TECHNOLOGY REVIEW

BY: ROHAN KHANNA

The Google Cloud NLP tool is a natural language processing tool that uses various machine learning techniques to provide insights on unstructured text. Cloud Natural processing tools give users the ability to integrate text analysis tools in their applications with all the heavyweight processing done on the cloud. This review will walk through some of the key features of the tool along with a comparison with an open-source NLP tool kit, Stanford NLP

Google platform gives users the ability to do content classification, syntactic analysis, entity analysis, sentiment analysis, and entity sentiment analysis. It offers these services through two main tools: the AutoML natural language and the Natural language API. The Natural language API is meant for quick analysis as it uses thousands of pre-trained classifications. It categorizes text based on existing categories. The AutoML natural language gives you the ability to customize your nlp experience. For instance, you can classify text to your own domain-specific keywords and hence AutoML is the way to go if a user is looking for domain-specific NLP tools. AutoML, unlike Natural language API, also provides support for large datasets. It enables support for 5000 extra custom label classifications, 1 million documents, and a 10MB document size. Given these extra features, AutoML is more costly to use. Google Cloud NLP does a great job of providing text analysis in a number of languages. Although the content classification is only available for the English language, the other forms of analysis are available for multiple languages. For instance, Syntactic and entity analysis are available for 12 different languages. Google Cloud NLP also provides analysis for the Chinese language which traditionally has been a roadblock for many NLP Tools.

Google recommends using its client libraries for using the Natural Language API. A useful feature of the API is that it supports multiple programming languages. It has client libraries for C#, Java, Go, Node.js, PHP, Python, and Ruby. Google has done a great job of making it extremely easy to install and set up these libraries for each of these languages. For example, if you are using node.js you can easily use npm to install the client library. Similarly, for each of the other languages, a one line code execution will download the library in your environment. The documentation provided by google is immense. Each of these libraries has an in-depth API documentation that guides you on how to use the libraries. Content to this Natural Language API is provided as a text string. It is then internally processed as tokens and Google has limits on how long these strings and tokens can be. Currently, the maximum text size supported is 1,000,000 bytes with a token quota of 100,000 tokens. In addition, the API can handle 600 requests per minute and 800,000 requests per day. The API is a REST API and uses

JSON requests and responses. A Key point of google's NLP offering is the ease of use of this API. Google has done a fantastic job with encapsulating the complex NLP algorithms in easy to use functions for the users.
As an example of this :
A user can easily go about creating a client, inputting the text and the format of the text, and using the WriteSentiment function to get an in-depth sentiment analysis of the text provided.

```
var client = LanguageServiceClient.Create();
var response = client.AnalyzeSentiment(new Document()
{
    Content = text,
    Type = Document.Types.Type.PlainText
});
 WriteSentiment(response.DocumentSentiment, response.Sentences);
```

An extremely useful feature of the REST API is the ability to perform multiple operations in a single request. Natural language API can also integrate with Google cloud and provide such analysis on files stored in google cloud making it a great ecosystem for text analysis. The API also automatically detects the language for the text if not specified by the caller. Overall, this API is very useful for users looking to get quick insights on their text data and users who do not require much text customization.

The AutoML offering by google pertains to the crowd of users wanting the ability to customize their text analysis. It allows users to create custom machine learning models that will fit well for the user's specific data needs. The AutoML offers custom classification, entity extraction, and sentiment analysis models. AutoML has an easy to use User interface to create datasets to train the machine learning model on. AutoML uses 80% of documents in datasets for training, 10% for validation, and the remaining 10% for testing. A cool feature is that Users have the ability to specify which documents to use for training. After creating a dataset, users can provide their training model either through code or through the web user interface. The users can use the REST API or code using the client libraries offered for Python, Java, Go, Node.js, PHP, and Ruby. AutoML uses precision and recall to measure how well the model is capturing information.
Precision indicates how many of the assigned documents were actually supposed to be assigned to the entity they were assigned. Recall indicates, from all the documents that should have been identified as a particular entity or label, how many were supposed to

be assigned to that entity or label. As with the Natural language API, Google has done a great job with the documentation. They have provided detailed tutorials on how to set up and use auto ml and hence overall, the entire Google Cloud NLP tool is excellent for beginners.

The Stanford NLP tool is another popular tool that helps with generating insights from text data. In comparison to google's offerings, this tool does not have the same level of documentation and hence may not be suitable for complete beginners. In addition, The software offered in this toolkit is primarily java-based although there have been extensions in other languages such as python and ruby. Moreover, the traditional Stanford NLP toolkit, unlike Google's offering, is designed to work on the English language and apart from Stanford CoreNLP, it does not have support for many other languages. The source code of this package is open source, unlike googles which are great for people looking to contribute code to NLP toolkits. This toolkit is also great for users who would like to deep-dive into the specifics of the algorithms behind text information systems offerings like sentiment analysis algorithms. The Google NLP kit is great if you are just looking to quickly use these algorithms on your text. However, if you want to go into the specifics and create your own algorithms then Stanford NLP is the way to go. The Toolkits offerings include tokenization, part-of-speech tagging, named entity recognition, parsing, and coreference. The amazing thing about Stanford NLP is the freedom it gives a user. A user can use the Toolkit to both train and implement their own complex text processing algorithms using features like tokenization and POS tagging or use pre-trained models provided as part of the stanza offering under Stanford NLP. Also, Stanford NLP is free if not used for commercial purposes, unlike Google's offerings which charge based on the number of requests and data you use.

In conclusion, Google's offerings are definitely much better for beginners than Stanford's offerings for beginners because of their vast documentation and ease of use. Also, they are extensible to many use cases because of the number of languages and programming languages they support. In addition, through the use of its cloud google provides a great ecosystem for users looking to generate good insights from their texts. With all of google's services having high reliability and availability, it is a great choice for companies to use this tool. Stanford NLP may not be ideal for a beginner looking to generate quick insights from his/her text but is great if used by the beginner to learn and implement the fundamentals of text processing algorithms. The customizations it allows makes it a great tool to use for research and in industry. It gives users a chance to deep-dive into the specifics of the code, unlike Google's offerings which generally tend to give a high-level view of the internals. Moreover, since its open-source people can

make contributions to the package and continuously work to improve it. Overall, It would be recommended for people who are some prior experience with NLP.

REFERENCES:

https://blog.api.rakuten.net/top-10-best-natural-language-apis/
https://cloud.google.com/natural-language/docs/basics
https://nlp.stanford.edu/software/
https://cloud.google.com/automl