number of words in d

If we already computed the TF value and if this produces a vectorized form of the document, why not use just TF to find the relevance between documents? why do we need IDF?

Let me explain, though we calculated the TF value, still there are few problems, for example, words which are the most common words such as “is, are” will have very high values, giving those words a very high importance. But using these words to compute the relevance produces bad results. These kind of common words are called stop-words, although we will remove the stop words later in the preprocessing step, finding the importance of the word across all the documents and normalizing using that value represents the documents much better.

Document Frequency

- This measures the importance of document in whole set of corpus, this is very similar to TF.
- The only difference is that TF is frequency counter for a term t in document d, where as DF is the count of occurrences of term t in the document set N.
- In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = \text{occurrence of } t \text{ in documents}$$

To keep this also in a range, we normalize by dividing with the total number of documents.

Our main goal is to know the informativeness of a term, and DF is the exact inverse of it, that is why we inverse the DF

Inverse Document Frequency

- IDF is the inverse of the document frequency which measures the informativeness of term t.
- When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as “is” is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

- Now there are few other problems with the IDF, in case of a large corpus, say 10,000, the IDF value explodes. So to dampen the effect we take log of IDF.
- During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = \log(N/(df + 1))$$

- Finally, by taking a multiplicative value of TF and IDF, we get the TF-IDF score, there are many different variations of TF-IDF but for now let us concentrate on the this basic version.

$$tf\text{-}idf(t, d) = tf(t, d) * \log(N/(df + 1))$$

How to Compute:

- Typically, the **tf-idf weight is composed by two terms:** the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.
    1. **TF:** Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

       $$TF(t) = (\text{Number of times term } t \text{ appears in a document}) / (\text{Total number of terms in the document})$$

    2. **IDF:** Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

       $$IDF(t) = log\_e(\text{Total number of documents} / \text{Number of documents with term } t \text{ in it})$$

Example:

- Consider a document containing 100 words wherein the word *cat* appears 3 times.
- The term frequency (i.e., tf) for *cat* is then (3 / 100) = 0.03.
- Now, assume we have 10 million documents and the word *cat* appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4.
- Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.