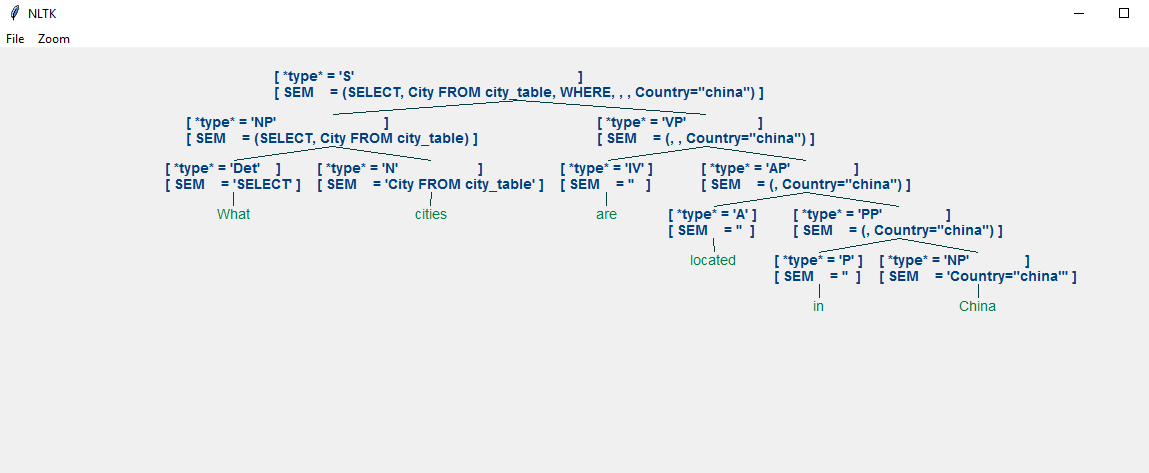
**An overview of Natural Language Processing**

Natural Language Processing (or NLP for short) is the subfield of Computer Science that aims at understanding and parsing text. While it has received much less publicity and fewer publications than Computer Vision it is nonetheless an interesting and promising field.

In NLP we use Corpora, which are large text collections. Usually, these are parsed in chunks: letters, words, sentences, paragraphs, or documents. Let’s review some of the most common tasks performed with NLP:

* **Part of Speech (POS) Tagging**: Identify and mark the part of speech of a certain word within a sentence. This is usually the first step in a bigger task. As an example, the sentence "Mary visited her grandparents" will be tagged (using Python’s nltk library) like this: [('Mary', 'NNP'), ('visited', 'VBD'), ('her', 'PRP$'), ('grandparents', 'NNS')]. Don’t worry, reading it is simple: NN refers to nouns, so “Mary” is a *proper noun* while “grandparents” code stands for *noun plural* (usually the ‘s’ in the end of the word denotes plural). As you’ve guessed, VB is a verb, and VBD is *past tense verb* (notice how many words in past tense end with ‘d’). ‘Her’ is a *personal pronoun*.
* **Similar Words**: Finding words with similar meaning or some other common characteristic (e.g. similar spelling). Let’s see words that are identified as similar to “Lord” in The Book of Genesis: earth, land, father, man, field, children, brother, son, ground, woman, ark, country, place, way, sister, waters. Some are close (e.g. father) but most are not. This occurs because the function used works on statistics and counts co-occurrences of words and whether words appear in similar contexts. Recently better techniques have been developed using Neural Networks, like **Word2Vec** which transforms word into vectors, again based on co-occurrences but the vectorization is designed so that these ‘meanings’ are learned. Word2Vec has allowed us to perform intuitive ‘word arithmetic’ by encoding useful relationships among the data. A common example is: king – man + woman = queen.
* **Statistics**: Calculating how often a word appears in a text, in how many sentences, or different forms, or finding the average length of a word, or how many times on average each word is used, **lexical richness** (which, by the way, is 16 for The Book of Genesis), or even reading difficulty.
* **Features**: Machine Learning algorithms like the ones we reviewed in previous articles do not work with text, they need numerical vectors. Converting text into vectors can be done in multiple ways, we will only mention **Bag-of-Words**, where each sentence is considered a collection of words. For every sentence, we have a vector with length equal to our dictionary (usually the most common words in a text). Each position in the vector indicates the presence or absence of a word. If the third position refers to “apples” and we have 0 then that word is missing from our sentence. If the fourth position refers to “oranges” and we have a value of 2, then that word appears twice in our sentence. There are many more ways to extract features, we leave that to a later article.

There are many more applications that use NLP: Autocorrect, Automatic Language Translation, Chatbots/Assistants (like Siri, Cortana, etc), Spam Detection, predicting movie ratings based on reviews, filtering toxic comments, and even Voice Recognition and Generation. We will review some of these in more detail in future articles (with code),so stay tuned!