# Cosine similarity

Thursday, November 22, 2018 1:55 PM

https://blog.exploratory.io/demystifying-text-analytics-finding-similar-documents-with-cosine-similarity-e7b9e5b8e515

# Jaccard similarity

*Jaccard similarity* is a simple but intuitive measure of similarity between two sets.

$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$J(doc1,doc2)=doc1∪doc2doc1∩doc2

For documents we measure it as proportion of number of common words to number of unique words in both documets. In the field of NLP *jaccard similarity* can be particularly useful for duplicates detection. *text2vec* however provides generic efficient realization which can be used in many other applications.

For calculation of *jaccard similarity* between 2 sets of documents user have to provide DTM for each them (DTMs should be in the same vector space!):

From <http://text2vec.org/similarity.html>

# Cosine similarity

Classical approach from computational linguistics is to measure similarity based on the content overlap between documents. For this we will represent documents as bag-of-words, so each document will be a sparse vector. And define measure of overlap as angle between vectors:

$similarity(doc_1, doc_2) = cos(\theta) = \frac{doc_1 doc_2}{|doc_1||doc_2|}$similarity(doc1,doc2)=cos(θ)=doc1doc2/(doc1||doc2|)

By *cosine distance/dissimilarity* we assume following:

$distance(doc_1, doc_2) = 1 - similarity(doc_1, doc_2)$distance(doc1,doc2)=1−similarity(doc1,doc2)

It is important to note, however, that this is not a proper distance metric in a mathematical sense as it does not have the triangle inequality property and it violates the coincidence axiom.

Calculation of cosine similarity is similar to jaccard similarity:

From <http://text2vec.org/similarity.html>