## Python Tools

used in MusicMood

(website: http://sebastianraschka.com/Articles/2014\_musicmood.html; github: https://github.com/rasbt/musicmood)

**pandas** - Python Data Analysis Library

**scikit-learn** - Machine Learning in Python (classification, regression, clustering, …)

**hdf5** - lets you store huge amount of numerical data

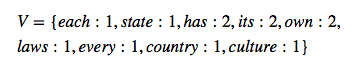
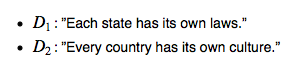
**Natural Language Toolkit** - classification, tokenization, stemming, tagging, parsing, semantic reasoning

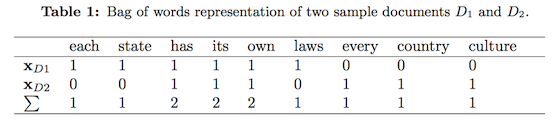
# NAIVE BAYES AND TEXT CLASSIFICATION

(http://sebastianraschka.com/Articles/2014\_naive\_bayes\_1.html#3\_1\_2\_stop\_words)

## THE BAG OF WORDS MODEL

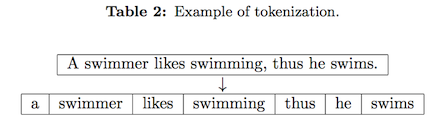
* **feature extraction**
  + dictionary: non-redundant items
  + vectorization: construct d-dimensional feature vectors



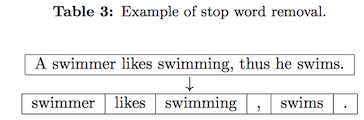


* **tokenization**
  + break down text corpus into individual elements

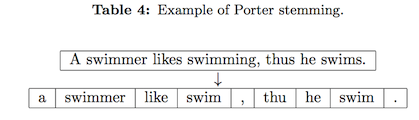
-> input for natural language processing algorithm



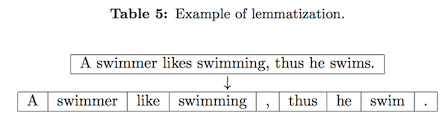
* + remove stop-words
    - * search against language specific stop word dictionary OR
      * create a stop list by sorting all words in the entire text corpus by frequency
      * then remove all stop words

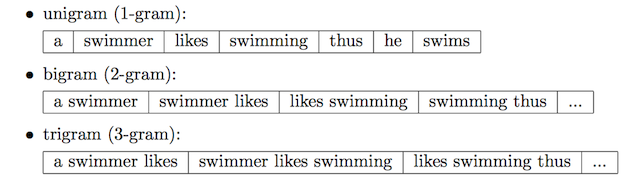


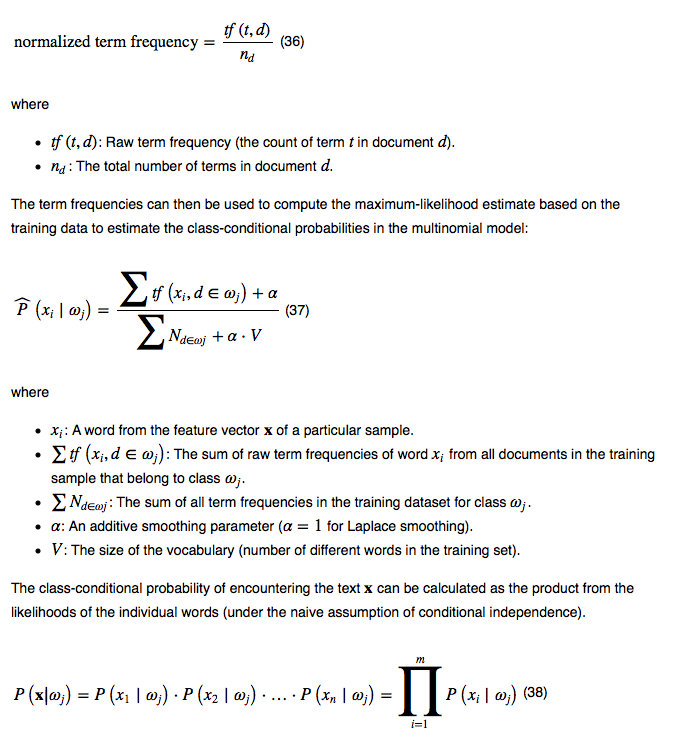
* + (stemming OR) **lemmatization** *-> use natural language tool kit!!!*
    - stemming: transform words into root from



* + - lemmatization: obtain canonical forms of words (lemmas)



* **construction of n-grams**
  + define tokens as sequence of n items
* **multinomial naïve bayes** *(characterize text documents)*
  + term frequency

****

* + Term Frequency - Inverse Document Frequency (Tf-idf)

*-> often used in related papers*

* + - Weighted term frequency

*(useful when stop-words still in text)*

* + - assumes that the importance of a word is inversely proportional to how often it occurs across all documents

