Introduction to Natural Language Processing

Document Classification

Problem Motivation

- You’re Google News, and you want to group news articles by topic.
- You’re a legal tech firm, and you need to sift through 100,000 pages of legal documents to find the relevant ones.

Overview of Approach

- Compile documents
- Featurize them
- Compare their features

Simple Sample Problem

- 2 documents: “blue house” and “red house”
- Could featurize based on word counts
  - “blue house” → (red, blue, house) = (0, 1, 1)
  - “red house” → (red, blue, house) = (1, 0, 1)

Bag of Words

- 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)
- “red red house” → (red, blue, house) = (2, 0, 1)

Comparing the Features: Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

“red house” vs “blue house” →
(1,0,1) vs (0, 1, 1) →
similarity = .5

Is there a better way to featurize?

- Bag of words is naive--just word counts!
- Every word has equal weighting
  - “the” and “data” have different predictive power
- Can adjust word counts based on their frequency in the corpus

TF-IDF

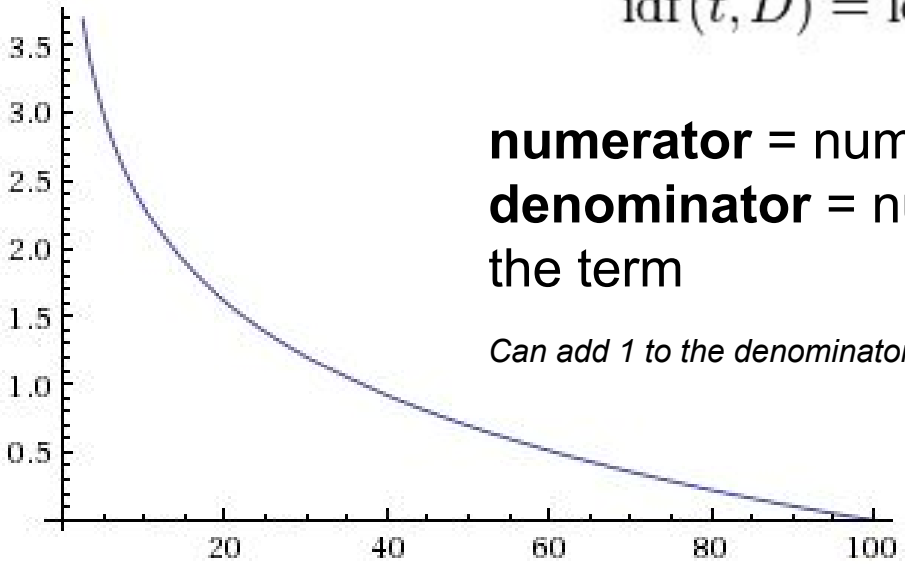
$$TF * (?)$$

- How to adjust the term frequency?
- Words found in only one document should have highest weighting
- Words found in every document should have lowest weighting

IDF - Inverse Document Frequency

$$TF * IDF$$

$$\text{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$



numerator = number of documents in corpus
denominator = number of documents in corpus containing the term

Can add 1 to the denominator to avoid zero division.

Feature Engineering for Text

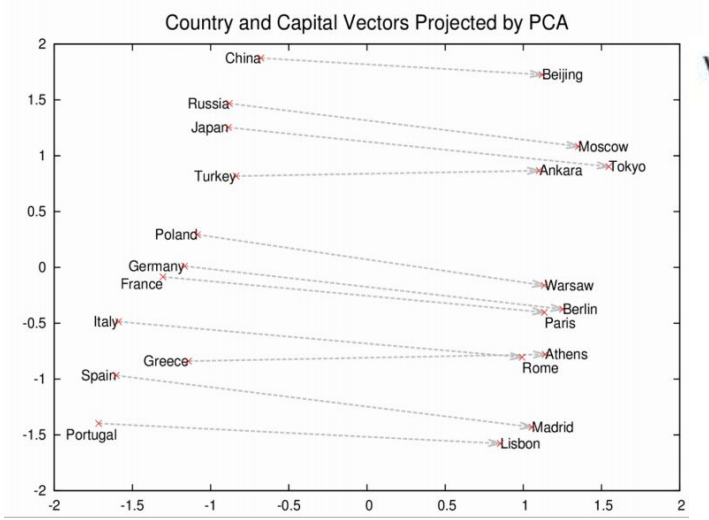
- Tokenize
- Remove stop words
- Stemming
  - e.g. “walked” → “walk”
- Lemmatization (better than stemming)
  - e.g. “people” → “person”
- N-grams / skip grams

More Advanced NLP Problem Types

- Sentiment analysis
  http://nlp.stanford.edu/sentiment/index.html
- Machine translation
  https://medium.com/s-c-a-l-e/how-baidu-mastered-mandarin-with-deep-learning-and-lots-of-data-1d94032564a5
- Probabilistic parsing

Word2Vec

- optional material, see word2vec repo
- converts words into vectors
  - geometric relationships between vectors represent semantic relationships between words



vec(“man”) – vec(“king”) + vec(“woman”) = vec(“queen”)