Natural Language Processing

Turning language into numbers

Background & Definitions

- A *document* is a sample of text data.
- A *corpus* is a set of documents.
- A *vocabulary* is the set of terms in a language.
- A *grammar* defines the syntactically correct formulations in a language using parts of speech formulations.

Morphology

Words are often made up of a:

- **prefix** - characters before the root
- **root** - the most reduced form of a word
- **suffix** - characters after the root

Morphology

Prefixes and suffixes can create

- **inflectional morphs** - maintain the meaning and part of speech *ex. plural: book, books*
- **derivative morphs** - change the meaning or part of speech *ex. teach, teacher*

Preprocessing Text

1. Tokenization
2. Stop word removal
3. Stemming / Lemmatization

Tokenization

Breaking up sentences into chunks called *tokens*.

Stop word removal

Stop words are words that add little value to the task the data will be used for.

- Commonly used filler words ex: the, and
- Frequently used words in a corpus

Stemming & Lemmatization

Stemming is a heuristic based approach to removing prefixes and suffixes.

- Usually fast
- Makes mistakes ex: saw -> s

Stemming & Lemmatization

Lemmatization uses morphological analysis and the token's part of speech to determine the appropriate lemma or reduced format for the word.

- Computationally intensive
- Very effective ex: am -> be

n-grams

- **unigrams**: ”machine”, ”learning”, ”natural”, ”language”, ”processing”
- **bigrams**: ”machine learning”, ”natural language”, ”language processing”
- **trigrams**: ”natural language processing”

Feature Representations

Document Term Matrix

A document term matrix is an $n \times m$ matrix representing a corpus of n documents and a vocabulary containing m terms.

Values in the document term matrix may vary.

Bag of Words

Stores the number of times a term appears in each document.

Bag of Words

Denver is nicer than Boulder.

Boulder is nicer than Denver.

Term Frequency

$$\underset{BoW}{tf_{t,d}} = c_{t,d},$$

$$\underset{Normalized}{tf_{t,d}} = \frac{c_{t,d}}{\sum_{i \in \mathcal{V}} c_{i,d}},$$

$$\underset{LogScaled}{tf_{t,d}} = \log\left(1 + c_{t,d}\right).$$

TF-IDF

$$idf_t = \log \frac{n}{df_t} \approx log \frac{n}{df_t + 1},$$

$$tfidf_{t,d} = tf_{t,d} \times idf_t,$$

where df_t is the document frequency of a token t.

Spam

"ii mayke are you th0usands of free for a $$$s surfling teh webz meeting early next week"

Source

Hashing Vectorizer

Use token hash instead of token itself to represent token index.

- More computationally effective
- Can handle cases where words in the test set don't exist in the training set

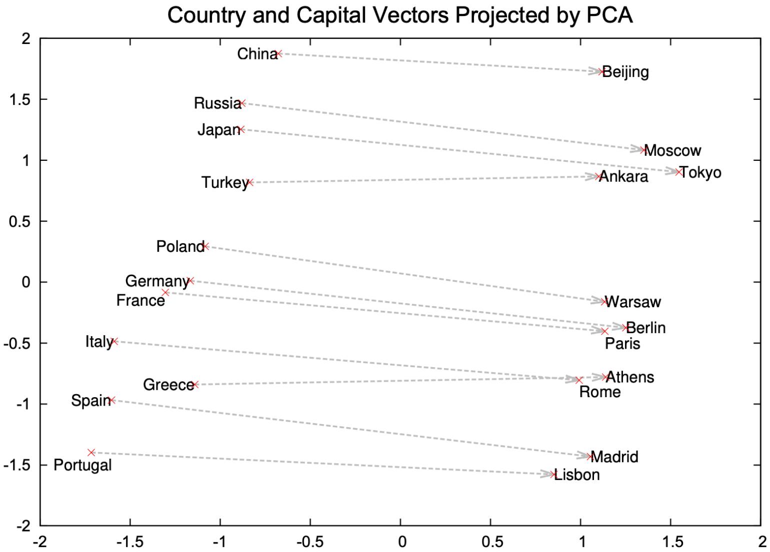
Learned Word Embeddings

- word2vec
- GloVe

Linguistic Arithmetic

$$\text{King} - \text{Man} + \text{Woman} \approx \text{Queen}$$

Linguistic Arithmetic



Demo

Topic Modeling

Latent Dirichlet Allocation (LDA)

- Assumes that there exist multiple topics within a corpus and that each document belongs to many
- Determines the topics and assigns a true/false value for each document
- Feature representation is a vector of topic belonging

Questions