**Reading 8**

**Representing text in natural language processing**

The process of transforming text into numeric stuff, similar to what we did with the image above, is usually performed by building a**language model**.

These models typically assign probabilities, frequencies or some obscure numbers to words, sequences of words, group of words, section of documents or whole documents. The most common techniques are: 1-hot encoding, N-grams, Bag-of-words, vector semantics (tf-idf), distributional semantics (Word2vec, GloVe).

# 1-hot encoding model

If a document has a vocabulary with 1000 words, we can represent the words with one-hot vectors. In other words, we have 1000-dimensional representation vectors, and we associate each unique word with an index in this vector. To represent a unique word, we set the component of the vector to be 1, and zero out all of the other components.

This representation is rather arbitrary. It misses the relationships between words and does not convey information about their surrounding context. This method becomes extremely ineffective for large vocabularies.

# N-grams language model

N-gram language models estimate the probability of the last word given the previous words. The longer the context on which we train a N-gram model, the more coherent the sentences we can generate. Even with very large corpus, in general, N-gram is an insufficient model of language because language has long-distance dependencies. Furthermore, the N-gram model is heavily dependent on the training corpus used to calculate the probabilities.

# Bag-of-words language model

These are represented by text as a bag of words, as if it were an unordered set of words, while ignoring their original position in the text, keeping only their frequency.

he bag-of-words representation of text in a simple sentiment analysis example with the two classes positive (+) and negative (-).The purpose is to classify the last sentence as either positive or negative.

This task is solved by a so-called **Naive Bayes Classifier**, which uses the words frequencies in the bag-of-words of each class to compute the probability of each class *c*, as well as the conditional probability of each word given a class,

Bag-of-words language models rely on the term frequency TF, defined as the number of times that a word occurs in a given text or document. Bag-of-words helps in sentiment analysis.

# Vector semantics

We define a word by counting what other words occur in its environment, and we can represent the word by a vector, a list of numbers, a point in N-dimensional space. Such a representation is usually called **embedding**.

**TF-IDF language model**

Vector semantic models use the raw frequency of the co-occurrence of two words. In natural language, raw frequency is very skewed and not very discriminative.

TF-IDF algorithm is by far the dominant way of weighting co-occurrence matrices in natural language processing, especially in information retrieval. The TF-IDF weight is computed as the product of the term frequency and the inverse document frequency. It helps us to assign importance to more discriminative words.

The term or word frequency is calculated as the number of times the word appears in the document.

The document frequency of a given term or word is the number of documents it occurs in. The inverse document frequency is the ratio of the total number of documents over the document frequency. This gives a higher weight to words that occur only in a few documents.

GloVe optimizes the embeddings directly so that the dot product of two word vectors equals the log of the number of times the two words will occur near each other (within a 2-words window, for example). This forces the embeddings vectors to encode the frequency distribution of which words occur near them.

**Topic Modeling Tutorial with Latent Dirichlet Allocation (LDA)**

**Data Pre-processing**

Pre-processing the text is important because language is ambiguous at all levels: lexical, phrasal, semantic. We make the text data into all lowercase and remove **stopwords**.

We also **tokenize** our data.Tokenizers work similar to regular expressions and are used to divide tweets into lists of words.

**Latent Dirichlet Allocation (LDA)** is a probabilistic transformation from bag-of-words counts into a topic space of lower dimensionality. Text data are will be seen as a distribution of topics. Topics are represented by a distribution of all words in the vocabulary.

**Topic Coherence**technique is usually preferred to Perplexity techniques. With coherence, we quantify the coherence of a topic by measuring the degree of semantic similarity between its high scoring words. This leads to topics which are more human interpretative. The technique selects the top frequently occurring words in each topic. It then computes and aggregates all pairwise scores (UMass) for each of the words to calculate the coherence score for a particular topic.