SyntaxNet: Neural Models of Syntax from Google

Submitted by: Supriya Puri (puri6@illinois.edu)

Language is the heart of most human endeavors – being able to communicate is one of the most important socio-economic indicators of success. But unlike a computer language, which is highly structured, human speech however is not always precise and organized and thus can be hard enough for a human to understand, let alone a computer. Despite language being one of the easiest things for human mind to learn, it holds some complex variables in its linguistic structure such as slang, regional dialects and social context which makes it difficult for the computers to master. Natural Language Processing is a field of study which focuses on computer understanding and manipulation of the human language in a smart and valuable way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as sentiment analysis, speech recognition, translation, and part of speech tagging.

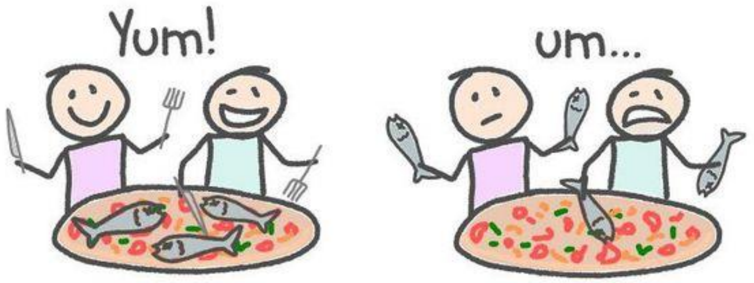
SyntaxNet is an open-source neutral network framework implemented in Tensorflow that provides a foundation for Natural Language Understanding Systems. This Tensorflow based dependency parsing library has been introduced by Google in 2016, as an approach towards how computer systems can and understand human language to process it in an intelligent way. The library gives access to a line of neural network parsing models published by Google researchers over 2014-15.

How does SyntaxNet work?

Syntactic parsing is a bottle-neck technology in NLP whose purpose is to draw exact meaning of the text. Syntax analysis checks the text for meaningfulness comparing the rules of formal grammar. It describes a sentence’s grammatical structure to help another application reason it. SyntaxNet is a framework for syntactic analysis. Given a sentence as an input, it tags each word with a part of speech (POS) tag that describes the relationship between words in the sentence represented in the dependency parse tree. SyntaxNet parses sentences to understand what role each word plays and how they all come together to create real meaning. The system tries to identify the underlying grammatical logic---what's a noun, what's a verb, what the subject refers to, how it relates to the object---and then, using this info, it tries to extract what the sentence is generally about---*the gist*, but in a form, machines can read and manipulate

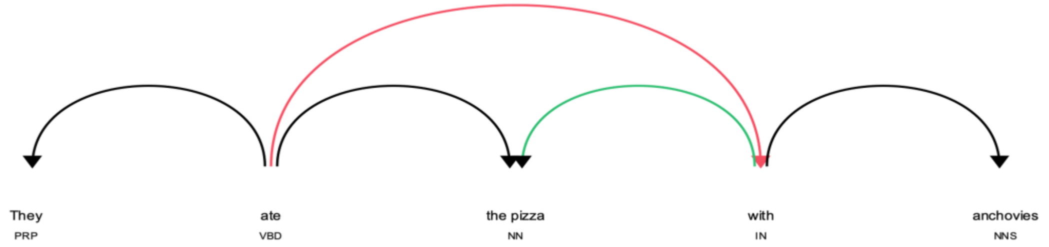
Let’s take a very simple example:

They ate the pizza with anchovies



Creative Commons Attribution-NonCommercial 2.5
James Constable, 2010

A correct parse would link “with” to “pizza”, while an incorrect parse would link “with” to “eat”:



SyntaxNet is a library for training and running syntactic dependency parsing models. SyntaxNet applies neural networks to the ambiguity problem. An input sentence is processed from left to right, with dependencies between words being incrementally added as each word in the sentence is considered. At each point in processing many decisions may be possible—due to ambiguity—and a neural network gives scores for competing decisions based on their plausibility. Instead of simply taking the first-best decision at each point, multiple partial hypotheses are kept at each step, with hypotheses only being discarded when there are several other higher-ranked hypotheses under consideration.

One model, Parsey McParseface provided by SyntaxNet offers a particularly good speed/accuracy trade-off.

Parsey McParseface:

Parsey McParseface is an English parser that has been trained to analyze English text. It is built on powerful machine learning algorithms that learn to analyze the linguistic structure of language, and that can explain the functional role of each word in each sentence. On a standard benchmark consisting of randomly drawn English newswire sentences and combining machine learning and search techniques, Parsey McParseface recovers individual dependencies between words with over 94% accuracy, beating our own previous state-of-the-art results. It also leans on SyntaxNet’s neural-network framework for analyzing the linguistic structure of a sentence or statement, which parses the functional role of each word in a sentence. While the accuracy is not perfect, it’s certainly high enough to be useful in many applications. The major source of errors at this point are examples such as the prepositional phrase attachment ambiguity, which require real world knowledge (e.g., that a street is not likely to be in a car) and deep contextual reasoning.

New Models and Upgrade:

Building machine learning systems that work well for languages other than English has been an ongoing challenge. From time to time, because of years' worth of research on multilingual language understanding, various upgrades have been incorporated and have been made available to anyone interested in building systems for processing and understanding text. The upgrade extends TensorFlow to allow joint modeling of multiple levels of linguistic structure, and to allow neural-network architectures to be created dynamically during processing of a sentence or document.

Parsey's Cousins – a collection of pretrained syntactic models for 40 languages has been introduced and can analyze the native language of more than half of the world's population at often unprecedented accuracy.

Parsey and Parsey's Cousins can be operated over sequences of words. SyntaxNet upgrade made it, for example, easy to build character-based models that learn to compose individual characters into words (e.g., 'c-a-t' spells 'cat'). By doing so, the models learnt that words can be related to each other because they share common parts (e.g., 'cats' is the plural of 'cat' and shares the same stem; 'wildcat' is a type of 'cat'). New pretrained models called ParseySaurus which used the character-based input representation made it easier to predict the meaning of new words based both on their spelling and how they are used in context. Accuracy improved (reducing errors by as much as 25%), particularly for morphologically-rich languages like Russian, or agglutinative languages like Turkish and Hungarian.

Comparision with spaCy:

spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. Unlike NLTK, which is widely used for teaching and research, spaCy focuses on providing software for production usage. spaCy also supports deep learning workflows that allow connecting statistical models trained by popular machine learning libraries like TensorFlow, PyTorch or MXNet through its own machine learning library Thinc.

SyntaxNet is a library for researching NLP models, while spaCy is a library for applying NLP models in production. SyntaxNet's parser and POS tagging models are more accurate, load faster, and consume much less memory than spaCy. spaCy's models are faster, and come with built-in sentence segmentation, which SyntaxNet lacks. Real world text is not so well behaved, particularly from social media, where punctuation is bad and new lines are used inconsistently. SyntaxNet is trained to expect input with perfect segmentation. If just dependency parsing is needed, SyntaxNet is the best choice but for entity recognition, sentiment analysis, coreference, then we will need to include other systems like spaCy. The best of both worlds will be to load the analyses of SyntaxNet into spaCy for use, possibly with spaCy used as a sentence segmentation pre-process.

More details on Documentation and installation guides can be found on:
Tensorflow GitHub repo:
https://github.com/tensorflow/models/tree/f2f25096d3dc6561a855dab914cf2913100728d6/research/syntaxnet
More information about documentation : https://openbase.com/python/syntaxnet
Library installation: https://libraries.io/pypi/syntaxnet-with-tensorflow

Conclusion:

Until recently one couldn’t write a software to control a car, tweak the tone of an email or analyze customer sentiments through feedback. But NLP has helped make this inevitable true. I agree that within a large value chain, SyntaxNet is a low-level technology. While the accuracy is not perfect, it’s certainly high enough to be useful in many applications. The major source of errors at this point are examples such as the prepositional phrase attachment ambiguity and deep contextual reasoning. It would be great to see a bridge between Parsey McParseface and spaCy, so that you can use the more accurate model with the sweeter spaCy API. The next steps must include measures to solve text-based ambiguity by providing a range of pre-trained models, adapted to different languages and genres. For instance, in well edited text such as a financial report, you want the model to consider capitalization as a decisive indicator — but if you’re parsing tweets, capitalization is almost meaningless.

References:

- What is natural language processing? - https://algorithmia.com/blog/introduction-natural-language-processing-nlp
- NLP in TensorFlow — All You Need for a Kickstart - https://aqsakausar30.medium.com/nlp-in-tensorflow-all-you-need-for-a-kickstart-3293d7d2630e
- SyntaxNet in context - https://explosion.ai/blog/syntaxnet-in-context
- Announcing SyntaxNet - https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html
- https://www.wired.com/2016/05/google-open-sourced-syntaxnet-ai-natural-language/
- Parsey McParseface - https://thenextweb.com/news/google-just-open-sourced-something-called-parsey-mcparseface-change-ai-forever
- spaCy - https://en.wikipedia.org/wiki/SpaCy
- SyntaxNet vs spaCy - https://www.quora.com/How-does-Googles-open-source-natural-language-parser-SyntaxNet-compare-with-spaCy-io-or-Stanfords-CoreNLP
- SyntaxNet Upgrade - https://ai.googleblog.com/2017/03/an-upgrade-to-syntaxnet-new-models-and.html
- Parsey’s Cousins- https://ai.googleblog.com/2016/08/meet-parseys-cousins-syntax-for-40.html