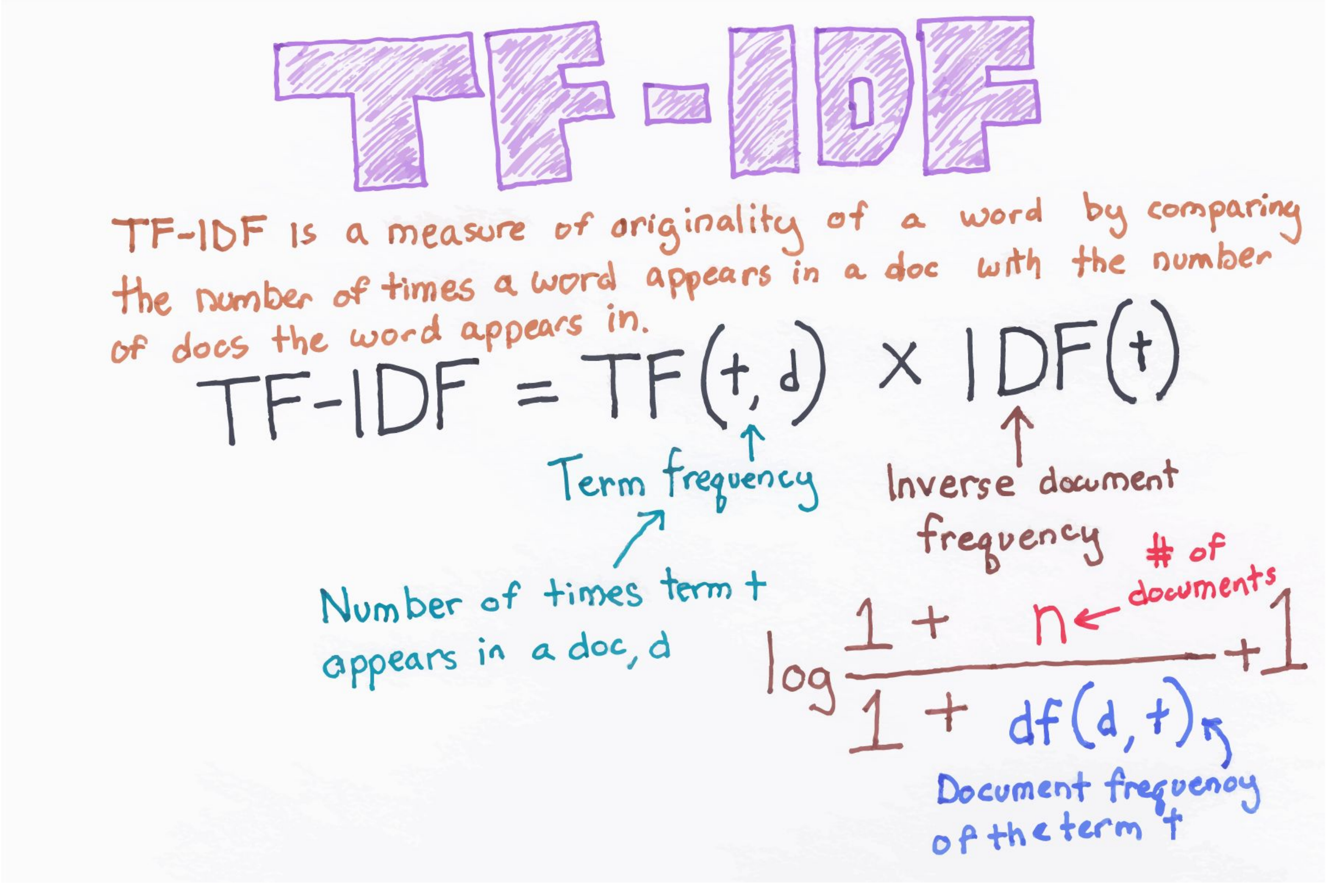
Text Analysis

September 8, 2019

Text Analysis Approaches



TF-IDF

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

$$\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)$$

Term frequency

Number of times term t appears in a doc, d

Inverse document frequency

$$\log \frac{1 + n}{1 + df(d, t)} + 1$$

# of documents

Document frequency of the term t

Text Analysis Problems

- Detecting Sentence Boundaries
- Text Search
- Stemming
- Document Clustering
- Grammar Check
- Summarization
- Sentiment Analysis
- Named Entity Recognition
- Language Translation
- Text Completion

Natural Language Processing

Note - This is NOT Neuro-Linguistic Programing (unless you're into psychotherapy)

Natural-language processing (**NLP**) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

In linguistics, a **corpus** (plural *corpora*) or **text corpus** is a large and structured set of texts (nowadays usually electronically stored and processed). They are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory.

MASC - http://www.anc.org/data/masc/ - Over 500K words of written and spoken data, including 25K words from each of 19 genres, all or parts of the data annotated for 17 different annotation types.

The New York Times Annotated Corpus - https://catalog.ldc.upenn.edu/ldc2008t19

Penn Treebank Tags

Part-of-speech tags are assigned to a single word according to its role in the sentence. Traditional grammar classifies words based on eight parts of speech: the verb (**VB**), the noun (**NN**), the pronoun (**PR**+**DT**), the adjective (**JJ**), the adverb (**RB**), the preposition (**IN**), the conjunction (**CC**), and the interjection (**UH**).

Chunk tags are assigned to groups of words that belong together (i.e. phrases). The most common phrases are the noun phrase (**NP**, for example *the black cat*) and the verb phrase (**VP**, for example *is purring*).

Relations tags describe the relation between different chunks, and clarify the role of a chunk in that relation. The most common roles in a sentence are **SBJ** (subject noun phrase) and **OBJ** (object noun phrase). They link **NP** to **VP** chunks. The subject of a sentence is the person, thing, place or idea that is *doing* or *being*something. The object of a sentence is the person/thing affected by the action.

https://www.clips.uantwerpen.be/pages/mbsp-tags

NLP Libraries

- NLTK - Natural Language Toolkit - starting point for most people interested in NLP. Easy to use Python library with a lot of examples (Google NLTK Book).
- Stanford CoreNLP - State of the art NLP library from your friends at Stanford. Check the license before use in a commercial product.
- OpenNLP - NLP library similar to CoreNLP, but under the Apache License.
- spaCy - Python based industrial scale NLP library. Easy to set up and use.
- Spark NLP (John Snow Labs) - Cluster compute for NLP tasks.

NLP Libraries

Let’s take a minute to check out the spaCy library to parse some text:

https://colab.research.google.com/github/HSV-AI/presentations/blob/master/2019/190417_spaCy.ipynb

Text Comparison

Naive Bayes Approach -

This was my first introduction to text comparison in Grad school at F.I.T. The project was intended to allow a user to enter a question, and then determine the best newsgroup to post the question to get a response.

$$P(q \mid ng) = P(ng \mid q)P(q) / P(ng)$$

Probability computed simply based on how many times the words in the question appeared. Did not take context or order of the words into consideration.

Worked much better than I expected.

Text Comparison

Text Frequency Inverse Document Frequency - TF/IDF -

Numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.

Normally used as an N*M matrix where

N = number of words in vocabulary

M = number of documents in corpus

Text Comparison

https://colab.research.google.com/github/HSV-AI/bug-analysis/blob/master/Topic_Modeling_LDA.ipynb

https://colab.research.google.com/github/HSV-AI/bug-analysis/blob/master/Doc2Vec.ipynb

Word Vectors

Word2Vec - based on a paper from Google - https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Great explanation of how it is built: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

List of Pre-trained models: http://ahogrammer.com/2017/01/20/the-list-of-pretrained-word-embeddings/

Word Vectors

Word Mover's Distance - WMD

Based on a paper from Washington University in St. Louis - http://proceedings.mlr.press/v37/kusnerb15.pdf

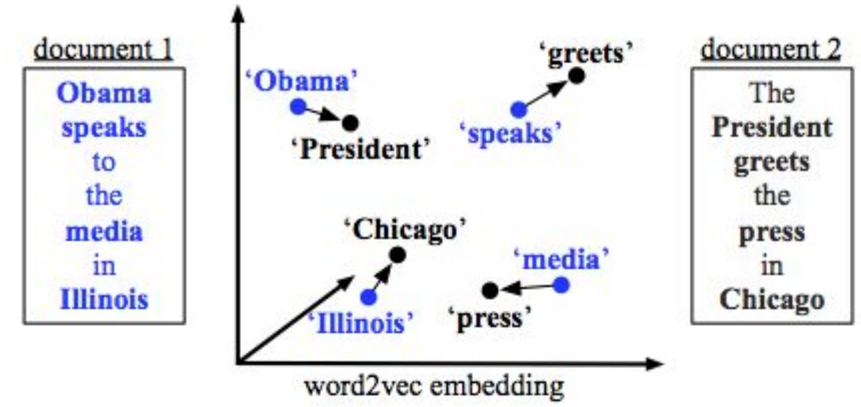


Figure 1. An illustration of the *word mover's distance*. All non-stop words (***bold***) of both documents are embedded into a *word2vec* space. The distance between the two documents is the minimum cumulative distance that all words in document 1 need to travel to exactly match document 2. (Best viewed in color.)

Word Vectors

Here’s another walk through some interesting word vector exercises:

https://colab.research.google.com/github/HSV-AI/presentations/blob/master/2019/190508_Word_Document_Vectors.ipynb

Questions/Comments?

?