Text Analysis and Visualization with Python

Arieda Muço

Central European University

Simple Example

You have 2 documents:

1. Blue House
2. Red House

Our corpus will consist of all the words in the documents namely: Red, Blue, House. The vector representation in the “Bag of Words" approach:

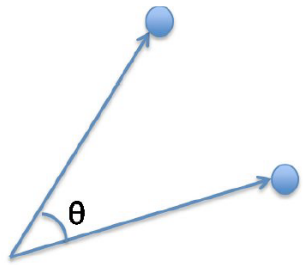
- "Blue House" -> (red, blue, house) -> (0, 1, 1)
- "Red House" -> (red, blue, house) -> (1, 0, 1)

Once we have vector representation, we can do analysis. Algorithms can handle numbers (vectors).

Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. The cosine similarity is particularly used in positive space, text data, where the outcome is neatly bounded in $[0, 1]$.

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



Term Frequency and Inverse Document Frequency

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

TF-IDF

TF Term- Frequency is the raw frequency of a word normalized by the dumber

IDF Inverse Document Frequency is the number of documents normalized by the number of documents that contain the term. For terms that are present in every document, this will lead to an IDF value of zero (that is, log(1)). For this reason, one of the possible normalizations for IDF is $1+\log(\mathrm{N}/DF_x)$.

TF-IDF

TF-IDF term x in document y

$$TF_{x,y} \times log(\frac{N}{DF_x})$$

- $TF_{x,y}$ = frequency of x in y
- DF_x = number of documents containing x
- N total number of documents

TF-IDF

The intuition behind TF-IDF is that words which are too frequent or too rare are not representative