

Zipf Lock: Not Just a Good Idea - it's the Law!

assignment 1 - comp791a - winter 2003

some student (1234567)

17th February 2003

Abstract

In this 'paper' we discuss Zipf's Law theoretically, and attack it empirically, approaching it generally through a series of 7 varieties of experiments performed on 19 different electronic texts chosen from 8 different subject types using a PERL/UNIX software implementation and the diminishing free time of the Modern Undergraduate. The purpose of the software is to generate rank/frequency and frequency/frequency-of-frequency tables robustly and efficiently, and will be used to generate our experimental data. Time permitting, a musical ode to Zipf will also be composed for timpani and various wind instruments.

1 Introduction

In "*Human Behavior and the Principle of Least Effort*", George Kingsley Zipf pronounced his unifying theory of human behaviour as it related empirically to laws of natural language: the Principle of Least Effort. In general, this principle states that humans will typically act in such a fashion to minimize their expected effort. In natural language, this means that in a discourse between a speaker and a listener, the speaker will typically attempt to minimize his effort by using fewer and more frequent words (resulting in ambiguity for the listener, but a simpler vocal construction), and the listener will typically prefer to have minimized their effort by observing more and less frequent words (resulting in specificity for the listener, but a more complicated vocal construction).

Specifically, for each word type in a corpus, counting the frequency of this word and then sorting the resultant list of words by decreasing frequency will give us a ranking (hence every position in this list is called a 'rank' r) of words against their frequency f . Zipf's Law specifically states that given these variables $f \propto 1/r$, which means that for some constant ' a_{nk} ', $f \cdot r = a_{nk}$. For example, this means that the 75th most common word should occur about 7 times more frequently than the 525th most common word. Our task is to verify or deny the validity of Zipf's Law.

2 Software Implementation

Despite recent experience using Java for the purposes of NLP, the author chose the PERL interpreted programming language as the implementation language for these experiments, based primarily on the verbal and written suggestion of Dr. Leila Kosseim. Other relevant factors of the language that positively determined its use were the following:

- Robust regular expressions: For the purposes of token extraction, case normalization, punctuation handling and so forth, the use of regular expressions was of high importance. PERL has one of the most sophisticated regular expression subsystems of any language¹.
- Simple file handling: As multiple files needed to be opened and parsed, the straightforward file handling subsystem in PERL was highly attractive.
- Primitive hash table data structure: PERL has builtin mechanisms that treat hash tables as primitive types, and give full support to this type of data structure. Because of this fact, and our intention to count words using such a structure, we are especially encouraged to use this language.

2.1 Program Description

The program is very simple, most of its 'code mass' being attributed to functionality for reading and traversing files and directories. The important functions and their descriptions are found in table 1. There are a few global variables:

<code>FreqByWord</code>	A hash table where a string (a word type from the text) serves as the key and an integer (the frequency of that word type) as the value.
<code>FoFbyF</code>	A hash table where an integer (a possible frequency of a wordtype in our text) serves as the key and an integer (the frequency of that frequency) serves as the value.
<code>sourcefiles</code>	An array where each element is a string representing the filename of a source text file on which to perform the requisite word counting.
<code>*SENSITIVE</code>	Boolean flags that determine whether or not punctuation sensitivity and/or case sensitivity should be active, for the purposes of experimentation.

¹In fact the PERL standard of regular expression syntax is matched in its practical and theoretical popularity only by the simpler base UNIX regular expression system.

function	description
getSourceFiles()	shifts through command-line arguments sequentially, adding files to the file array as necessary.
getSourceDirectory(dirname, index)	given the name of a directory, and the number of files already found (index), adds all contents of a directory to the file array, recursing into further directories as needed.
countWords(sourcefile)	given a sourcefile, this function will update the hash table <code>FreqByWord</code> by word
SortWords()	returns an array of (word, frequency) pairs from data in <code>FreqByWord</code> sorted by increasing frequency (the index of this array represents the respective word's rank.
Output(filename)	writes selections of verbose output in the form of tables to <code>stdout</code> , and less-verbose but more complete representations to a file of the form <code>results/*.dat</code> fulfilling requirements 1-3 of the assignment.
init()	creates a results directory.
main()	serves as the entry point to the program.

Table 1: Important functions in `a1.comp791a.1234567.pl`

2.2 How to Run the Program

The program assumes a UNIX-style filesystem and should be run locally or remotely on a Linux or Solaris computer. It also assumes that the user has a compatible PERL environment that can evaluate the program. Note that this program has been successfully tested on the following machines (described by their respective operating systems and versions of Perl):

- Author's Machine => (RedHat Linux 8.0 , PERL v5.8.0 built for i386-linux-thread-multi)
- `clac.cs.concordia.ca` => (RedHat Linux 7.2, PERL v5.6.0 built for i386-linux)
- `alpha.cs.concordia.ca` => (Solaris , PERL version 5.005_03 built for sun4-solaris)

The program will initially be in a tarball file. Copy it to an appropriate directory and untar it. This will create its own directory with the following contents:

1. `a1.comp791a.1234567.pl` the executable source file. If not already executable, perform a `chmod u+x` on this file
2. `data/` a directory containing a sample cross-section of the data used in the experiments detailed in section 3.1.

3. [a1.comp791a.1234567.pdf](#) This document is Adobe *.pdf format, viewable from a variety of programs (`gv`, `xpdf`, `acroread`, `xdvi`, ...).

Performing the following steps will un-tar and run the program:

```
% tar zxvf 1234567-a1.tar.gz
% cd 1234567-a1
% ./a1.comp791a.1234567.pl data
```

Note that the program takes as arguments *any number of directories and files*. If a given argument is a directory, then it is *recursively searched* for *all* text files contained therein. All text files found on the command line or in a searched directory (not beginning with a '.' or ending with a '~') are added to the list of test documents.

The program will produce resultant data files in the *.dat format in the auto-generated `results` directory.

3 Experiments

It was the intention of the author to expose Zipf's Law to highly skeptical, or at least generally thorough, experimentation and testing with the intention of casting empirical dispersion on Zipf's Principle of Least Effort

3.1 Corpora

All Corpora were retrieved via the internet portal "**The Online Books Page**"² in the 'Gutenberg text' plaintext file format. In total, **19** texts were selected and classified according to the Library of Congress classification system standard for the categorization of texts (detailed briefly in table 2. In table 3, we outline those documents in terms of their categorical classification, their titles, authors, filename, file size in kB and their word *token* count³.

In our corpora there is an average of **297555** word tokens per document.

3.2 Description of Experiments

We wished to provide a comprehensive analysis of the phenomena surrounding Zipf's Law, and thus defined the following series of experiments.

3.2.1 Empirical Confirmation of Zipf's Law

The fundamental purpose of this exercise was the simple confirmation that Zipf's Law is generally a good approximation to the reality of rank vs. frequency in normal textual documents. We run our program over our complete collection of

²<http://digital.library.upenn.edu/books/>

³Word 'tokens' are defined as in "Foundations of Statistical Natural Language Processing" (Manning & Schütze, 2002), page 21, and were derived using the UNIX `wc -w` command.

Category	Description
A	General Works
B-BD	Philosophy
D	History: General, and Regions Outside the Americas
PS	Literature: American
QA	Mathematics and Computer Science
QC	Physics
M	Music
TK	Electrical Engineering (and computer networks), Nuclear Engineering

Table 2: Library of Congress classification system.

19 documents, then derive the least-squares straight-line fit through the graphical representation of the rank vs. frequency data. Measurements of success will be based on the least, greatest and mean experimental deviation from the straight line for all texts.

It is important to state that since we have transformed our domain and ranges (in rank vs. frequency respectively) from quadratic space into log-log space, we must also transform our representation of a straight line from quadratic space into a straight line in log-log space. Recall the straight-line formula $y = m \cdot x + b$ in quadratic space that plots a straight line with y-intercept b and slope m . Our transform to log-log space will be done thusly: First, write the linear equation taking the logs of both the dependent variable y and dependent variable x :

$$\log(y) = m \cdot \log(x) + b$$

Eliminating the log notation on the left-hand-side, we must raise 10 to both sides of the equation, resulting in:

$$y = 10^{m \cdot \log(x)} \cdot 10^b$$

$$y = 10^b \cdot 10^{\log(x)^m}$$

$$y = b_1 \cdot x^m$$

Given the last equation, we now have a nice, simple form that we can use to approximate our data by a straight line in log-log space, as will come in later sections.

3.2.2 Corpus Size and Zipf's Law

Based on the data procured in the experiment described in section 3.2.1, we wish to determine if there is any evidence to verify the claim that corpus size has any influence in the success of Zipf's Law by comparing the degree of accuracy procured in each of the documents in the previous tests against the size of the

#	Cat.	Title	Author	Filename	File Size	wc
1	A	"Literary and Philosophical Essays: French, German and Italian"	N/A	litpe10.txt	908 kB	154750
2	B-BD	"The Yoga Sutras of Patanjali: The Book of the Spiritual Man"	Patanjali	patan10.txt	192 kB	32304
3	B-BD	"Laws"	Plato	plaws10.txt	948 kB	166013
4	B-BD	"The Republic"	Plato	repub11.txt	1,235 kB	218110
5	B-BD	"A Treatise Concerning the Principles of Human Knowledge"	George Berkeley	prhkn10.txt	233 kB	39737
6	D	"Travels Through France and Italy"	Tobias Smollett	ttfai10.txt	857 kB	144169
7	D	"The Ruins: or, Meditation on the Revolutions of Empires; and The Law of Nature"	Constantin-Francois Volney	ruins10.txt	602 kB	99786
8	D	"Medieval Europe"	Henry William Carless Davis	mdvlp10.txt	333 kB	54230
9	D	"The Deeds of God Through the Franks"	Guilbert of Nogent	7deed10.txt	531 kB	89949
10	D	"War and the Future: Italy, France and Britain at War"	H.G. Wells	wrftr10.txt	372 kB	63425
11	PS	"Moby Dick"	Herman Melville	moby10b.txt	1,256 kB	214112
12	PS	"The Fall of the House of Usher"	Edgar Allan Poe	usher10.txt	54 kB	8808
13	PS	"Frank's Campaign: or, The Farm and the Camp"	Horatio Alger	frcmp11.txt	365 kB	63643
14	PS	"The Adventures of Huckleberry Finn"	Mark Twain	hfinn11.txt	596 kB	114570
15	QA	"Free as in Freedom: Richard Stallman's Crusade for Free Software"	Sam Williams	freed10.txt	475 kB	75421
16	QA	"The Economy of Machinery and Manufactures"	Charles Babbage	cnmmm10.txt	641 kB	107745
17	QC	"Relativity: The Special and General Theory"	Albert Einstein	relat10.txt	225 kB	36854
18	M	"Mozart, the Man and the Artist, As Revealed in His Own Words"	W.A. Mozart, F. Kerst, H.E. Krehbiel	wamma11.txt	200 kB	34892
19	TK	"Edison: His Life and Inventions"	Frank L. Dyer, Thomas C. Martin	ehlai10.txt	1,608 kB	271054

Table 3: Experimental corpora.

individual texts (in terms of word tokens).

3.2.3 Zipf vs. Mandelbrot

We will examine the degree to which Mandelbrot's formula can give a more accurate depiction of the rank vs. frequency relationship between words in a text by observing its curve properties in conjunction with results found in section 3.2.1.

3.2.4 Case Sensitivity and Zipf's Law

We will attempt to determine if case sensitivity will play any part in verification of Zipf's Law. This is easily accomplished through normalizing the case of all alphabetic characters and will allow us to better determine the robustness across different scenarios.

3.2.5 Punctuation Noise and Zipf's Law

We will attempt to determine if punctuation will modify our verifications of Zipf's Law. Punctuation noise is determined by whether or not we wish to include punctuation marks as part of our word tokens. Consider the case where the word "sentence" ends a sentence. In the previous sentence (and in this one), are the space-separated tokens {"sentence" sentence. sentence} all equivalent to one another? Normally we assume that they are all equal in terms of word type, but we wish to determine if this class of problem has any effect on the robustness of Zipf's Law to withstand different scenarios. This experiment will easily be accomplished through the removal of punctuation from word tokens.

3.2.6 Categorical Sensitivity and Zipf's Law

We will attempt to determine if there is any relation between the class of a text and its adherence to the general Zipf's Law for word rank versus frequency. It is the author's hypothesis that there will be no substantial distinction that the classification of the text can influence in terms of the results. Put another way, I claim that *in general* the category of the text will have only minimum influence on the accuracy of Zipf's Law on real data - and that the two phenomena are relatively independent. This will be confirmed if the average error rate is a reasonably constant value across all categories.

3.2.7 Zipf's Law Extended into Word-Length Dimension

According to other of his laws and results of Zipf's Law, Zipf predicts an inverse relationship between the frequency of words and their length. The length of words is easily determined and we will graphically determine if such an inverse relationship exists, for a subset of our data, with the intent of discrediting Zipf's work.

3.2.8 Miscellaneous: Effect of Collocation-Handling, Morphology-Handling

We propose further experimentation in the following areas:

- Test with collocation handling. Currently we only consider word tokens as collections of characters that are separated by whitespace, although the point has been made that collections of contiguous words have meaning that is beyond the sum of their parts. That is, a series of word tokens can sometimes serve the same purpose as single space-delimited word tokens and therefore can be considered more than just a disjoint collection of its component words. Examples of collocations would be word groups such as “blue screen of death”, “black box”, and “Microsoft Windows”⁴. Collocations would be entered into a lexicon or dictionary and clustered together to form single words in our array of words.
- Morphology handling. We wish to understand how considerations of morphology would affect our verification of Zipf’s Law. Morphology handling in our case would mean taking the root or stem of a wordform as the word type for which we wish to determine the frequency. For example, “quick” and “quickly” both have the same stem - “quick”.

4 Results & Analysis

Below we provide the graphical results of our experiments. A small cross-section of the tabular data which is used to construct these graphs can be observed in Appendix A in fulfillment of the requirements of submission. Naturally, full disclosure of all data is not present in this document due to the constraint in document size, but can be obtained by request from the author (some_address@cs.concordia.ca).

Graphical data was obtained using the `gnuplot` program (included free with Linux, which is also free). The following executable script was written to automatically generate graphical *.eps files from data (then converted to *.gif format using `convert`).

```
#!/usr/bin/gnuplot
set xlabel "Rank"
set ylabel "Frequency"
set zlabel "Word Length"
set logscale xy
f(x) = b*x**m
x = -2
b = 10000
```

⁴These are all collocations specific to computer terminology. Indeed, different topical domains has different terminology, and any experimentation in this area would likely have to take the class of document into account when performing collocation-handling.

```

fit f(x) 'filename.txt.dat' using 2:1 via m, b
set output "images/filename.txt.eps"
set terminal postscript eps
plot f(x), 'filename.txt.dat'

```

4.1 Empirical Confirmation of Zipf's Law

Our confirmation of Zipf's Law will come by plotting a trendline *in loglog space* which will approximate *all* of our collected data (through every single point in the rank domain). Experimental confirmation of Zipf's Law will only come satisfactorily if this plotted trendline is a 'reasonable' approximation to the trendline.

Our numerical error measurement will be the ASYMPTOTIC STANDARD ERROR⁵ of the plotted line falls within a relatively very small (has a very small percentage of error). Asymptotic Standard Error, basically, estimates the offset error between the projected plot (our straight line). and our actual data, so in this case it will give us an error estimate on two variables. Note that all of these values were obtained directly via the `gnuplot` program (which also produced our graphs), and can be observed in Table 4.

In table 4 we see the asymptotic standard error for variables m and b for each of our 19 texts, with case and punctuation sensitivity both set to `off`. We note that the lowest error is **0.1725%** and **0.06018%** for each respective variable (interestingly, both occurring for the "Edison: His Life and Inventions" text, which we currently believe to be the text most convincing of Zipf's Law). The worst error in each case is **0.5417%** and **0.1249%** respectively and the average error is **0.244595%** and **~0.0819%** respectively. We provide two log-log graphs from two sample texts in figure 1 that show a very short text (a) versus a medium-length text (b). Note that both graphs give reasonable approximations to the straight line in terms of accuracy, although the y-error in the graphs may seem to be large for the higher ranks, recall that these errors will typically vary no more than $\sim 10/10000$, or 0.1%.

In general, since even our 'worst-case performance' gives asymptotic standard error estimates of relatively small values for each unconstrained variable (a few tenths of a percent), we are satisfied that in the most unconstrained case, Zipf's Law is acceptably accurate.

⁵Defined as "The real (enlarged) error is only slightly larger than the asymptotic error when the data approximate the model. Use enlarged error when misfit is due to systematic error. The modelled asymptotic standard error (precision) is that which applies, in the limit, when data fit the model. It is the smallest possible standard error (highest possible precision), and implies that the data accord with the measurement model, i.e, that misfit is random. If you hypothesize that noise in the data is systematic and not random (the worst case), and so wish to see standard errors (precisions) enlarged to encompass this systematic lack of fit discovered in the current analysis, type:

SE = Real ; "Real" can be abbreviated to "R"

SE=Real is the more conservative approach. It is recommended for exploratory analysis.

Real S.E. = Model S.E. * sqrt(Max(INFIT Mnsq, 1/INFIT Mnsq))"

(http://www.winsteps.com/facetman/5_7_2.htm)

doc	1	2	3	4	5
ASErr	m:0.5417%, b: 0.1628%	m: 0.1972%, b: 0.06753%	m: 0.1914%, b: 0.06576%	m: 0.1839%, b: 0.06367%	m:0.183%, b: 0.06339%
doc	6	7	8	9	10
ASErr	m: 0.2292%, b: 0.07699%	m: 0.2175%, b: 0.07368%	m: 0.2097%, b: 0.07152%	m: 0.205%, b: 0.06994%	m: 0.2012%, b: 0.06874%
doc	11	12	13	14	15
ASErr	m:0.3806%, b:0.1249%	m: 0.3078%, b: 0.1018%	m:0.2857%, b: 0.09598%	m: 0.2686%, b: 0.09073%	m: 0.2531%, b:0.08542%
doc	16	17	18	19	
ASErr	m: 0.2376%, b: 0.08107%	m:0.1824%, b: 0.06316	m:0.1992%, b: 0.06809%	m:0.1725%, b: 0.06018%	

Table 4: Asymptotic Standard Error for straight line approximations to our data

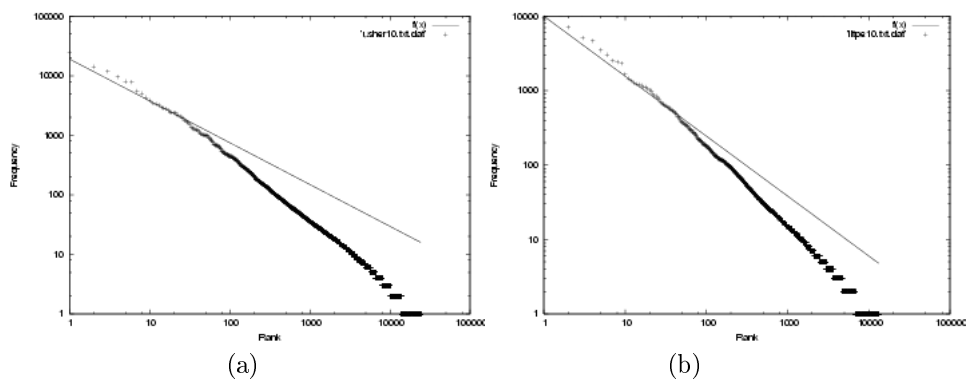


Figure 1: log-log values graphed by rank vs. frequency for texts (a) “The Fall of the House of Usher” by Edgar Allan Poe, and (b) “Literary and Philosophical Essays: French, German and Italian”

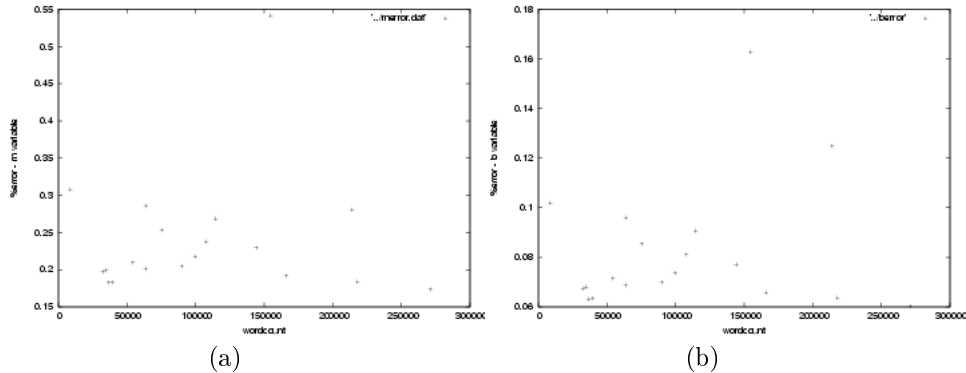


Figure 2: Corpus Size versus (a) ASError in m, and (b) ASError in b

4.2 Corpus Size and Zipf’s Law

We observe that although the data obtained would suggest a possible *graphical* relationship between the size of the corpus and the shape of the rank-frequency curve as it approximates a straight line. For example, In figure 1 one can observe a slight ‘bulge’ in the lower ranks of graph (a) which offsets the general tendency of the rest of the data, whereas graph (b) tends more towards a straight line throughout all data. In this case, “The Fall of the House of Usher” contains about $8808/154750 \cong 5.7\%$ the amount of words as “Literacy and Philosophical Essays”. This early ‘bulging’ of the graph would suggest a difference, but numerically across multiple texts we see that any difference is for all normal purposes negligible.

Our preliminary assumption was that smaller texts would deviate most drastically from Zipf’s Law. To verify this, we plot the average asymptotic standard error as the variable dependent on the length of a text and search for any general trends. Two graphs will be drawn - one for each of the unconstrained variables to our straight-line formula in log-log space, each with case and punctuation sensitivity both set to `off`. As can be seen in the two graphs in Figure 2, our data does not offer to us any trends that would add any specialization to Zipf’s Law. That is, it appears as though corpus size would not play a significant role in the defamiation of Zipf’s Law, as the data’s approximation to a straight line seems to be approximately random, although the very shortest text *does* give slightly worse-than-average performance than does the mean text length, this is not sufficient and we determine that even across widely different sizes of corpora, Zipf’s Law is basically confirmed, although further tests would be required.

4.3 Zipf vs. Mandelbrot

This experiment has been delayed for reasons described in Section 5

4.4 Case Sensitivity, Punctuation Noise and Zipf’s Law

With an average of ~ 12378.8 sentences per text⁶, and an average of **36005.8** word types per document⁷, we can estimate that in the absence of other information a randomly chosen word type will lead (or finish) only **0.3438** sentences in the document. Put another way, a randomly chosen word type will lead or finish only $2.7E^{-5}\%$ of all sentences. Of course, some words, determinants such as “the” for example, are much more likely to lead a sentence than others, and some other words are more likely to terminate a sentence (and therefore be punctuated), but we assume that these differences between word types are highly negligible given the extreme value of our calculated probability. What this suggests is that a random word type is very infrequently coupled with a punctuation mark, or capitalized abnormally.

As can be verified by the user of our program, altering the case or punctuation sensitivity had virtually no impact on the general shape of the curve. Tests were run for all four combinations of these two variables and in each case no discernable or characteristic change could be observed in the shape of the curve, and the asymptotic error differed only minimally.

Although one of our more informal tests (because of the predictability of the result), we claim that punctuation and case sensitivity in any combination have but effect on the shape of the rank vs. frequency information.

4.5 Categorical Sensitivity and Zipf’s Law

It was the assumption that the category of a document will have minimal, if any influence at all on the accordance of rank/frequency data to Zipf’s Law, but this assumption was shown to be possibly erroneous. As data in Table 5 will show, the category of a document does not play a substantial role in this respect, however we actually *do* see that from our most performant categories, so to speak, which are {B-BD, QC, M, TK}, all but B-BD have to do with the reporting of technical or specialized data or information (their focus is more concentrated). Philosophical writings also, intuitively, tend to be concentrated on a specific avenue of human behaviour or of civilization. The observation we make is that it appears as though the more concentrated a text is meant to be topically, by a possible consequence of its domain (category), then the more likely it is to conform with Zipf’s Law. There is little theoretical evidence to back up this claim, but it is persuaded empirically by the fact that the more unconstrained categories, namely “General Works” and “Literature”, deviate more drastically from the straight-line approximation than their ‘peers’, although far more data would be required to rigorously prove this claim (and perhaps some more advanced techniques of .

A similar experiment would involve the individual author’s effect on the properties of rank vs. frequency. We have two documents by the same author-

⁶A rough estimate obtained through a series of UNIX commands of the form `%grep “. ” A/litpe10.txt | wc -l`

⁷Obtained from running `%ls | xargs tail` on a directory of our *.dat files, and averaging.

Category	Average error
A	m:0.5417%, b:0.1628%
B-BD	m:0.188875%, b:0.0650875%
D	m:0.21252%, b:0.072174%
PS	m:0.310675%, b:0.103353%
QA	m:0.24535%, b:0.083245%
QC	m: 0.1824%, b: 0.06316%
M	m: 0.1992%, b:0.06809%
TK	m: 0.1725%, b:0.06018%

Table 5: Error as a function of document category.

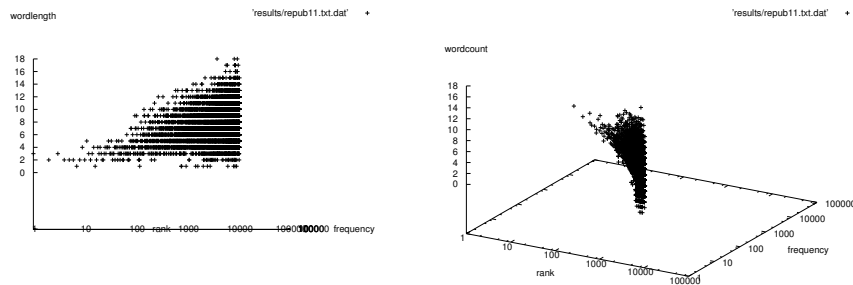


Figure 3: Rank vs. Frequency vs. Word Length for “The Republic”

Plato, but a full experiment of this type would require at least 6-7 authors, each credited with 2-3 texts (estimate).

4.6 Zipf’s Law Extended into Word-Length Dimension

Zipf claims that there exists an inverse relationship between between the frequency of words and their length, and our program very easily generates wordlength information for each word and stores this in our output files. It is readily simple for us to then generate straight-line approximations to the frequency, which shows a band in the frequency/wordlength dimension as shown in Figure 3 (a). Figure 3 (b) shows the 3-dimensional shape of rank vs. frequency vs. wordlength (rank and frequency are represented in logscale). These graphs verify (at least for a single text), that the more frequent words tend to be short in size, although the more infrequent words tend to take on a wide range of possible word lengths.

4.7 Miscellaneous Experimentation

The experiments involving collocation and morphology will, unfortunately, not be a part of this report. Namely, in each case we paint our reasons for omitting

each case from our list of experimentations:

- Collocation handling, while easy to program, would be more time-consuming to 'perfect'. Namely, it would require an extensive list of collocations compiled by experts, only very limited ones are readily available free online. It is the assumption of the author that this type of experimentation would not affect the validation of Zipf's Law significantly, because of the accepted and documented rarity of collocations relative to 'standard' word forms.
- Experiments on morphological complexity. Performing this task as previously described would easily be performed using WordNet's builtin capabilities for performing this function (which has interfaces for PERL), and would effectively reduce the number of word types, and we would like to see any further changes that might occur. Although this would be an easy task that is just within the realm of this subject - this too was not performed.

The simplest reason for not performing these experiments is that the author did not have enough time, given other undergraduate responsibilities. According to various miscellaneous web literature, it has become evident that these two types of experiments would not offer any immediate new insights, or disprove Zipf's Law in any substantial way. It is my expectation that these experiments will be delayed as 'future work' because of their interesting nature.

5 Conclusion and Future Work

In general we have repeated confirmations of Zipf's Law in the details provided above, although further work is required to validate thoroughly. We are especially intent on comparing Mandelbrot to Zipf in terms of accuracy, and further experiments will involve automated determination of differences between the two approaches.

APPENDIX A: Supplementary Data

Below are the first 100 ranks (100 most frequent words) from a single sample text: "The Republic" by Plato. Also included is the frequency of frequencies for this text. Note that the source of this text is included in the electronic submission and can be verified via the command `% ./a1.comp791a.1234567.pl data/B-BD/repub11.txt`

APPENDIX B: Source Code