Introduction to
Natural Language Processing

NLP

Imagine you work for Google News and you want to group news articles by topic

Or you work for a legal firm and you need to sift through thousands of pages of legal documents to find relevant ones.

This is where NLP can help!

NLP

We will want to:

- Compile Documents
- Featurize Them
- Compare their features

NLP

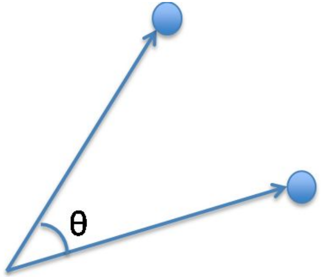
Simple Example:

- You have 2 documents:
  - “Blue House”
  - “Red House”
- Featurize based on word count:
  - “Blue House” -> (red,blue,house) -> (0,1,1)
  - “Red House” -> (red,blue,house) -> (1,0,1)

NLP

- A document represented as a vector of word counts is called a “Bag of Words”
  - “Blue House” -> (red,blue,house) -> (0,1,1)
  - “Red House” -> (red,blue,house) -> (1,0,1)
- You can use cosine similarity on the vectors made to determine similarity:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



NLP

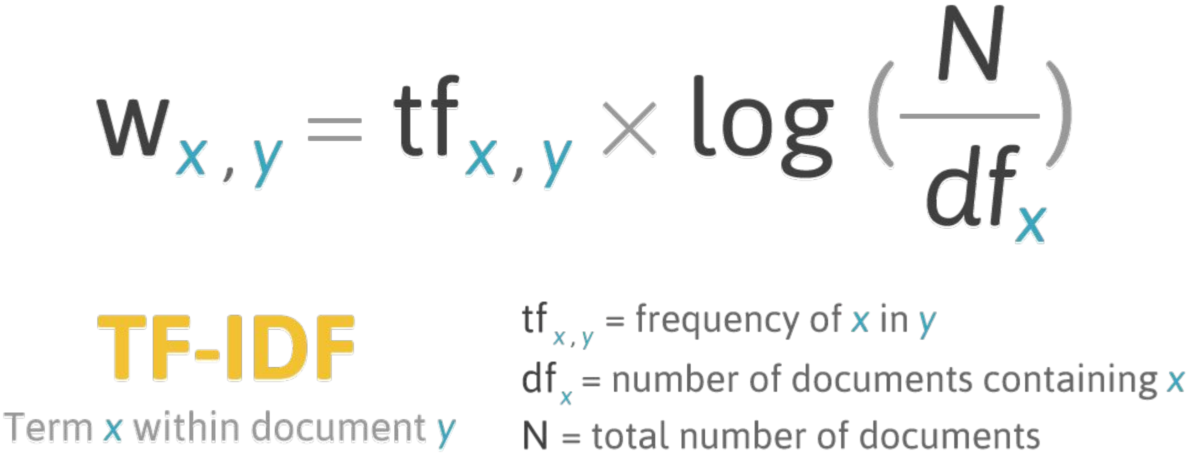
- We can improve on Bag of Words by adjusting word counts based on their frequency in corpus (the group of all the documents)
- We can use TF-IDF (Term Frequency - Inverse Document Frequency)

NLP

- Term Frequency - Importance of the term within that document
    - TF(d,t) = Number of occurrences of term t in document d
- Inverse Document Frequency - Importance of the term in the corpus
    - IDF(t) = log(D/t) where
        - D = total number of documents
        - t = number of documents with the term

NLP

- Mathematically, TF-IDF is then expressed:



$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term *x* within document *y*

$tf_{x,y}$ = frequency of *x* in *y*

df_x = number of documents containing *x*

N = total number of documents