Concepts of Text Analytics

Text Analytics is thought of as unstructured data because it does not have tables and fields. There is also the fact that text is very dirty. People misspell words, text can have underlying meanings and contain synonyms and homographs and can be domain specific. Therefore, the field of text analytics is quite hard and requires specific knowledge and tools. There is a text analytics process and it has 10 steps:

1. Text identification – mining, parsing, identification, extraction, categorization, clustering
2. Text mining – extraction of concepts, entities, relations, events
3. Text categorization – creation of taxonomies
4. Text clustering – search access, web crawling, indexing, duplicate document identification
5. Search access – analyze all major file formats and all major languages (Natural language/Semantic toolkits)
6. Entity/relation modeling
7. Link analysis – link analysis, link text repositories
8. Sentiment analysis – ability to identify and analyze sentiments, people, places, and other information from various text formats
9. Summarization – document summarization features and records management
10. Visualization

And there are many applications for text analytics that include:

* Social media monitoring
* Scientific discovery for life science and more
* E-discovery and records management
* Automated ad placement

# Terminology

According to the chapter in “Data Science for Business” most of the terminology is borrowed from Information Retrieval (IR).

* Document – one piece of text no matter how large or small
* Tokens or Terms – a single word
* Corpus – collection of documents

# Basic Approaches

The ‘bag of words’ approach is to treat every document as a collection of individual words. This approach ignores grammar, word order, structure, punctuation and treats every word as a potentially important word. The way this is represented is every word is a token and each document is represented as either a 1 (contains the word) or 0 (does not contain the word.

Instead of representing the document as a 1 or 0 when looking at a word, the Term Frequency approach counts the number of times a word appears in the document and returns that. The previous two approaches can be combined to return a structured table with tokens as column names and the document it refers to as the row name.

Sometimes it is useful to measure the sparseness of a word because we want words that are not too common, but not too rare also. Should we be interested in a word that occurs only once? This is where the Inverse Document Frequency (IDF) equation comes in. The equation

Is used to calculate said sparseness of a word. This can be though of as an equation to boost rare words in a document. As an example, a token that appears once in a document has a very high IDF and common stop words have an IDF of around 1.

Combining the Term Frequency and Inverse Document Frequency approaches yields the TFIDF approach and the equation is

This is document specific and term counts with the document come from the TF part and the document counts across the corpus come from the IDF part.

# Beyond Basic Approaches

The N-gram sequences approach is used similarly to the bag of words approach except for it preserves the structure and return sequences of adjacent words as well as the words separately. The adjacent pairs of words are called bi-grams. N-grams are useful when phrases are important but single word values might not be and the downside is that it produces mass amounts of features.

Named Entity Extraction is an approach that can recognize common named entities such as New York or Game of Thrones. This approach is knowledge intensive meaning that it requires a large corpus to train on or needs to be hard coded with knowledge of such named entities

Topic models are another more advanced approach. Instead of returning word values as the final classifier, this approach maps the words to specific topics and then returns the topics as a final classifier. This is useful for search engines because the query can contain words that do not match exactly and still return the relevant documents. Methods used to create topic models include matrix factorization methods, like Latent Semantic Indexing, and Probabilistic Topic Models, like Latent Dirichlet Allocation.