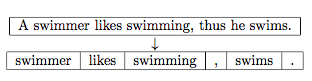
Stop Words

*Stop words* are words that are particularly common in a text corpus and thus considered as rather un-informative (e.g., words such as *so*, *and*, *or*, *the*, …”). One approach to stop word removal is to search against a language-specific stop word dictionary. An alternative approach is to create a *stop list* by sorting all words in the entire text corpus by frequency. The stop list — after conversion into a *set* of non-redundant words — is then used to remove all those words from the input documents that are ranked among the top *n* words in this stop list.

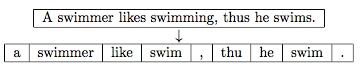
**Table 3.** *Example of stop word removal.*



Stemming and Lemmatization

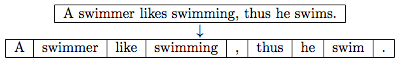
*Stemming* describes the process of transforming a word into its root form. The original stemming algorithm was developed my Martin F. Porter in 1979 and is hence known as *Porter stemmer* [[8](https://sebastianraschka.com/Articles/2014_naive_bayes_1.html#References)].

**Table 4.** *Example of Porter Stemming.*



Stemming can create non-real words, such as “thu” in the example above. In contrast to stemming, *lemmatization* aims to obtain the canonical (grammatically correct) forms of the words, the so-called *lemmas*. Lemmatization is computationally more difficult and expensive than stemming, and in practice, both stemming and lemmatization have little impact on the performance of text classification [[9](https://sebastianraschka.com/Articles/2014_naive_bayes_1.html#References)].

**Table 4.** *Example of Lemmatization.*



(The stemming and lemmatization examples were created by using the Python NLTK library, [http://www.nltk.org](http://www.nltk.org/).)