Automatic Text Analysis Using Drupal

By Herman Chai

Computer Engineering

California Polytechnic State University, San Luis Obispo

Advised by Dr. Foaad Khosmood

June 14, 2013

Abstract

Natural language processing (NLP) is a field of computer science that is concerned with the interpretation of human language by computers. NLP has a multitude of different applications in the fields of computer science, artificial intelligence, and linguistics. The Automatic Text Analysis Using Drupal project is intended to increase the availability of NLP tools that can be easily used by the general public. By integrating a front-end content management system like Drupal with different back-end NLP applications, users can receive automatic analysis of their text without knowledge of the system.

Contents

1 Introduction

1.1 Automatic Text Analysis Using Drupal

The Automatic Text Analysis Using Drupal project is intended to increase the availability of NLP tools that can be easily used by the general public. By integrating a front-end content management system like Drupal with different back-end NLP applications, users can receive automatic analysis of their text without knowledge of the system. Using Drupal's module system, the Python Natural Language Toolkit (NLTK) and the Stanford Parser can be integrated and used in order to provide statistical analysis to user-supplied text.

2 Background

2.1 Drupal

Drupal is an open source content management platform used in websites and applications. A standard installation of Drupal, known as Drupal core, contains basic features that include user account registration, menu management, system administration, and other features common to a content management platform. Using add-ons, known as modules, a user can modify the standard Drupal installation and add different features or themes to their website.

The Drupal system is built upon the open source Web development platform known as a LAMP stack (Figure 1). LAMP is short for Linux, Apache, MySQL, and PHP. In this case, Drupal uses PHP to interface with a MySQL database that is hosted on an Apache server running on Linux. By combining the open source software a developer is able to create dynamic Web sites or other Web applications.

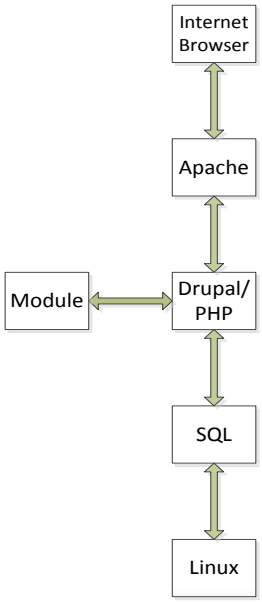

Figure 1: Basic Drupal Architecture

Drupal is built and supported by an active community that contributes thousands of different themes and modules. These modules are developed using the PHP scripting language that is suited for web development. Drupal's module system is based on the concept of "hooks" which allow modules to interact with the Drupal core. A hook is a PHP function that has a defined set of parameters, a specified result type, and once implemented performs a specific action. To extend on the Drupal core, a module simply needs to implement any of the available hooks that are described in the Drupal developer documentation.

2.2 Natural Language Processing

Natural language processing (NLP) is a field of computer science that is concerned with the interpretation of human language by computers. NLP has a multitude of different applications in the fields of computer science, artificial intelligence, and linguistics. Many of these applications require the use of statistics in characterizing text. These statistics can be obtained using NLP applications like the Python Natural Language Toolkit (NLTK) and the Stanford Parser.

The NTLK is a package that can be downloaded and used in Python programs in order to work with human language data. It contains libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. By using the NLTK, developers can characterize text in terms of statistics and from there use those statistics in other NLP applications.

The Stanford Parser is a package that contains Java implementations of different natural language parsers. Parsers are NLP tools that generate parse trees of user-supplied text. A parse tree is an ordered tree that represents the grammatical structure of a sentence. The tree describes which groups of words go together and the subjects and objects of the sentence.

2.3 Features Selection and Machine Learning

Machine learning is the construction of systems that can learn from or be trained with data. In relation to NLP, a machine learning system could be trained for a multitude of different applications. These could include authorship attribution which is the process of determining an author of a work based upon characteristics found in other works, speech recognition which is the adaptation of spoken word into text, or any number of other applications.

Machine learning uses a process known as feature selection in order to select relevant features for the construction of the machine learning model and system. In this particular project, one of the parsers that can be found in the Stanford Parser package was trained with data known as the Penn Treebank. Training a parser with the Penn Treebank allows it to learn a collection of syntactically annotated data and construct a machine learning model. The parser would then be able to create parse trees of sentences based on the data it was trained with.

The statistics obtained from this project could potentially be used as part of the feature selection process in an NLP application. The statistics could be used as characteristics for different works and used to train a machine learning system.

3 Project Description and Architecture

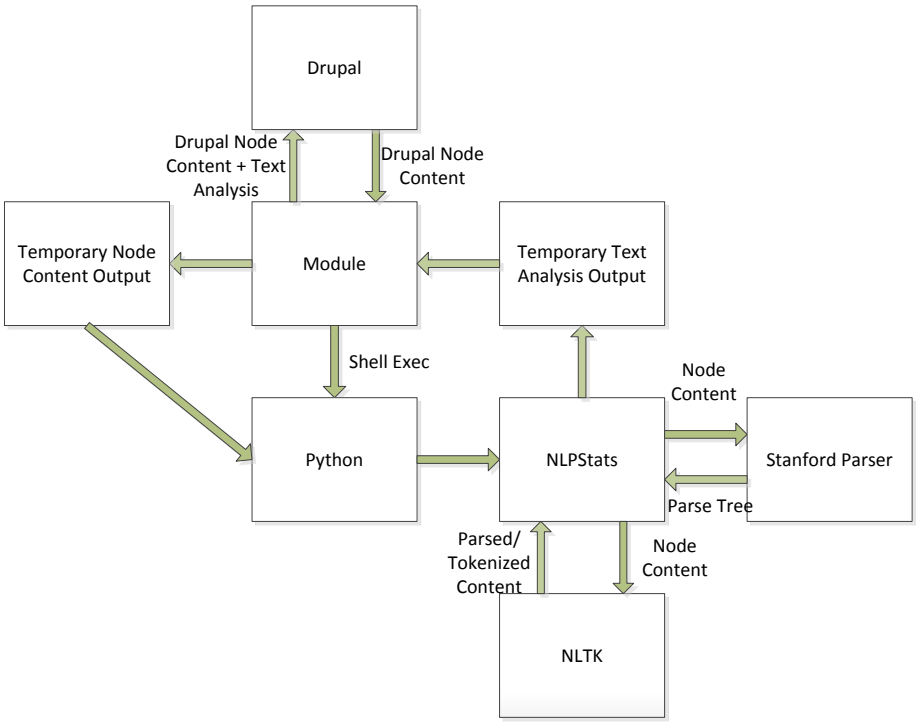

Figure 2: Module Architecture

The module developed for this project initially did all of its text analysis directly, however as development progressed the text analysis was outsourced to the NLPStats Python program in order to make use of the NLTK. The general architecture for the module (Figure 2) is relatively simple with the majority of the functionality being completed by the NLPStats program. The module implements hooks to pull the node content from the database then strips the HTML tags from the content and outputs it to a temporary text file. The module then calls the PHP shell_exec command to execute Python and run the NLPStats program. The NLPStats program takes in the temporary text file and other arguments and performs statistical analysis on the text using the NLTK. During this process it also passes the content of the text to the Stanford Parser which generates a parse tree of each sentence in the text. The results of the Stanford Parser are then returned to NLPStats and compiled into the rest of the results of the analysis. The results of the analysis are then output to a temporary text file that is read by the module. The module then returns the node content along with the results of the analysis back to Drupal to be displayed on the Web site.

4 Results

4.1 Data

In its current state the project is able to analyze and output approximately 17 different features of the text, however there are several other features that could still be analyzed, many of which could be obtained by analyzed the parse trees generated by the Stanford Parser. Currently the

parse trees are only used for one of the features that are output to the user. The features can be grouped into three different types (Figure 3) that include the lengths of the text in terms of number of characters, words, syllables, etc., data obtained from analyzing the parse trees, and calculated readability measures using the lengths of the text.

Type	Feature
Lengths	Characters (Tot, para, sent, word)
Lengths	Capitalized Words (Tot, para, sent, word)
Lengths	Type/Token Ratio
Lengths	Word (Tot, para, sent)
Lengths	Sentences (Tot, para)
Lengths	Paragraphs
Lengths	Syllables (Tot, para, sent, word)
Lengths	Numerics (Tot, para, sent, word)
Lengths	Vowels (Tot, para, sent, word)
Lengths	Punctuation (Tot, para, sent, word)
Parse	Nodes (sent)
Readability	Automated Readability Index
Readability	Coleman Liau Index
Readability	Flesch Reading Ease
Readability	Flesch-Kinkaid Grade
Readability	Gunning Fog Index
Readability	Lix Formula

Figure 3: Feature List by Type

4.2 Sample Run

To understand the usefulness and potential application of this project, we will compare two similar texts, John. F. Kennedy's inaugural address and Martin Luther King Jr.'s "I Have a Dream" speech. These works serve as good examples for comparison since they are both speeches and were delivered in the same time period of the 1960's. They are also of similar length with JFK's speech having a length of approximately 1,300 words and the "I Have a Dream" speech having a length of approximately 1,600 words. Also the content of the speeches although not quite the same are similar in some ways.

Since both of these texts were meant to be spoken and not read, it is not surprising that there is little variation in many of the features like characters per word, syllables per word, and vowels per word. There is however a clear difference when looking at words per paragraph (Figure 4) and syllables per sentence (Figure 5) as well as certain readability measures. JFK's speech has fewer words per paragraph with an average of (49.25) and a max of (96) while MLK's has an average of 55.73 and a max of 206. JFK's speech however has more syllables per sentence with an average of 35.18 and a max of 112 while MLK's has an average of 27.11 and a max of 97. JFK's speech has a Flesch-Kincaid grade of 10.746 while MLK's has a grade of 8.659. This shows a clear difference in the structure of the speeches. JFK has longer sentences containing more words however MLK has longer paragraphs containing more sentences and this can be seen just by looking at the two speeches. This also explains the difference in readability grade as

JFK's speech has more syllables per sentence those sentences will tend to be more complex. By using this data we can characterize different works and use those characterizations for different applications like authorship attribution.

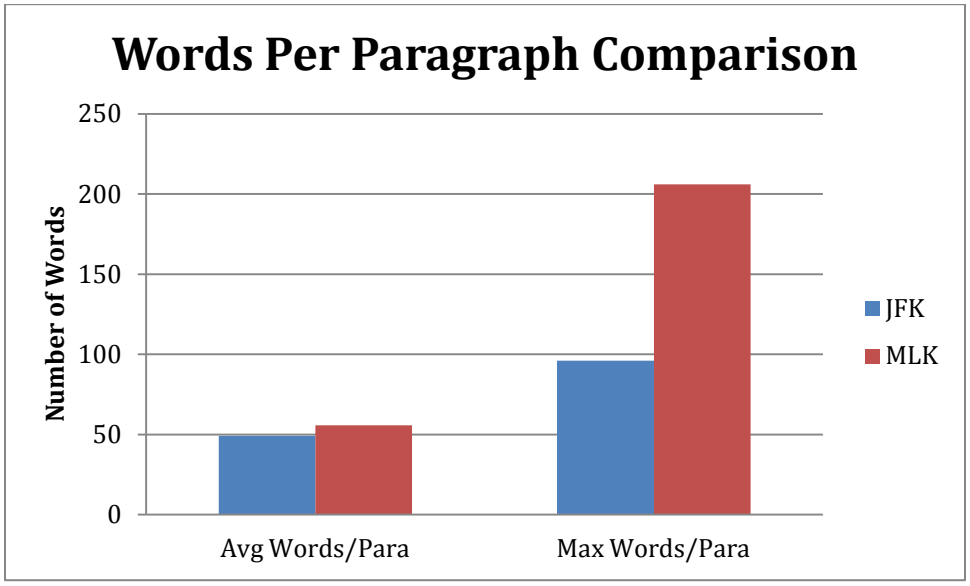

Figure 4: Words per Paragraph Comparison between the two speeches

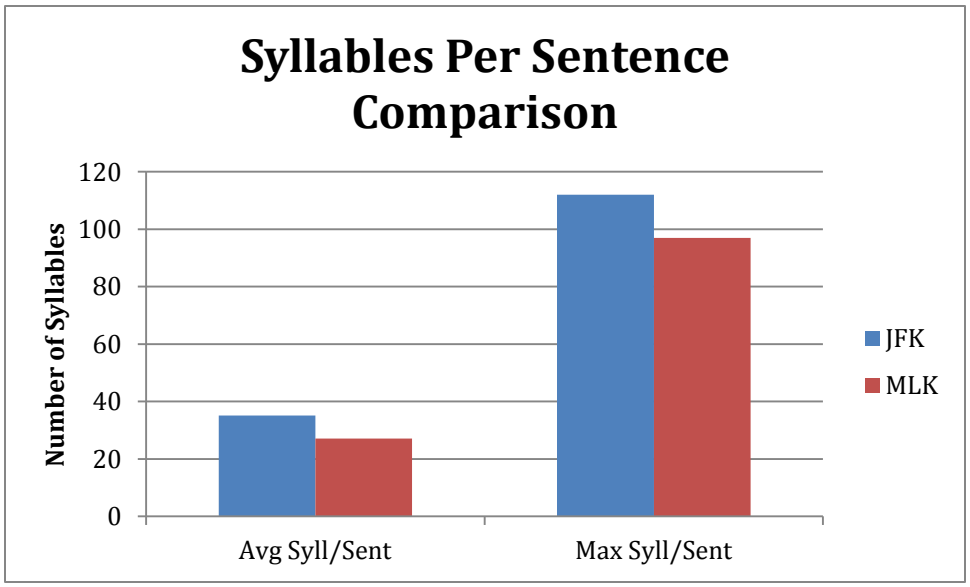

Figure 5: Syllables per Sentence Comparison between the two speeches

5 Conclusion

The Automatic Text Analysis Using Drupal system is an easy to use NLP tool that can quickly gather the statistical features of a document. This type of tool could be used in multiple applications such as authorship attribution and other NLP related machine learning systems.

With the development of this system I have learned to be able to adapt quickly in terms of using other programming languages and contributing to other programs. As a part of that is also learning to adapt to user expectations and requirements that can change as development progresses. I did not realize how important the field of NLP was to our current technologies as well as future technologies and I hope to be able to contribute to its growth with this project and continue to see it grow in the future.