**NLP**

Natural language processing. A system or algorithm that engages in a means of attempting to extract content from text, or generate text that can be interpreted as having content.

**NLU**

Natural language understanding. A system or algorithm that tries to extract content or meaning from text.

**NLG**

Natural language generation. A system that attempts to assemble language together in a fashion that conveys meaning or content. Chatbots and the like.

**Feature Engineering**

Determining beforehand what features you want an algorithm to focus on. Requires subject matter experts, and is dissimilar to how humans learn. Can be useful in specific business applications of AI-like systems, but not for the real deal.

**Feature Learning**

The algorithm determines for itself which features from the data are important. Useful in situations where it is unknown what features are important, or for deriving new information where SMEs fail or are inaccurate.

**Statistical NLP**

Translates sentences and words into mathematical vectors. Can be used to determine if sentences are “close” or “far” from each other, based upon contents, but can result in incorrect analogies. In essence, very similar to METACAT. Easily scalable.

**Symbolic NLP**

A largely rules-based system, similar in nature to symbolic logic. Involves symbol pushing, but doesn’t have actual content. Requires expert knowledge of linguistics and semantics to pull off, to build the system. Much more in common with expert systems.

**Top-Down NLP**

An approach that starts, obviously, at the top. Begin at the top of the taxonomy chart with your documents, finding large, broad classifications before moving down the chain, specializing and refining at each level.

Bottom-Up NLP

Starting at the bottom, with a single document as an example. Tag and do all the low-level defining here, gradually working your way up the chain. Has a tendency to categorize and tag, rather than build taxonomies.

**Case-Based Reasoning**

Allows a system or algorithm to use previous information to build a history about some kind of data.

**Hypernym**

Noun with a more general definition than another. “Dog” is a hypernym of “German Shepherd.”

**Hyponym**

Noun with a more specific definition than another. “Dog” is a hyponym of “mammal.”

**Troponym**

As per hypernym, but for verbs.

**Tokenization**

Breaking sentences down into words. String tokenization, it’s a thing.

**Stop Words**

Words that are useless, or if not useless, then will just clutter up the analysis. Connecting words, or very common words, might fall into this category; in addition, some words may be counted as stop words for particular projects, depending on the content and purpose.

**Stemming**

The process of turning conjugated words into their root form. Run, running, runs – all of these should really be counted as one word, run, and stemming is the process by which that happens.

**Unigram/Bigram**

Unigram is a single word; bigram is two. Bet you can’t guess what a “trigram” would be.

**TFIDF**

Term Frequency, Inverse Document Frequency. A mathematical way of figuring out if a particular term is occurring more or less frequently in a given document than it “should.”

**Dale-Chall Readability Formula**

A formula that compares the words used in a document with a known list of words that children of a particular age should know.

**Lexical Knowledge Base**

A lexicon, but also contains how words relate to each other. Thesaurus, and such, as well as hypernyms and hyponyms (q.v.).

**POS-Tagging**

Part-of-speech tagging. Labeling different parts of speech with a particular tag so they can be identified, such as “noun” or “verb” or the like.

**Chunker**

A thing that “chunks” sentences, into things like noun phrase, verb phrase, and so forth. A sort of shallow parser, useful but powerful.

**PPMI**

Positive point-wise mutual information, often shortened to PMI. A measure of how likely words are to show up together, to determine if two words are more likely to show up together or near each other in a given text.

**Document Clustering**

A process by which a set of documents are organized into clusters, groups that are vaguely related in some fashion. These are typically not a strict taxonomy: instead the idea is more to work with similarity or “themes,” one might say.

**Content-Based Classification**

Organizing documents based upon their content. “Spam or not-spam” is the classic example: we have an idea of the content of a document, and we use that to classify it into a cluster.

**Descriptor-Based Classification**

Rather than examples, this sort of classification instead is more of a request or description of what sorts of documents should go in given clusters.

**Canonical Topic Modeling**

A situation in which the clusters or topics that documents are to be sorted into are received from an authority of some kind.

**Organic Topic Modeling**

A system in which documents are sorted without any sort of external system in place to enforce a particular sort of organization.

**Entity-Centric Topic Modeling**

A system of organizing documents in a way that focuses on entities within them: for instance, a set of topics may be related to a particular entity, so regardless of the rest of their content, they would be put into that entity’s cluster.

**Sentiment Analysis**

A process by which the general feeling of a document can be determined. Useful for social media.