Chapter 10 Representing and Mining Text

1/4/21 2:14 PM

Representation

Document: one piece of text
Token / term: units of words (can be single word for simplest)
Corpus: a collection of documents

1. *Bag of Words*

- Treat every document just as a collection of individual words
- Ignore grammar, word order, sentence structure, and punctuation

If taking each word as a possible feature, then each word is a token, and each document is represented as one (if the word is present in doc) or zero (if the word is not present)

2. *Term Frequency*

Steps of performance:

a. Case has to be normalized, either all capitalized or lowercase
b. Many words need to be stemmed. E.g. *announces, announced* and *announcing* are all reduced to the term announc
c. Stopwords are removed
**Stopword elimination may not always be good idea (for example, when representing titles of movies, with or without a "The" can be different items)

3. *Measuring Sparseness: Inverse Document Frequency*

a. If the word is too rare, in clustering the doc can only be classified into single point cluster, which is not helpful.
So we often impose a small arbitrary lower limit on number of documents in which the word must appear
b. If the word is too common, in clustering all doc will be put into one cluster
So we often impose an arbitrary upper limit on number too

Inverse Document Frequency (IDF) of a term

$$\text{IDF}(t) = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t}\right)$$

4. *Combining: TF-IDF*

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

- It's specific to a single document
- Whereas IDF depends on the entire corpus
- Each document becomes a single feature vector, and the corpus is a set of these feature vectors

Beyond Bag of Words

1. *N-gram Sequences*

- "Bag of n-grams up to three":
  - Each document is represented by features of its individual words, adjacent word pairs (bi-grams) and adjacent word triples (3-grams)
- N-gram is useful when particular phrases are significant but the component words are not
- The main disadvantage is the increase of feature set, which requires much more storage space

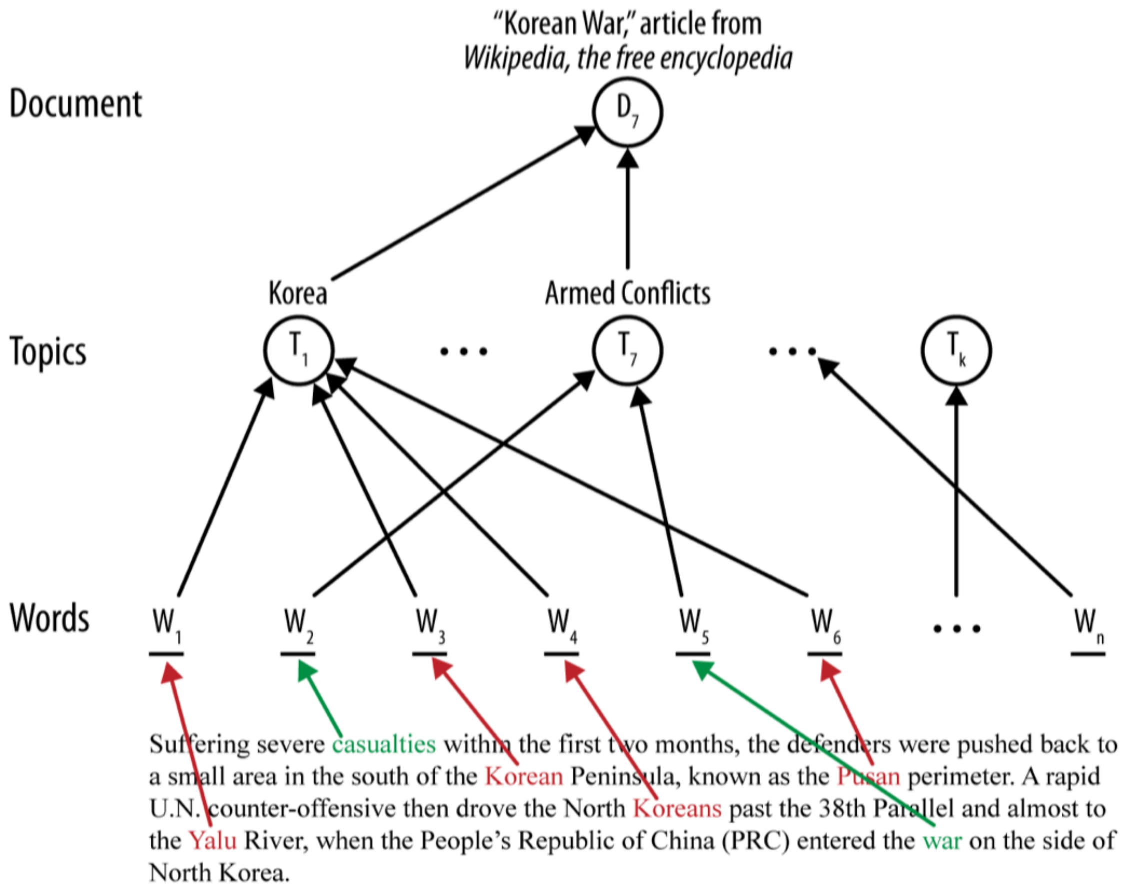
2. *Name Entity Extraction*

- Name Entity Extractor: extract unique terms such as "Silicon Valley" or "New York"
- Very knowledge based: has to be learnt, or coded by hand

3. *Topic Models*

- Modeling documents with a topic layer

-


"Korean War," article from
Wikipedia, the free encyclopedia
Document
D_7
Korea
Armed Conflicts
Topics
T_1
T_7
T_k
Words
W_1
W_2
W_3
W_4
W_5
W_6
W_n
Suffering severe casualties within the first two months, the defenders were pushed back to a small area in the south of the Korean Peninsula, known as the Pusan perimeter. A rapid U.N. counter-offensive then drove the North Koreans past the 38th Parallel and almost to the Yalu River, when the People's Republic of China (PRC) entered the war on the side of North Korea.

  - First model the set of topics in a corpus separately (usually topics are learnt from unsupervised approach from data mining)
  - Each word token is mapped to one or more topics
  - Final classifier is defined from these intermediate topics rather than words

- One advantage: in search engine, for example, a query doesn't need to match specific words of documents, but can still be classified as relevant
- General methods for creating topic models include:
  - Matrix factorization methods, such as Latent Semantic Indexing
  - Probabilistic Topic Models, such as Latent Dirichlet Allocation