

# AI Lab 6: Document Similarity

## Introduction:

In this lab, we will be taking four sentences as a document. We then calculate how much these documents are similar to each other.

## Procedure:

We tokenize the documents and remove stop words. Then we obtain a list of words i.e terms in the documents. We use two methods to calculate similarity in this lab. They are:

a) Jaccard similarity:

In jaccard similarity, we don't consider the frequency of repeating terms in a document. It is given by:

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

where, the numerator denotes the number of terms common in both documents A and document B. The denominator is the union of documents A and B which contains all the terms in either A or B.

b) Cosine Similarity:

In this technique( tf ), we calculate the term frequency and document frequency of each terms. The term frequency is the frequency of terms in a document

Document frequency(  $df_i$  ) is the number of documents in the corpus that contains the term. Its inverse is known as inverse-document frequency and is given by:

$$\begin{aligned}idf_i &= \text{inverse document frequency of term } i, \\ &= \log_2 (N / df_i )\end{aligned}$$

Cosine similarity is calculated as:

$$\text{CosSim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2 \cdot \sum_{i=1}^t w_{iq}^2}}$$

Here  $W_{ij}$  is weight obtained as  $tf_{ij} * idf_i$  for the term  $i$  in document  $j$

## Implementation:

This lab was implemented in Python 3.5.2.

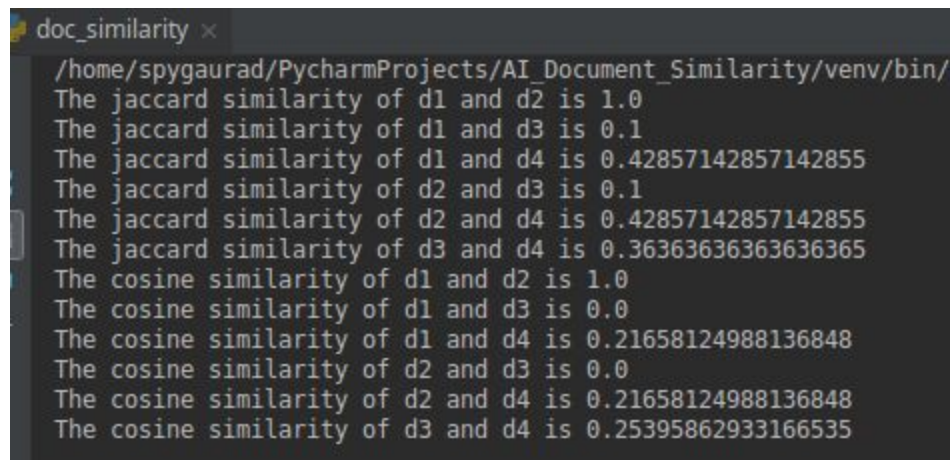
The documents taken as sample:

```
d1 = "I am Sam."  
d2 = "Sam I am."  
d3 = "I do not like green eggs and ham."  
d4 = "I do not like them, Sam I am."
```

Package nltk was used to tokenize the texts and remove stop words:

```
d1 = nltk.word_tokenize((re.sub(r"\W", " ", d1)).lower())
```

## Observations



The screenshot shows a terminal window titled 'doc\_similarity' with the following output:

```
/home/spygaurad/PycharmProjects/AI_Document_Similarity/venv/bin/  
The jaccard similarity of d1 and d2 is 1.0  
The jaccard similarity of d1 and d3 is 0.1  
The jaccard similarity of d1 and d4 is 0.42857142857142855  
The jaccard similarity of d2 and d3 is 0.1  
The jaccard similarity of d2 and d4 is 0.42857142857142855  
The jaccard similarity of d3 and d4 is 0.36363636363636365  
The cosine similarity of d1 and d2 is 1.0  
The cosine similarity of d1 and d3 is 0.0  
The cosine similarity of d1 and d4 is 0.21658124988136848  
The cosine similarity of d2 and d3 is 0.0  
The cosine similarity of d2 and d4 is 0.21658124988136848  
The cosine similarity of d3 and d4 is 0.25395862933166535
```

## Conclusion

Using Jaccard similarity, due to the ignorance of frequency of words in a document and the concept that terms that repeat in many documents are not much significant, the similarity calculated by this method is not optimal. Using cosine similarity we can obtain the desired output.

**Source Code :** [https://github.com/spygaurad/Document\\_Similarity/blob/master/doc\\_similarity.py](https://github.com/spygaurad/Document_Similarity/blob/master/doc_similarity.py)