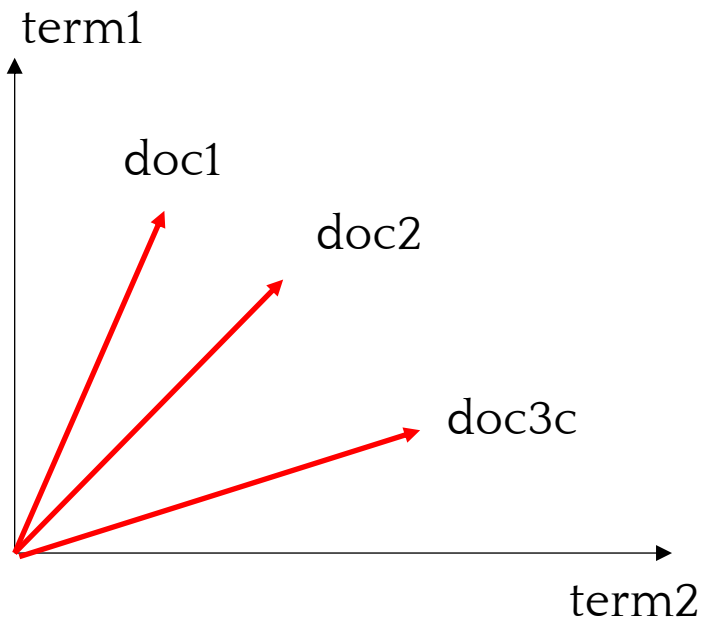
Document Similarity

Ch.4 Text Analytics with Python. Dipanjan Sankar. 2019. Apress
Sanket Gupta. 2018. Overview of Text Similarity Metrics in Python

Document Similarity Metrics



space: {t1, t2, t3...tN}

document 1 → $doc_1 = (x_{1,1}, x_{1,2})$

document 2 → $doc_2 = (x_{2,1}, x_{2,2})$

document 3 → $doc_3 = (x_{3,1}, x_{3,2})$

doc_i - a vector representing i document

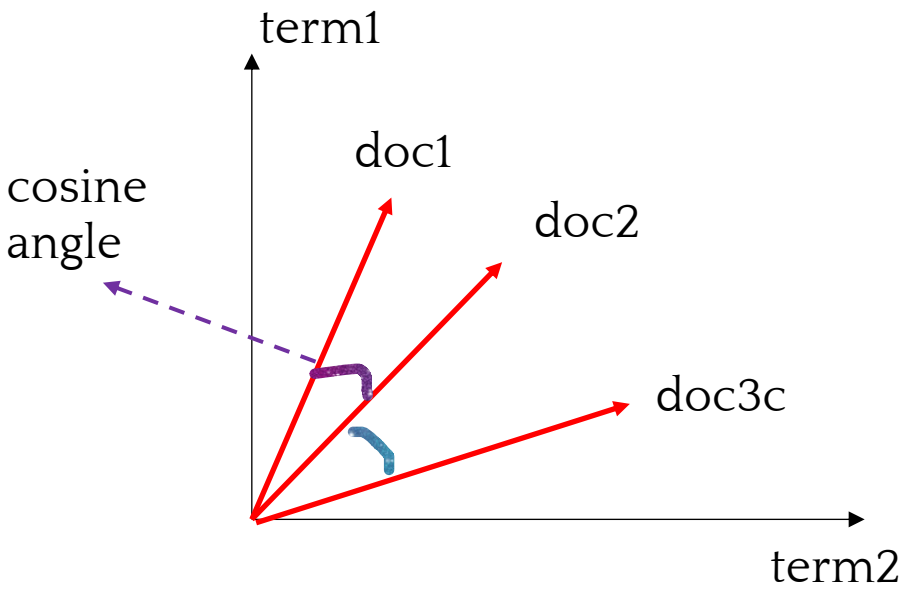
$x_{i,j}$ - the value of occurrence of j index term in i document

A distance or similarity based metric is used to identify how similar are text documents

Similarity Metrics Jaccard Similarity, Cosine Similarity, Euclidean Distance ...

Similarity metrics calculate scores (distance between two objects).

Cosine Similarity



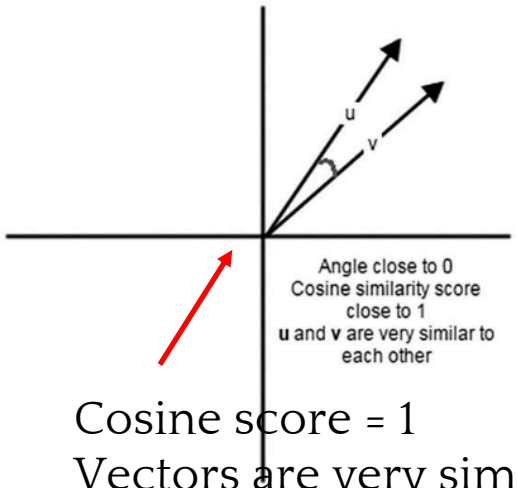
vector1 vector2 components of vectors

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

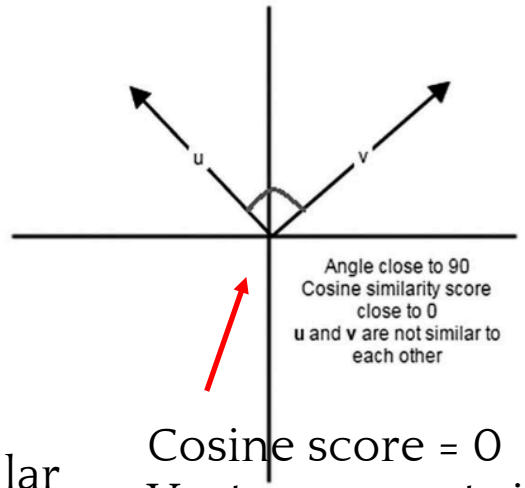
Euclidean Distance

x=(5,0,3,0,2,0,0,2,0,0)
y=(3,0,2,0,1,1,0,1,0,1)

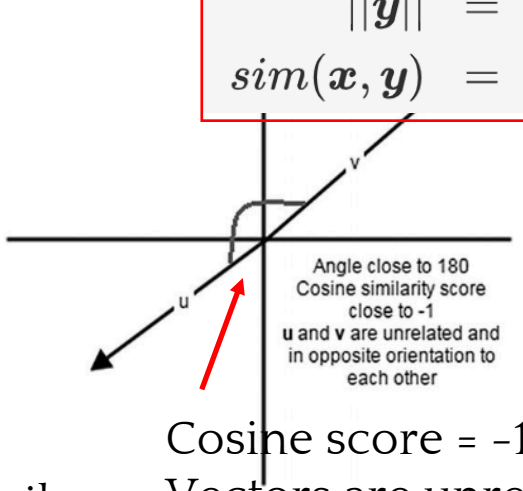
$$\begin{aligned} \boldsymbol{x}^t \cdot \boldsymbol{y} &= 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times \\ &\quad +0 \times 0 + 0 \times 1 = 25 \\ ||\boldsymbol{x}|| &= \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0} \\ ||\boldsymbol{y}|| &= \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0} \\ sim(\boldsymbol{x}, \boldsymbol{y}) &= 0.94 \end{aligned}$$



Cosine score = 1
Vectors are very similar



Cosine score = 0
Vectors are not similar



Cosine score = -1
Vectors are unrelated

Cosine Similarity

Corpus

Document 1: AI is our friend and it has been friendly
Document 2: AI and humans have always been friendly

corpus =['AI is our friend and it has been friendly',
'AI and humans have always been friendly']

Step 1. Get TF

	ai	friend	friendly	humans
0	1	1	1	0
1	1	0	1	1

```
vectorizer =
CountVectorizer(stop_words='english')
t = vectorizer.fit_transform(corpus)
modelt = t.toarray()
```

Step 2. Get Cosine Similarity

	Doc1	Doc2
Doc1	1.000000	0.666667
Doc2	0.666667	1.000000

```
from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity(model)
d = pd.DataFrame(get_cosine_sim(corpus), index=['Doc1','Doc2'])
d.columns=['Doc1','Doc2']
```

Cosine Similarity Score = 0.66

(Sanket Gupta, 2018. Overview for Text Similarity Metrics in Python)

Jaccard Similarity

Document 1: AI is our friend and it has been friendly
Document 2: AI and humans have always been friendly

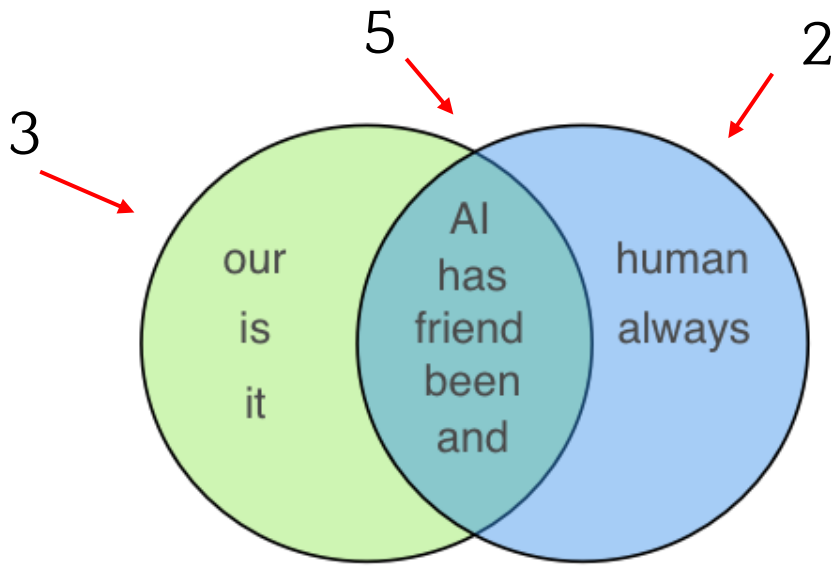
The size of intersection divided by size of union of two sets

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Intersection

Union

$$s_{ij} = \frac{p}{p+q+r}$$



p = N attributes in both sets
q = N attributes in set i
r = N attributes in set j

5/(5+3+2) = 0.5

100% - all members are shared
0% - no shared members

Lemmatization is preferred to reduce words to their root word

Difference between Jaccard and Cosine Similarities

Jaccard

Takes unique set of words for each sentence / document

Will not change if a word is repeated

Use in cases where duplication does not matter

Cosine

Takes total length of the vectors (from bag of words term frequency or tf-idf)

Will change if a word is repeated

Use in cases where duplication matters