Zach Bryant
Capstone 1 – Gary McKenzie
4/21/2017
Research Paper

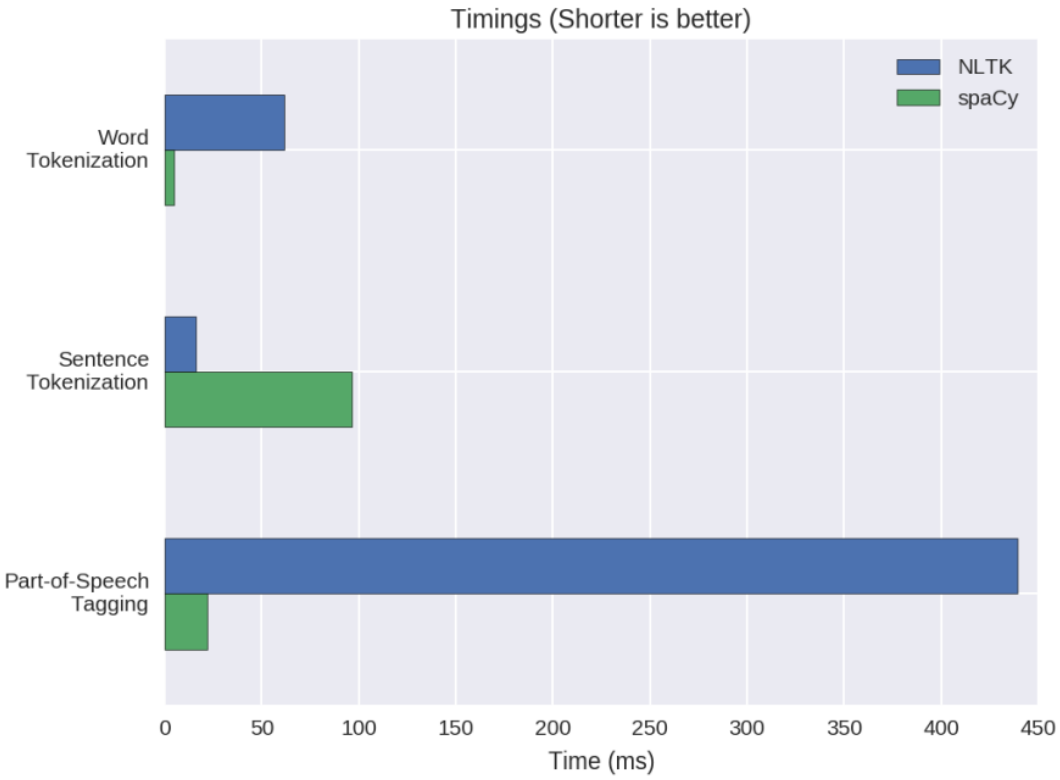
NLP: Natural Language Processing

Ever since computers were invented, computer scientists have been working on ways to model human intelligence. This includes teaching computers how to understand natural language. Natural language is any language that humans use to communicate with each other on a daily basis. Natural language processing or NLP is a field of study in computer science that falls under the category of artificial intelligence and aims to solve the problem of allowing computers to understand, interpret, and speak natural language. This paper will be focusing on the applications of NLP and general information about NLP that is good to know for programming purposes.

So what is natural language processing to begin with? It is the processing of natural language by computers. Natural language is any human made language – not a computer language like C, Java, or Fortran nor a mathematical language like numbers and arithmetic. Any system that takes natural language as an input and does processing on it is a NLP system. For example, when Google Translate is used to translate a word from English to Spanish, this is a natural language process. The input is a human language – English. Some processing is done on it to change it to the desired output – Spanish, a natural language. This is just one of many examples of NLP systems. Some other examples of NLP systems include: automatic summarization, sentiment analysis, speech recognition, and automatic generation of keyword tags. An important thing to note about NLP is that is considers the hierarchical structure of human language – words make a phrase, phrases make a sentence, and sentences convey ideas (1).

Why would anyone want to use natural language processing? NLP has many uses, but more importantly, for a computer to be considered intelligent, it should be able to understand, interpret, and speak natural language. The more that is done to improve the algorithms surrounding natural language processing, the more intelligent computers will become. With this in mind, some algorithms used in NLP include: language modeling, parsing, text classification, and part-of-speech tagging. Additionally, another element that is usually paired with NLP is machine learning. With NLP, it is possible derive more information from the endless data being pumped through the Internet in the form of natural language. Meaning, NLP can be used to look at the tweets on a particular topic and be able to tell if people think positively or negatively about a certain subject. This information could then be used to help improve products and cater more towards customer's needs and wants. NLP could also be used to analyze an article and be able to find other articles that are related to it. Furthermore, NLP can be used to discover patterns in the way that news outlets report information. With fake news being a hot topic today, NLP is becoming more and more relevant to developers who want to fight back against false information.

What tools are developers using to work with natural language processing? One way that a developer can work with NLP is through Python. Python is a high-level programming language that can be used for just about anything. With Python, it is possible to import a library called the Natural Language Toolkit or NLTK (2). With Python and NLTK, it is possible to program a NLP system to analyze text and produce desirable output. Another Python library that is used for NLP is spaCy, but which library should a developer use? NLTK provides nine different stemming libraries which allows developers to finely tune their models (3). A stemming algorithm tries to remove suffixes from words in order to find the “root word” or stem of a given word (4). SpaCy on the other hand implements one single stemmer (3). This stemmer is maintained and updated by the developers of spaCy. So with NLTK, the developer is left to be creative in developing the best stemmer algorithm by having a choice of nine different libraries. Whereas with spaCy, the developer is given one stemmer algorithm is being constantly improved upon by its creators. This means that a user of spaCy could update to the latest version of spaCy and find that their application has boosted because of it without having to do any extra work. However, there is a downside of using spaCy in that the developer may have to rewrite test cases as spaCy gets updated. Another difference between NLTK and spaCy is that NLTK does all of it processing with strings (3). Its inputs are strings and its outputs are strings. Contrastly, spaCy implements an object-oriented approach. When spaCy parses text, a document object is returned whose words and sentences are also represented as objects. With each object, there is also a number of attributes and methods that can be attached. Really, NLTK versus spaCy is a matter of strings versus objects. As far as performance between the two libraries, here is a graph showing the timings for word tokenization, sentence tokenization, and part-of-speech tagging for NLTK and spaCy.



(3)

NLTK out performs spaCy only in sentence tokenization, but this is most likely the result of different approaches. NLTK simply silts the text into sentences whereas spaCy creates a whole syntactic tree for each sentence (3). SpaCy's method may be slower, but it offers a way to reveal much more information about the text that is being analyzed. SpaCy seems like an obvious choice for a common-use application, but there is also one more thing to keep in mind. SpaCy is limited to English only. However, it is possible to convert the desired text to English through NLP in order to use other languages with spaCy, but it will require some creativity from the developer as this process is not built into spaCy.

What is an example of a NLP use case? As mentioned earlier, NLP is typically paired with machine learning algorithms. This is done because one large set of rules will not always return the most accurate information. With machine learning, a NLP algorithm can automatically learn rules and make statical inferences by reading text (5). A general rule is that the more data that is analyzed, the more accurate the model will be. Social media serves as a great use case of NLP. Companies track what people say online and they can do this through Twitter and Facebook for example. Specifically, if a company wanted to track Tweets to see all the mentions of their brand on Twitter, then the algorithm should start by retrieving each tweet that mentions the brand. Then each tweet should be processed through a sentiment analysis algorithm that outputs a sentiment rating.

In conclusion, natural language processing is a very complex topic in computer science, but if mastered, it provides a path to improving the intelligence of computers to levels that compare to human intelligence. Natural language is just any language that humans use to communicate. Any system that processes natural language is a natural language processing system. Some use cases of NLP are: translation, language modeling, parsing, text classification, and part-of-speech tagging. Computer scientists are interested in NLP because with the creation of the Internet, there is endless amounts of data in the form of natural language that can tapped into with a NLP algorithm. The two best NLP libraries for Python are NLTK and spaCy. The difference between the two comes down to strings versus objects. NLP is often paired with machine learning and a relevant use case of NLP is social media monitoring. Companies monitor social media in order to figure out what people are talking about when it comes to their products and brand name. In the end, natural language processing has many uses and the more that it is researched and developed, the smarter and more intelligent computers will become.

Sources

1. http://blog.algorithmia.com/introduction-natural-language-processing-nlp/

John Rehling, an NLP expert at Meltwater Group, adapted from *How Natural Language Processing Helps Uncover Social Media Sentiment.*

2. https://en.wikipedia.org/wiki/Natural_Language_Toolkit

3. http://blog.thedataincubator.com/2016/04/nltk-vs-spacy-natural-language-processing-in-python/

4. https://pypi.python.org/pypi/stemming/1.0

5. http://blog.algorithmia.com/introduction-natural-language-processing-nlp/