CSC Senior Project: NLPStats

By Michael Mease

Cal Poly San Luis Obispo

Advised by Dr. Foaad Khosmood

March 16, 2013

Abstract

Natural Language Processing has recently increased in popularity. The field of authorship analysis, specifically, uses various characteristics of text quantified by markers. NLPStats serves as a tool designed to streamline marker extraction based on user needs. A flexible query system allows for custom marker requests, adjustment of result formatting, and preprocessing options. Furthermore, an efficiently designed structure ensures that users retrieve information quickly. As a whole, NLPStats enables anyone, regardless of NLP experience, to extract important information about the text of a document.

Contents

1 Introduction and Background **1**
1.1 Background of Natural Language Processing 1
1.2 Introduction to NLPStats 1

2 Markers **2**
2.1 What are Markers? 2
2.2 Markers used in NLPStats 3
2.2.1 Lengths Script 3
2.2.2 Phrases Script 3
2.2.3 Readability Script 3
2.3 Extracting Markers 3

3 Marker Analysis System **4**
3.1 Overview 4
3.2 The Query File 5
3.3 Preprocessing 6
3.4 The Results File 7

4 Example Use **7**
4.1 Comparison 7
4.1.1 Lengths 7
4.1.2 Phrasal Markers 8
4.1.3 Readability Measures 9
4.2 Conclusion 9

5 Implementation Options **9**

6 Conclusion **10**

References **11**

A Appendix **12**
A.1 Marker List by Script 12
A.2 Example Query File 13
A.3 Results File Excerpt 14

1 Introduction and Background

1.1 Background of Natural Language Processing

Natural language processing (abbreviated NLP) is a field of computing dealing with the automatic interpretation and generation of language. With recent developments in the processing power of computers, natural language processing has increased in popularity. [4] Some of the main challenges of NLP include:

Authorship Attribution - Determining the author of a document based upon style.
Speech Recognition - Converting spoken words into text.
Summarization - Compressing a document's text into a representative summary.
Sentiment Analysis - Determining expressions of emotion based on the semantics of text.

Each of these tasks requires a unique approach to NLP. For example, summarization can rely on paraphrasing sections of a document based on key phrases or words. Authorship attribution, on the other hand, relies more heavily on textual analysis and "style-markers" which are generally believed to correspond to a unique author. The variety of tools available for accomplishing these NLP tasks are great in number, but generally have broad purposes. The natural language toolkit (NLTK), for instance, has a large set of features for everything from tokenization to classification. [8] Online tools are also available, including publicly available implementations of the widely popular Stanford parser which generates parse trees of user-supplied text. Another, more relevant tool to this project is JGAAP, the Java Graphical Authorship Attribution Program. [2] JGAAP specifically targets the field of authorship attribution by applying user selected marker analysis to supplied documents with authors labelled by the user. Instead of reporting statistics as NLPStats does, JGAAP places a focus on statistical analysis. These tools, among many others, have helped progress the field of NLP greatly in recent years.

1.2 Introduction to NLPStats

For this project (NLPStats) we intend to further increase this variety of modern NLP tools with a focus specifically on authorship analysis. This project is implemented primarily in Python using NLTK as a primary tool for document processing and analyzing "style-markers". As mentioned before, the analysis of "style-markers" (further discussed in Section 2) is popular in authorship attribution. Authorship attribution itself, however, is not the only task that utilizes these markers: plagiarism detection, authorship verification, and author profiling can all rely on marker analysis. [6] Because of the importance of markers to these various tasks, we want to create a tool that automatically extracts these markers based upon user needs. Traditionally, someone who wanted to extract markers from text would need to design algorithms to detect and analyze each individual marker. NLPStats removes this overhead and allows those interested in authorship analysis to quickly and

accurately retrieve statistics about a given document or set of documents.

The primary focus of this project is to design a fully modular system that supports the extraction of a wide variety of NLP markers. The system also must present these results in both human and computer readable formats so that further processing can be done either manually or through automatic processing of the results file. Ideally no knowledge of NLP will be necessary to utilize NLPStats and retrieve statistics describing a document's characteristics.

The use of NLPStats streamlines many NLP tasks by categorizing related markers into "scripts". These scripts can be queried individually to retrieve only those markers that a user cares about. In addition, for those seeking more than the base feature set, the modular nature of this tool allows users to create new scripts containing additional extraction methods and markers. The extendability of NLPStats, thus, is high in that each script is independent and new scripts can be added at will. Further details of the NLPStats design are discussed in Section 3.

2 Markers

2.1 What are Markers?

As mentioned in Section 1, markers serve an important purpose in NLP, but what exactly are they? Markers are features of text that describe the text in some way. For example, the length of a document in characters and the number of words are both simple instances of a marker. These types of markers, specifically, are both grouped in what is called "lexical" markers. Markers can also fall under the syntax or semantics categories. [9] Table 1 outlines several different types of markers and some examples of each type.

Marker Type	**Description**	**Examples**
Syntactic Markers	Syntax-related properties	Parts of speech, phrase structure
Semantic Markers	Meaning of words	Synonyms, word dependencies
Lexical Markers	Properties of word text itself	Average word length, number of syllables

Table 1: Outlines various types of markers used in authorship analysis.

Essentially, markers characterize text and can be thought of as statistics of the text. In authorship attribution, for instance, these markers are considered to quantify an author's style. The overall quantification of a large set of markers enables a person to attribute a document to a most likely author. For example, consider a marker tracker the average number of syllables per sentence. Author A submits three documents, each of which has 15 syllables per sentence. Author B submits three documents as well, but each of these has nearly 25 syllables per sentence. If an unseen document by one of the two authors

contains 23 syllables per sentence, it is more likely to be author B since this author uses more syllables in his writing. Now imagine using hundreds or thousands of markers in combination and generating a net distance between an author's profile and an unseen document. This is just one example of how markers can be used in NLP.

2.2 Markers used in NLPStats

In NLPStats, markers are only analyzed – the application of the results is left up to the user. Currently, three sets of markers have been included in the "script" system: lengths, phrases, and readability. Each script contains one set of related markers that can be independently requested by a user.

2.2.1 Lengths Script

This script contains most of the length-related markers that fall under the aforementioned lexical marker category. The lengths script includes vowel counts, punctuation counts, word lengths, syllable counts, and capitalized words. Each individual length statistic is replicated on a per paragraph, per sentence, and, if applicable, per word basis. Appendix A contains a full list of these markers along with those mentioned in the following sections.

2.2.2 Phrases Script

This script contains phrase count ratios (ex: prepositional phrases:verb phrases) and counts of words in specific types of phrases. Part-of-speech tagging along with phrasal analysis is required to obtain the statistics in this script. Since using the Stanford parser would require Java, a shallow parser implemented in the pattern.en module was used here. [3] In order to obtain more reliable statistics, a separate script could be created in the future that uses an interface to the Stanford parser.

2.2.3 Readability Script

This script contains readability scores for various readability measures, such as the popular Flesch-Kinkaid Grade. [1] These measures generally rely on ratios between syllables, words, characters, or sentences.

2.3 Extracting Markers

Before any scripts are called, a document's text is organized into nested lists. The inner list contains string elements, with each string being a single sentence. This list represents a single paragraph, and the other list represents the entire document as a list of paragraphs. This structure is passed to a script along with a set of "quantifiers" which will be discussed later. Since this process only needs to be completed once, a user can gather many statistics

in a single script without preprocessing a document more than once.

Once a script is called, the implementation of the script itself determines how markers are extracted. As an example, consider the lengths script described above. In this script, numerous lengths such as word length and syllable counts are requested by a user. In order to track these markers, the text is processed sentence by sentence. The words are first tokenized by NLTK's word tokenizer, then each token is processed and placed into a list created solely for a single marker. In this list, the statistic itself is stored rather than the word. Figure 1 helps to clarify this process.

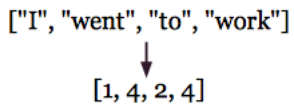

Figure 1: Visualization of characters per word in length script.

Once each of the relevant lists are filled with statistics, the minimum, maximum, average, median, and variance of each marker can be calculated. In addition, since these lists maintain the structure of the original sentence list, the per paragraph and per sentence values can also be calculated and displayed to the user. This system allows for a wide range of statistics, from the variance of the number of syllables per sentence to the shortest paragraph in words or characters. It is this versatility that allows a serious researcher to obtain exactly the statistics they need from a specific document with very little effort.

The general process for any script is to process text sentence by sentence (tokenizing if needed) and gather relevant counts while filling new lists with the appropriate marker values. Then, the average, minimum, maximum, etc. values are calculated and stored in a string. Finally, if appropriate, the list containing these values is appended to the aggregate values for the user. In this case, the majority of the length statistics do append a list of values to each marker report.

3 Marker Analysis System

3.1 Overview

NLPStats, in brief terms, processes a query file, calls a script for each query, then writes script results to a file. The overall system is separated into the core preprocessing module and the scripts. The core module processes three files: a query file, a document file, and a results file. The query file contains information about what markers the user wants reported in the results file, while the document file contains the document to be analyzed. Once each file is validated (they exist) the core preprocesses the document file and tokenizes it into

sentences and paragraph lists as described in Section 2. The query file is then processed line by line with each line being a script call. The line "lengths 0012 10110" would call the "lengths" script in a specific way (the details of these query lines will be outlined later.) The length script runs and the results are stored. Once all queries finish, the system writes to the results file and then closes all files. Figure 2 outlines the architecture as a whole.

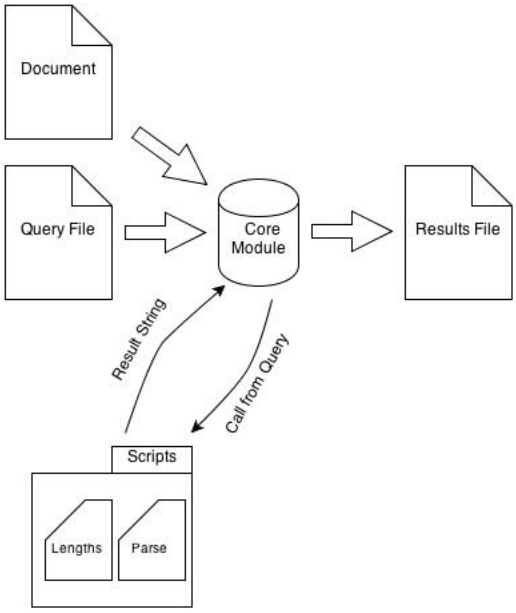

Figure 2: Architecture of NLPStats system.

3.2 The Query File

Queries allow a user to specify which sets of markers are placed into the results file. An example query is shown below in Figure 3. A single query requires three inputs by a user delimited by whitespace on a single line: the name of the script, the parameters, and the quantifiers. Each of these are explained below and used as described in Foaad Khosmood's system for analyzing and classifying language styles. [5]

(1)	(2)	(3)
lengths	0104	10010
read	1003	01101

Figure 3: An example query.

Here, the first query corresponds to the request for the "lengths" script run on a document with: stopwords removed, results printed with four digits past the decimal, and only the minimum and median of each marker statistic. Below, each section of a query is described.

(1) Name of script: The name of the script for analyzing a set of markers. The available scripts as of now are "lengths", "read", and "phrase". See Appendix A for a list of markers within each script.

(2) Parameters: The parameters used in preprocessing. Parameters are options a user can select to preprocess a file or determine output format. Parameters include (in order): "ignore capitalization", "remove stopwords", "stem words", and decimal precision. A parameter token has four numbers representing each of these parameters (ex: "0015"). The first three are binary and activated by placing a "1" in the parameters respective location. To only ignore capitalization, the first three numbers of this token would be "100". A "0" indicates that a user does not want a parameter activated. The fourth parameter is a 0-9 value determining how many digits to include after a decimal in reported marker statistics. A "3" here, for example, would produce the numbers 3.391 and 71.442 while a "1" would produce 3.3 and 71.4.

(3) Quantifiers: The types of values to report (min, max, etc.) These values are calculated for every statistic in the called script. Quantifiers include (in order): minimum, maximum, average, median, and variance. Similarly to parameters, each of these values can be requested with a "1" or ignored with a "0" in the query token. Any quantifier not requested with report "N/A" rather than the number in the results file.

Query files can contain infinitely many queries with each line being a single query. This allows, for example, the comparison of a document with stopwords to the same document without stopwords in single run of NLPStats.

3.3 Preprocessing

Preprocessing consists of reading query inputs and determining how to parse a document. Parameters, as mentioned in the query section above, are handled in the core module of NLPStats. Preprocessing is currently done once for each query since every query can have a different combination of parameters. Analyzing the query file for overlapping parameter combinations would improve efficiency since the preprocessed document list could be passed to each relevant script.

For each activated parameter, the words of a document are converted as appropriate. NLTK provides useful tools for stemming and removing stopwords. Several algorithms from the NLTK cookbook (by Jacob Perkins) were utilized here and in the scripts portion

of this project. [7]

3.4 The Results File

Results are printed in a structured format that is human readable. Initially, every statistic was grouped by quantifier and printed as CSV's. This required too much effort by the user to determine which statistics to look for, so explicit marker names are now provided. A single results entry (one marker) contains: a name, a series of quantifier values, and print out of the list of marker values by word, sentence, paragraph, or document. Figure 4 serves as an example for the syllables per sentence marker.

```
Syllables/Sentence - min: 0.00  max: 141.00  med: 35.00  ave: 43.66  var:
1601.38
[136, 79, 41, 19, 40, 19, 18, 41, 21, 17, 39, 16, 40, 21, 141, 42, 81, 20,
1, 133, 10, 26, 35, 66, 0, 19, 1, 62, 41, 19, 2, 15, 61, 1, 20, 36, 73, 61,
139, 1, 80, 41, 1, 3, 15, 140, 0, 134, 20, 19, 83, 20, 21, 21, 43, 62, 11,
129, 79, 42, 20, 91, 41, 25]
```

Figure 4: An example result entry from a result file.

Here, we can see each quantifier delimited by two spaces. The list below the quantifiers, denoted by the "[" symbol, contains the syllable count for each sentence in the document separated by commas. If the result marker were syllables per paragraph, this list would contain counts per paragraph rather than per sentence. Some lists, such as the per document markers, only contain one number since only a single document is being analyzed.

4 Example Use

To better understand the usefulness of NLPStats, we will compare two segments of text from Shakespeare: Act III of Macbeth and Sonnets 1-20. Macbeth is a famous play by Shakespeare and contains characters who speak in both blank verse and prose. [10] The sonnets, on the other hand, are highly structured sonnets of fourteen lines each. Each sonnet is treated as a paragraph in this example. Both Macbeth Act III and Sonnets 1-20 contain roughly 3,000 words and serve as good examples for comparison since they are written by the same author in different genres of writing.

4.1 Comparison

4.1.1 Lengths

Since the text of each sonnet is highly structured, it is not surprising that there is little variation in characters per paragraph, words per paragraph, and syllables per paragraph.

Macbeth, however, contains less syllables per paragraph (32.5) but a much greater range (1-263). Each sonnet has an average of 3.2 sentences and 140 syllables, higher than Macbeth by over a factor of four. The average paragraph length, however, is also roughly four times greater in the sonnets than in Macbeth, so this is not surprising.

The number of syllables per paragraph in the sonnets only spans a range of 24, again due to the consistency of the writing. The punctuation per word on average is .17 for the sonnets and .19 for Macbeth, showing a very consistent usage in both works.

4.1.2 Phrasal Markers

The adjectives per word in the sonnets is 60 percent higher in the sonnets than in Macbeth, which probably indicates that the sonnets are more descriptive and colorful. The sonnets also have higher noun usage per word. Macbeth uses many more proper nouns since the characters are referenced constantly. Other phrasal statistics, including verb, verb phrase, prepositional phrase, and noun phrases are nearly identical, however, showing some consistency between the two genres. One important thing to note is that the number of words used in various phrase types is consistently a factor of four greater in the sonnets in comparison to Macbeth (due to paragraph length being four times longer.) Figure 5 depicts these ratios between the two works.

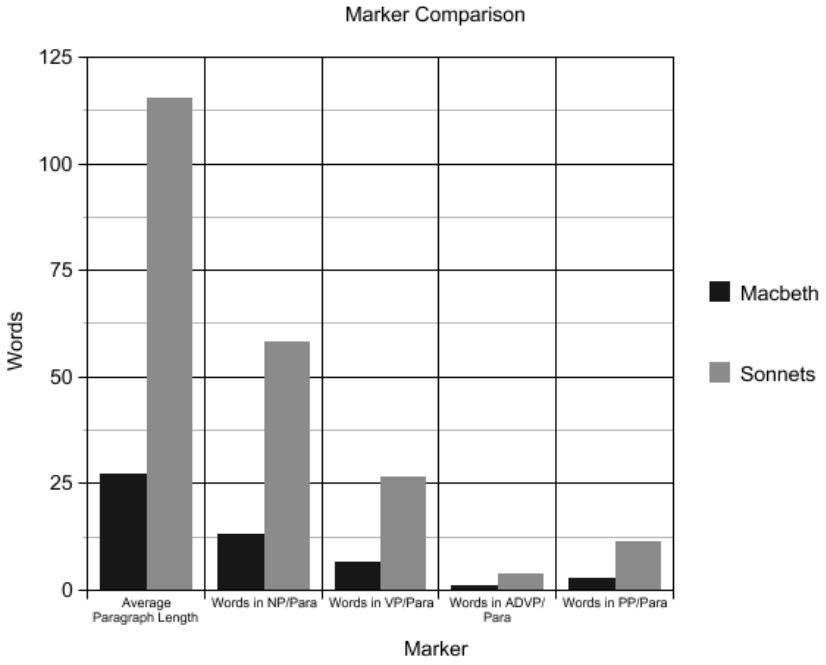

Figure 5: Graph displaying the ratios of words used in certain phrase types.

4.1.3 Readability Measures

Given a set of seven reliable readability measures, the two texts varied greatly. Macbeth, on average, was deemed appropriate for a sixth grader while the sonnets rated closer to early college level writing. Macbeth, being a play, is most likely designed to be seen and read by a large number of people. Given that eighth grade is currently the average reading level of a US citizen, Shakespeare seems to have done a good job appealing to large audience. The sonnets, however, seem to be written for a more sophisticated audience and serve as an artistic expression of emotion and appreciation of beauty.

4.2 Conclusion

There seems to be some remnants of style in the two works: consistency of phrase types, syllables per word, and part-of-speech ratios. The seemingly genre-related markers, such as adjective usage and syllable variation, seem to indicate a trend in the structure of the writing rather than the author's decisions while writing. In essence, its important to see the key characteristics that distinguish two types of writing. The difference in adjective usage shows how genres influence descriptions, while proper noun usage in Macbeth shows just how much characters names are said in a play. Even on the surface, these statistics reveal characteristics of text that are not always obvious or require knowledge of an author beforehand. With NLPStats, all of this comes nearly for free.

5 Implementation Options

Currently, NLPStats is command line based and requires Python 2.7 with NLTK. The Pattern.en module is also required to run the "phrase" script. Running this program from the command line is useful for the technically savvy, but what about the less experienced user? Originally, NLPStats was going to be implemented as a module in Drupal, a widely used content-management system. Unfortunately, issues with NLTK prevented this effort from continuing, but this most likely occurred due to issues with the server being used rather than NLTK itself. In the future, NLPStats could be used as a filter in Drupal, and the statistics could be stored in relation to a specific document that has this filter applied. This would allow users to easily gather statistics about documents on their website without requiring much effort.

Moodle, a similar system (although not as widely used), could also be used as a front-end for NLPStats. The user base is smaller, but the application is greater since the documents on Moodle are related to academics. Both implementation options offer a more convenient interface than the command line approach, but also require web-hosting for use of the program.

6 Conclusion

NLPStats is an easy to use NLP tool for those interested in quickly gathering statistics about a document. The versatility and efficiency of the program ensures that a user gets exactly what they want from a query in a reasonable amount of time. Developing this tool taught me to see user expectations as a primary focus of a software development project. Not only is it important to establish requirements, but to continually work with users to ensure that they are getting what they want, expect, and need from a developer. In the future, I hope to continue focusing on users to develop better products for people with real problems that need real solutions.

References

[1] "The flesch reading ease readability formula." [Online]. Available: http://www.readabilityformulas.com/flesch-reading-ease-readability-formula.php

[2] "Jgaap wiki main page," 2013. [Online]. Available: evllabs.com/jgaap/w/index.php/Main_Page

[3] "Pattern.en module website," CLiPS, 2009. [Online]. Available: http://www.clips.ua.ac.be/pages/pattern-en

[4] K. S. Jones, "Natural language processing: a historical review," *Current issues in computational linguistics*, October 1994. [Online]. Available: http://www.cl.cam.ac.uk/archive/ksj21/histdw4.pdf

[5] F. Khosmood, "Computational style processing," December 2011.

[6] M. Kimler, "Using style markers for detecting plagiarism in natural language documents," 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.3936

[7] J. Perkins, *Python Text Processing with NLTK 2.0 Cookbook*. Packt Publishing, 2010.

[8] N. Project, "Nltk official website," 2013. [Online]. Available: http://nltk.org/

[9] E. Stamatatos, "A survey of modern authorship attribution methods," 2009. [Online]. Available: http://www.icsd.aegean.gr/lecturers/stamatatos/papers/survey.pdf

[10] S. E. Team, "Macbeth writing style," 2008. [Online]. Available: http://www.shmoop.com/macbeth/writing-style.html

A Appendix

A.1 Marker List by Script

Script	Marker
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)
Phrase	Adjectives:Word
Phrase	Adverbs:Word
Phrase	Non-JJ/Non-RB:Word
Phrase	Determiners:Word
Phrase	Nouns:Word
Phrase	Past-tense Verbs:Verbs
Phrase	Plural Nouns:Noun
Phrase	Pronouns:Word
Phrase	Proper Nouns:Noun
Phrase	Noun Phrases:Total Phrases (TP)
Phrase	Verb Phrases:TP
Phrase	Adverb Phrase:TP
Phrase	Preposition Phrase:TP
Phrase	Conjuctive Phrases:TP
Phrase	Words in Noun Phrases per Paragraph
Phrase	Words in Verb Phrases per Paragraph
Phrase	Words in Adverb Phrase per Paragraph
Phrase	Words in Prepositional Phrase per Paragraph
Phrase	Words in Conjunction Phrase per Paragraph

Read	Automated Readability Index
Read	Coleman Liau Index
Read	Flesch Reading Ease
Read	Flesch-Kinkaid Grade
Read	Gunning Fog Index
Read	Lix Formula

A.2 Example Query File

The following query file will generate two calls to the script "lengths" when supplied to NLPStats, gathering all statistics with a precision of 2. The first call includes stop words while the second strips them. The "phrase" line does not report on variation or averages for those statistics. Finally, the "read" line increases precision to 3 and only requires an average value to be returned.

```
lengths 0002 11111
lengths 0102 11111
phrase 0002 11010
read 0003 00010
```

A.3 Results File Excerpt

The results file excerpt shown in Figure 6 features a report of 10 of the markers in the "lengths" script.

```
Type/Token Ratio - min: 0.39  max: 0.39  med: 0.39  ave: 0.39  var: 0
[0.39]

Total Words - min: 2310  max: 2310  med: 2310  ave: 2310  var: 0
[2310]

Words/Paragraph - min: 102.00  max: 131.00  med: 118.00  ave: 115.50  var: 44.35
[110, 120, 118, 106, 107, 118, 102, 110, 122, 117, 118, 120, 113, 114, 111, 111,
131, 119, 120, 123]

Words/Sentence - min: 0.00  max: 120.00  med: 30.00  ave: 36.09  var: 1044.05
[110, 66, 34, 20, 35, 16, 16, 34, 17, 13, 31, 16, 31, 15, 107, 34, 64, 20, 1,
101, 8, 25, 30, 47, 0, 18, 1, 52, 36, 15, 2, 14, 50, 1, 18, 32, 66, 52, 120, 1,
63, 31, 1, 3, 14, 114, 0, 111, 15, 15, 64, 17, 19, 20, 34, 58, 9, 110, 69, 33,
18, 70, 31, 22]

Total Sentences - min: 64  max: 64  med: 64  ave: 64  var: 0
[64]

Sentences/Paragraph - min: 1.00  max: 6.00  med: 3.00  ave: 3.20  var: 2.76
[1, 3, 5, 5, 1, 3, 2, 5, 5, 6, 2, 1, 6, 2, 1, 4, 4, 2, 3, 3]

Total Paragraphs: min: 20  max: 20  med: 20  ave: 20  var: 0
[20]

Total Syllables - min: 2794.00  max: 2794.00  med: 2794.00  ave: 2794.00  var:
0.00
[2794]

Syllables/Paragraph - min: 133.00  max: 157.00  med: 140.00  ave: 139.70  var:
28.31
[136, 139, 139, 133, 141, 143, 134, 137, 142, 135, 134, 139, 141, 140, 134, 142,
147, 140, 141, 157]

Syllables/Sentence - min: 0.00  max: 141.00  med: 35.00  ave: 43.66  var:
1601.38
[136, 79, 41, 19, 40, 19, 18, 41, 21, 17, 39, 16, 40, 21, 141, 42, 81, 20, 1,
133, 10, 26, 35, 66, 0, 19, 1, 62, 41, 19, 2, 15, 61, 1, 20, 36, 73, 61, 139, 1,
80, 41, 1, 3, 15, 140, 0, 134, 20, 19, 83, 20, 21, 21, 43, 62, 11, 129, 79, 42,
20, 91, 41, 25]
```

Figure 6: Example result from a "lengths" script.