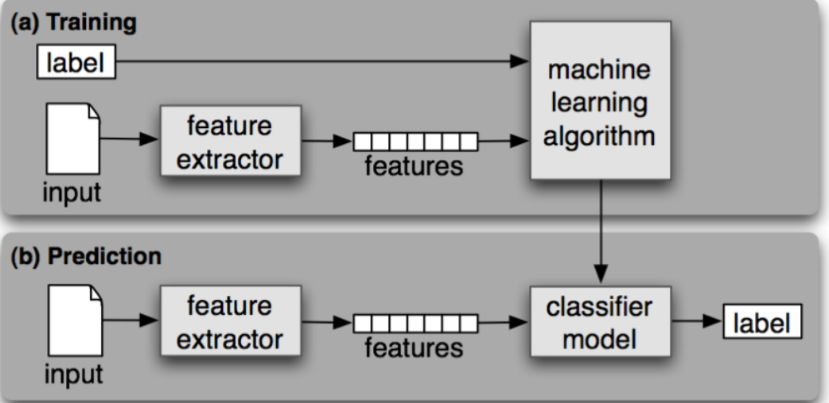
# Sentiment Analysis

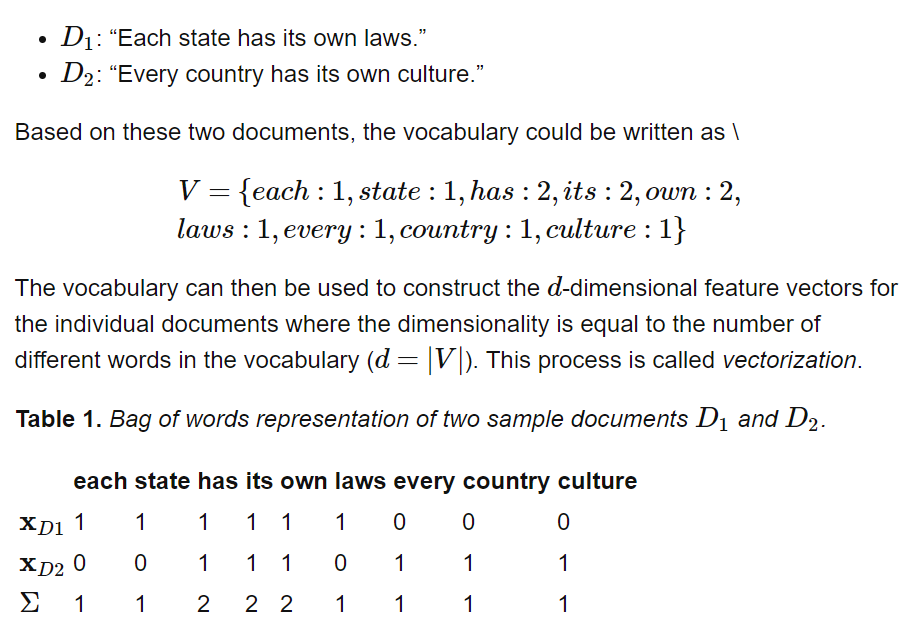
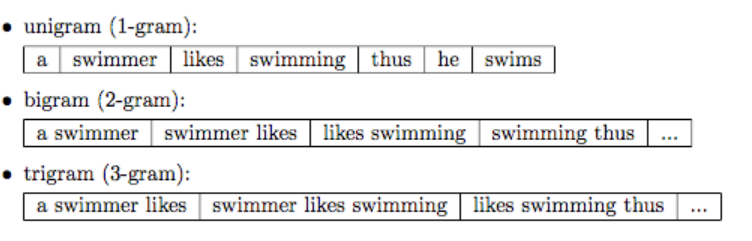
* Machine learning: Process of designing algorithms that make predictions
* NLP (Natural Language Processing): Describes the way computers process, understand and interpret human language
* Sentiment analysis – Automated process of understanding an opinion about a given subject (written/spoken language)
  + Unstructured info -> Structured data that can be analysed
  + Subjectivity classification (subj/obj)
  + Polarity classification (+ve,-ve,neut)
  + 80% of worlds data is unstructured and there is too much data to manually process. SA can process data in an efficient and cost-effective way
  + 60-65% agree when judging the sentiment of a piece of text
* Algorithms:
  + Rule-based: Based on a set of manually crafted rules
    - Use stemming tokenization, POS tagging, parsing, etc. (Classic NLP techniques)
    - Naïve approaches.
    - Require a lot of tweaking/maintaining the rules
  + Automated: Machine learning
    - SA task is modelled as a classification problem, where the classifier is fed with a text and returns the corresponding category
    - Model learns to associate input (text) to output (tag) based on test samples.
    - Feature extractor transfers the text input into a feature vector
    - Pairs of feature vectors are fed into the machine learning algorithm to get a model.

Machine Learning Steps

1. **Feature extraction/Text vectorization** – Transform text into numerical representation (vector). Each component represents the frequency of a word in a predefined dictionary (lexicon of polarised words).
   1. Traditional approach: Bag-of-words with their frequency

Methods of text vectorization

**Bag of words:**

* Define a fixed length vector where each entry corresponds to a word in the pre-defined dictionary of words. So, size of vector = size of dictionary
* For a text, count the number of times each word in dictionary appears in text. Put number in corresponding vector entry.
* To improve, various methods are used (instead of count):
  + Stop-words: Useless words that don’t affect meaning of sentence too much
  + Lemmatizing: Process of shortening words to its base meaning that is looked up in a dictionary (lemma)
  + N-grams: Token can be defined as a sequence of n items. (1-gram: Word consists of one word/letter/symbol)
    - Tokenization: Breaking down text corpus into individual elements.
* Problem: Does not capture meaning/context of text

**Tf-idf: Term Frequency-Inverse Document Frequency.**

Measures how import a term is relevant to a document and to a corpus. Term frequency: Number of times the term appears in one document. Inverse-document freaqunecy: How important a term is to a collection of documents. Multiply TF and IDF together. Returns vector per word per document based on frequency

Used in automatic blog tagging,

Corpus: Collection of documents.

**Word2vec:**

Algorithm that transfers words to vectors, so that words with similar meanings end up laying close to each other.

Return vectors per word based on co-occurrence info. Words with similar meanings end up laying close to each other. Is a predictive model -> Learned through neural networks.

* + Co-occurrence info: How frequently words appear with other words in a corpus (word’s context)
  + There are pre-trained vectors. You can train on your own data set, or create from scratch. <https://p.migdal.pl/2017/01/06/king-man-woman-queen-why.html>

GloVe: Return vectors per word based on co-occurrence info. Words with similar meanings are close to each other. Is a ‘count-based’ model -> doing dimensionality reduction on co-occurrence count matrix.

1. **Classification Algorithms (Machine Learning):**

* Naïve Bayes: Calculate probability of each tag for a given text and then output the highest one. Probabilities are calculated using features. <https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/>

Features: Info taken from text and given to algorithm. In this case, feature = word frequencies

Naïve: Assume that every word in a sentence is independent of others. So, when taking conditional probability, multiply probability of each word.

In order to improve the performance of Naïve Bayes, a variety of methods such as stopwords, lemmatizing, etc. can be applied.

* Linear Regression
* Support Vector Machines: Points in multidimensional space. Different sentiments belong to different regions of space. Better got limited number of samples
* Deep Learning: <https://machinelearningmastery.com/what-is-deep-learning/> Machine learning that is inspired by the structure and function of the brain. Performs great with huge amounts of data and supervised learning. Can perform automatic feature extraction from raw data (feature learning)

Analysis, Metrics and Evaluation:

* Cross validation: Use 75% of data to train, 25% to test. Repeat multiple times. Use average to determine accuracy
* Precision measures how many texts were predicted correctly as belonging to a given category out of all of the texts that *were predicted (correctly and incorrectly) as belonging to the category*
* Recall measures how many texts were predicted correctly as belonging to a given category out of all the texts that *should have been predicted as belonging to the category*. We also know that the more data we feed our classifiers with, the better recall will be.
* Accuracy measures how many texts were predicted correctly (both as belonging to a category and not belonging to the category) *out of all of the texts* in the corpus.
* Precision and recall are used to predict accuracy.

Applications:

1. Social Media Monitoring
2. Brand Monitoring
3. Customer Feedback
4. Customer Support
5. Product Analytics
6. Market Research and Analysis