# Using Similarities in Zipf plots of Natural and Artificial Texts to Distinguish Between Communication and Random Data

Niels Koek

March 15, 2005

# Acknowledgements

I would like to express my gratitude to all those who supported and helped me with he completion of this thesis. In particular I would like to thank my supervisor, Dr. J.J. Paijmans whose help, suggestions and support proved invaluable during the writing of this thesis. I would also like to thank Drs. M.W.C. Reynaert for reading my thesis and completing the examining committee.

# Contents

# List of Figures

# Chapter 1

# Introduction

Communication can be defined as the process of exchanging information using a common system of symbols [Sinha:2001]. These symbols can be anything from shapes and colors to words and sentences in a language. Many things communicate and many forms of communication exist. There is of course human language but many animals also communicate with each other in some form. This is done trough sounds, facial expressions or gestures and some believe even the movements of a bee are a form of communication [Frisch:1967]. The complexity of these forms of communication varies tremendously. While certain monkeys only have a vocabulary of 26 different calls [Rossie:2002], human language is much more complex with hundreds of thousands of words available and an unlimited number of combinations to use these words. Measuring the complexity of a form of communication can be done by applying certain quantitative measures from information theory [McCowan:1999].

Communication, however does not only apply to things that have evolved naturally. Mankind has created various forms of artificial of communication. Not only fantasy languages but also more practical forms of communication, like the protocols that machines use to communicate, indexing systems for libraries or the programming languages that were created so that human programmers have a way of communicating with computers and instruct them in what functionality to display. The development of these programming languages started in the middle of the 20th century and has progressed rapidly. Initially the programmer was required to type the ones and zeros of the machine code directly, followed by using more and more advanced languages like Assembly, Fortran or C++ that are translated by a compiler. The development of the programming languages was necessary to add func-

tionality, but also to make it easier for the human programmer to complete certain tasks. The development of these programming languages has also caused the languages to resemble natural languages more closely. Instead of only displaying the ones and zeros many parts of a computer program consist of commands that are understandable even for an untrained person. It would therefore be interesting to see if certain phenomena occurring in natural language can also be found in these programming languages. Perhaps programming languages can even help us to understand why these things can be observed in a natural language.

In this thesis certain methods that are used to analyze natural language and other forms of communication will be examined. The focus will lay on Zipf's rank-frequency law, which will be discussed in more detail in the following chapters. First a short summary will be given of methods that can be used to analyze both natural language and other forms of communication. This summary will be followed by an overview of how those methods have been applied to the analysis of different forms of communication. Zipf's law is not just a random phenomena in natural language. Scientists have argued that the emergence of Zipf law in a natural language text is actually a result of how these languages evolved [Ferrer:2003]. They base their finding on the principle of least effort: When a speaker and listener are communicating a speaker has to put more effort into forming a sentence when different words are available for use. For example if there are five different words for "chair", each describing a slightly different type of chair the speaker has to decide which word is best suitable in the sentence he is forming. On the other hand the listener will know the exact details about the chair if a word is used that describes a specific type of chair, it will cost him less effort. In their research Ferrer and Sol argue that Zipf's law appears in systems using symbolic reference.

Zipf plots for a natural language often show a more or less pronounced bulge; the second half of the line shows a slightly steeper slope then the first part of the Zipf plot. One theory to explain this phenomenon is given by Ferrer and Sole[Ferrer:2001a][Ferrer:2001b]. Here the bulge in the plot is explained by the presence of two distinct lexica: A first group of everyday words that are commonly used in most texts and a group of less frequent words. This lexicon is called the kernel lexicon by Ferrer, in this thesis it will be referred to as the *A lexicon* while the second group will be the *B lexicon*. The explanation offered by Ferrer for this phenomenon by theorizing that the lexica division and the resulting bulge in a Zipf plot is the direct result of the limits of the human brain. As plots of randomly generated texts, or *monkey languages*, do not show this bulge the question arises if the presence

of such a bulge, in a Zipf plot is an indication for real communication, as opposed to random data.

- In this thesis the theory of two separate lexica within communicative data will be further examined. Drawing on the similarities that exist between natural language and artificial languages, the presence of two different lexica in these two groups, and absence in random texts, will be shown. Further parallels can then be drawn to other fields of research like DNA.

A first step in determining whether or not the presence of a 'bulge' in a Zipf plot can be attributed to the presence of two lexica, Lexicon A and B, is to prove the presence of such groups. In the experiments described later in this thesis, the natural language text will be split into two groups: one of frequent, lexicon A words with function words included. A second file will contain lexicon B, with all remaining words. To further examine this phenomenon of a bulge being caused because a certain piece of text contains "meaning", various corpora of computer languages will be examined for the presence of two word groups. The major advantage that computer languages have over natural languages is that a division between daily used 'words' and others can be more easily made. Keywords, operators and standard functions are considered 'daily' words while all other user created identifiers are considered part of the less frequently used words. These keywords and standard functions are clearly defined for all program languages described in this thesis. After separating the two different groups of words for each corpus into an A and B lexicon, Zipf plots will be created. Because we assume that the bulge in a Zipf plot is created by the presence of the two word groups, we expect that at least in one of the plots of the separate groups such a bulge will not, or at least be less, visible. Because random texts do not consist out of different types of words a Zipf plot will be created for one of such texts to show that a bulge like in the non-random data is not present. If we indeed find that the bulge is absent for random, but present for non-random texts we can then conclude that the bulge is only present in communicative texts. Further parallels can then be drawn with other types of data to see if similar separations of word-group equivalents produce similar results.

In chapter two methods that are used to examine linguistic like data are discussed, including entropy and Zipf's laws. Following this chapter, chapter three will give a summary on various research area's where these methods have been applied. The focus will then shift to the experiments that were conducted for this thesis. In chapter four an overview will be given of the

corpora that were collected to perform the experiments. The Experiments themselves will be described in chapter five. Finally the conclusion will contain a short discussion on the results that this thesis yielded.

# Chapter 2

# Finding similarities in communication

In this chapter methods deciding on the complexity of a communications stream will be described. The focus will lie on Zipf's laws and entropic measures, as these have both been used in other experiments were human language was compared to other forms of communication[McCowan:2002].

## 2.1 Zipf's laws

Zipf's laws are two laws of word frequency which apply to natural languages like English, German, Chinese and Japanese [Zipf:1949]. The most famous of these two shows a relationship between the frequency of a word and the rank of that word in a sorted word frequency list. A plot of the log of the frequency on the Y-axis and the log of the rank on the X-axis results in an almost straight line with a slope of -1. Variations in the slope of the line most often occur for the highest and lowest ranking words.

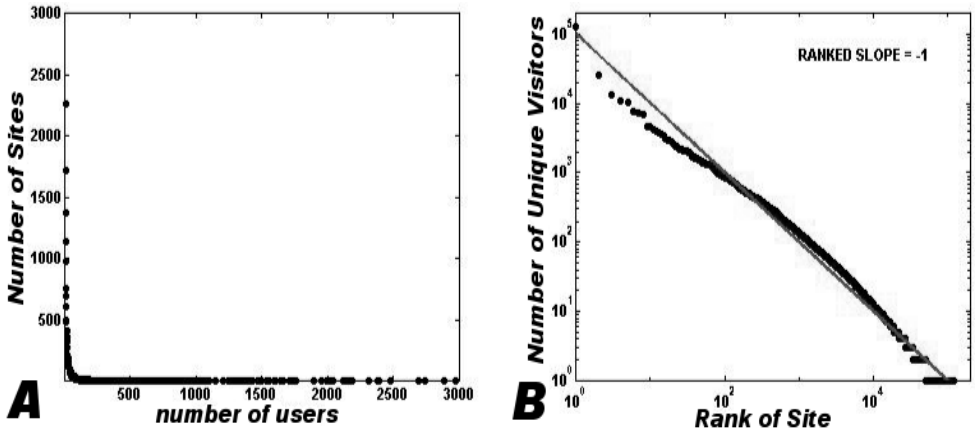


Figure 2.1: A) Linear scale plot of the distribution of AOL users among visited web sites. B) Distribution of site popularity frequencies on a log-log scale for AOL users. Please note that the "bulge" is present, even though this plot does not result from linguistic observation.

Apart from natural languages this law also applies to random texts, artificial languages [Cohen:1996] and animal communication [McCowan:1999]. It also holds for very different things like city populations, internet traffic and financial systems [Li:1999]. Figure 2.1A shows the visits of AOL users to different websites during one day in December 1997. Figure 2.1A illustrates what happens when the web visit data is put on a log-log scale; the Zipf distribution appears. The straight line shows what a perfect Zipf distribution should look like, the other line shows the actual distribution [Adamic:2002]. This rank-frequency law is often referred to as Zipf's law, however there exists a second law for word frequencies by Zipf. The second law is also known as the number-frequency law and describes a relationship between words with a certain frequency $n$ and other words with the same frequency. If a plot is drawn with the log of $n$ on the Y-axis and the log of the number of other words that also occur n-times on the X-axis the plot approximates a straight line with slope -0.5[Popescu:2003]. Figure 2.2 shows what happens when Zipf's second law is applied to the United States constitution.

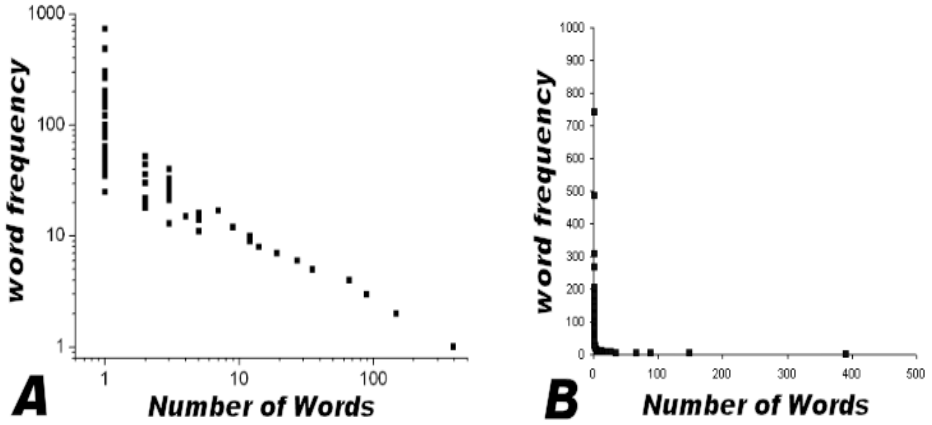


Figure 2.2: A) An example of a plot showing Zipf's number-frequency law in the US constitution. There are many words that occur only once, while there are relatively few that occur more often. B) The same graph, but on a regular scale.

## 2.2 Other Word Frequency Distributions

There are also other methods that can be used to describe the frequency distribution of words. In particular several distributions that are used in various fields of science can be adapted to estimate a total vocabulary size from a sample. Examples of these are the lognormal distribution [Carol:1967] and the generalized inverse Gauss-Poisson law [Sichel:1986]. These methods are included in this thesis to show that alternatives to Zipf also exist.

### 2.2.1 The lognormal law

The Lognormal Law corresponds with the notion of proportional effect [Laherrere:1997]. An example of this is the amount of money people in a country can make. It is much easier for a millionaire to make a thousand Euros than it is for someone with a smaller capital. Similarly a word that occurs often in a sample has a high probability of occurring often in the complete vocabulary. The Lognormal Law can be used to calculate the probability that a certain word will make up a certain percentage of a complete corpus that the sample was taken from.

### 2.2.2 Gauss-Poisson

The Gauss-Poisson distribution, also known as the poisson normal distribution, can be applied in the same way as the lognormal law to estimate vocabulary size. The Poisson distribution is a discrete probability distribution belonging to certain random variables N that count, among other things, a number of discrete occurrences that take place during a time-interval of given length[WikiPoisson:2005]. Given a sample the distribution can be used to estimate how many times a certain word will occur in the total corpus. Besides being used in linguistics the distribution applies to a wide variety of events ranging from the number of times a web server is accessed per minute, to the number of soldiers that died after being kicked by horses each year in each corps in the Prussian cavalry[1].

### 2.2.3 Comparison with Zipf

Baayen[Baayen:1993] compared the Lognormal law and the Poisson distribution to a generalization of the Zipf frequency/rank law. He found that the Zipf law and the Gauss-Poisson law performed better than the lognormal distribution for estimating vocabulary size, the difference in performance between Zipf's law and the Gauss-Poisson law was very small.

## 2.3 Entropy

Another way of analyzing communicative data is by measuring the entropy, or unpredictability, within a stream of symbols[2]. The amount of information in such a stream can be estimated by an entropy measurement [Shannon:1948]. There are several entropic orders which all measure different things. Zero-order entropy measures repertoire diversity and can be calculated using formula 2.1

$$H_0 = \log_2 N \tag{2.1}$$

[1]This example became famous because of the book "The Law of Small Numbers" written by Ladislaus Bortkiewicz. He used the example to illustrate that certain low frequency events, in a large population follow a Poisson distribution.

[2]During the creation of this thesis it became apparent that experiments involving Entropy would not yield results that could help in finding the answers to the questions that were raised in the introduction. However because many researchers who use Zipfian measures to analyze data with linguistical properties also use entropy, a short introduction will be given nevertheless.

In this formula $H_0$ is the number of bits that are required to represent a number of events N. In languages these events can be letters, morphemes or words, depending on what are considered to be the basic elements of the repertoire. The next order of entropy, first order entropy, is a measure for the information content of a set of events and can be calculated using formula 2.2

$$H_1 = \sum_{j=1}^{N} -p(A_j)\log_2 p(A_i) \tag{2.2}$$

Unlike zero-order entropy, first-order entropy takes into account the probability of each event occurring. The higher levels of entropy measure the dependency among communicative signals in sequences of multiple signals. For example 2nd level entropy measures the likelihood of signal B occurring when signal A has just been identified. Similarly 3th order entropy measures the likelihood of a signal C occurring, when A and B have already appeared. The formula for calculating 2nd order entropy is:

$$\begin{aligned}
&\mathrm{H}_2(\mathrm{AB}) = -\mathrm{p}(\mathrm{A}_1\mathrm{B}_1)\log_2(\mathrm{A}_1\mathrm{B}_1) - \\
&\mathrm{p}(\mathrm{A}_1\mathrm{B}_2)\log_2(\mathrm{A}_1\mathrm{B}_2) ... - \mathrm{p}(\mathrm{A}_1\mathrm{B}_\mathrm{N})\log_2(\mathrm{A}_1\mathrm{B}_\mathrm{N}) \\
&-\mathrm{p}(\mathrm{A}_2\mathrm{B}_1)\log_2(\mathrm{A}_2\mathrm{B}_1) - \\
&\mathrm{p}(\mathrm{A}_2\mathrm{B}_2)\log_2(\mathrm{A}_2\mathrm{B}_2) ... - \mathrm{p}(\mathrm{A}_2\mathrm{B}_\mathrm{N})\log_2(\mathrm{A}_2\mathrm{B}_\mathrm{N}) \\
&... \\
&-\mathrm{p}(\mathrm{A}_\mathrm{N}\mathrm{B}_1)\log_2(\mathrm{A}_\mathrm{N}\mathrm{B}_1) - \\
&\mathrm{p}(\mathrm{A}_\mathrm{N}\mathrm{B}_2)\log_2(\mathrm{A}_\mathrm{N}\mathrm{B}_2) - ... - \mathrm{p}(\mathrm{A}_\mathrm{N}\mathrm{B}_\mathrm{N})\log_2(\mathrm{A}_\mathrm{N}\mathrm{B}_\mathrm{N})
\end{aligned} \tag{2.3}$$

Higher level entropies can be calculated by taking the entropy from the previous level and adding the entropy of the next event occurring. For example 3th level entropy can be calculated using formula 2.4.

$$\mathrm{H}_3(\mathrm{ABC}) = \mathrm{H}_2(\mathrm{AB}) + \mathrm{H}_{\mathrm{AB}}(\mathrm{C}) \tag{2.4}$$

in which

$$H_{AB}(C)$$

is the entropy of event C, when both A and B have occurred.

Several studies have used entropy to compare animal communication like dolphins [McCowan:1999], monkeys [McCowan:2002] and whales [Seife:1999] to human language. A disadvantage of using entropy is the fact that to obtain reliable results the corpora on which the measurement are done need to be rather large, this can be a problem and will be discussed further in the next section.

# Chapter 3

# Previous research

In this chapter an overview of other attempts to use Zipfian and entropic measures on corpora with the intent to find linguistic properties will be discussed. This will include experiments on human, animal and artificial language corpora but we will also touch on other areas like the search for extraterrestrial life or DNA.

## 3.1 Human languages

### 3.1.1 Zipf

As mentioned earlier, the most famous example of Zipf's rank-frequency law is the first example that Zipf himself mentioned in his book[Zipf:1949]. Zipf shows that when you multiplied the rank of a word in a word frequency list of the English language by its frequency you roughly get the same number for all words. Zipf explained his law in terms of principle of least effort. The term 'law' is actually not an ideal description for what Zipf discovered as it is inaccurate for high and low ranking words and only reasonable correct for the middle ranges, therefor it can only be seen as an approximation. In the 1950s Benot Mandelbrot made modifications to Zipf's first law which made it slightly more accurate[Miller:1954]. Ever since Zipf published his rank frequency law in 1949 it has been used to analyze many natural languages including English, Russian and Chinese and in all cases Zipf-like behavior has been observed [Li:1999].

## 3.2 Animal communication

Many animals communicate in some form. For most animals, communication is limited to communication driven by instinct; communicative skills that the animal was born with. Animals born in captivity, with no contact with other members of their species, generally show the same communicative behavior as their wild counterparts that were raised by their parents. This includes the growling of a dog when it perceives a threat, the croaking of a frog at dusk and the signals of courtship certain birds express with their feathers to impress a potential mate [Nollman:1999]. However, some animal species *learn* how to communicate, instead of using their instincts. A example of animals learning to communicate can be found amongst whales and dolphins; the language of these animals evolves during their lifetime and different groups of whales and dolphins from the same species often use different dialects to communicate. For this paper the way in which a certain form of communicative skill was acquired by an animal is not important, the emphasis lies on what kind of similarities or differences there are between different forms of communication and specifically on how these can be measured. In the following sections different forms of communication, and the methods used by other researchers to compare these will be discussed.

### 3.2.1 Dolphins

Extensive quantitative research on corpora of dolphin communication was previously conducted by McCowan, Hanser and Doyle. They used both Zipfian and entropic measures [McCowan:1999] too compare the communication of bottlenose dolphins (Tursiops truncatus) to that of human language [McCowan:2002]. In their research particular interest is paid to the development of dolphin communication compared to that of humans. Using the average slope of the Zipf curve they analyzed the Zipf patterns for different age groups among the dolphin population, and compared the results to the Zipf patterns found in the language of human children, adolescents and adults. The results showed that although different for dolphins and humans, both species showed differences across the different age groups. Several institutions have collected corpora of so called Dolphin whistles. Dolphins communicate by sending out sound waves, which sound like whistles to humans. Figure 3.1 shows two examples of dolphin whistles by displaying the frequency of the whistle in time.

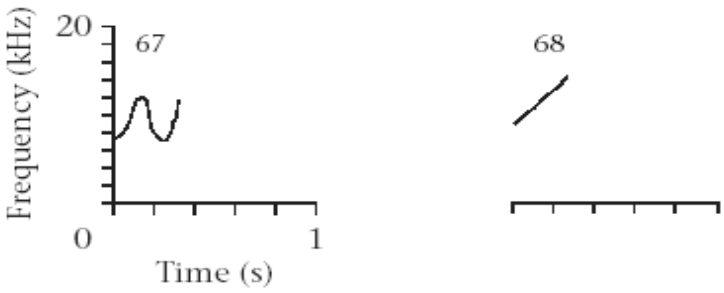


Figure 3.1: Two dolphin whistles from the Janik corpus

**Words versus Characters**

In the research conducted by McCowen and her colleagues both entropic and Zipf measures were used to compare human language to animal communication. A remarkable choice was made to compare words to dolphin communication in the Zipf experiments, while comparing characters to dolphin communication in the experiments which involved entropy. One would expect that a choice would be made to use either characters or words to compare to other forms of communication and not to switch between the two.

### 3.2.2 Whales

Like dolphins, whales communicate using so called whistles. The type of whistle depends on the activity of the whale. Unsurprisingly whales that are resting do not emit as many whistles as a group of whales on the hunt [Ford:2002]. Different groups of whales vocalize different types and quantities of whistles; for example groups of killer whales that have been together for long periods of time tend to communicate more often and emit types of whistles that more recently formed groups do not use. A possible explanation for this is that as a group of whales stays longer together, the social ties within the group become more complex and the need for communication increases. Research on whale communication is in its infancy, but at institutions like the University of Massachusetts scientists are trying to measure how much information whales actually communicate to each other[Seife:1999]. The whale corpora suffer from the same problems as the dolphin corpora; insufficient size and the fact that it is hard for a human observer to classify a whale whistle. A possible solution for this is currently

being worked on: the adaptation of using neural networks that analyze the patterns in whale communication and classify these into categories, without having a prior knowledge of how many categories there are [Murray:1998].

### 3.2.3 Squirrel Monkeys

The squirrel monkey (Saimiri sciureus) is a small primate that can be found in South-America. The squirrel monkey has 26 distinctly different calls which are used to stay in touch when foraging, barking when angry or to warn other members of the group when a threat is spotted [Rossie:2002]. One of the things that makes the communication of these monkeys interesting is that it is relatively easy to classify most calls. The monkeys have several warning calls, but the reaction of the group when a certain call is heard often gives away it's meaning. For example when a monkey gives the 'eagle alarm call' the group of monkeys will look at the sky to spot the threat before finding cover in the bushes. Previous researchers have compared the complexity of the squirrel monkey calls to that of bottlenose dolphins and human language using a Zipf coefficient and entropy measures [McCowan:2002].

### 3.2.4 Bees

Biologists have since long noted that honeybees perform a sort of dance upon their return to their hive. The behavior was first described by Aristotle [Aristotle:330BC] and is known as the bee- or waggle-dance. This dance forms the basis for one of the two main theories about how honeybees manage to communicate with each other [Frisch:1967]. According to the dance theory the bees returning with nectar use the circular and zigzag motions of the dance to communicate the location of the source of the nectar to the other bees in his comb. (See figure 3.2). According to this theory the dance contains information about the direction (relative to the sun) and distance of the food source. The alternative theory about how bees located their sources of food claims that the bee-dance would not be able to pinpoint the exact location of the nectar; however, according to supporters of the dance theory the dance is only used to give the other bees a very general idea of the nectar source [Sebeok:1990][Wenner:1967][Wenner:2002]. Attempts to use linguistic methods to do further research on the bee-dance have so far been hampered by inadequate corpora of dance date [Paijmans:2004].

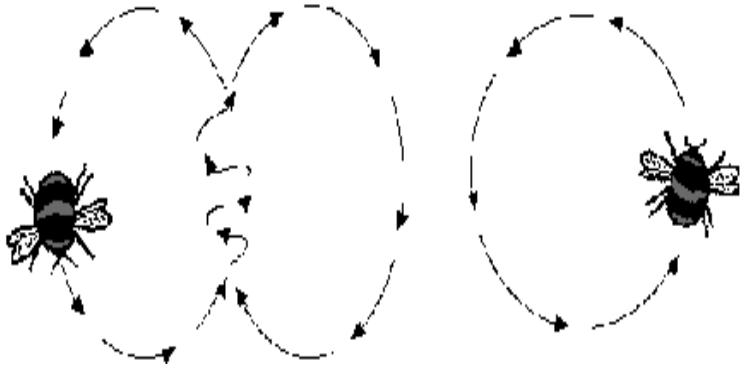

Figure 3.2: Two bees dancing, on the left side indicating a food source far away, on the right, a source nearby.

## 3.3 DNA

Deoxyribonucleic acid, or DNA, is a nucleic acid which carries genetic instructions for the biological development of all cellular forms of life and many viruses [WikiDNA:2004]. In other words; DNA can be seen as a blueprint for a lifework, a sequence of instructions to create a living organism. DNA is encoded with four so called basepairs[1]; the sequence in which these occur on a certain strand of DNA determine the function of that piece of DNA. DNA can be roughly divided into two groups, coding DNA and non-coding DNA. The first group of coding DNA consists out of strands of DNA which contain instructions for making proteins and other cell products. The second group of non-coding DNA makes up the largest part of the DNA of complex organisms like humans. Some parts of the non- coding DNA are involved in the regulation of the coding part of the DNA. However, not much is known about the non-coding part of DNA and it is often referred to as junk DNA as it has no apparent function. Because DNA shows a strong resemblance to a descriptive text with the four basepairs as its alphabet, researchers have employed linguistic methods to analyze the properties of DNA. Some researchers have even argued that a grammar can be formed for

[1]The bases can be abbreviated as A, T, C, and G; each base "pairs up" with only one other base: A+T, T+A, C+G and G+C; that is, an "A" on one strand of double-stranded DNA will "mate" properly only with a "T" on the other, complementary strand. The order does matter: A+T is not the same as T+A, just as C+G is not the same as G+C. However, since there are just four possible combinations, naming only one base on the conventionally chosen side of the strand is enough to describe the sequence

DNA that make use of the X-bar principle and C-command[2] relation used in linguistic syntactic theory [Vides:1992]. Researchers have also used Zipf's rank-frequency law on DNA, for example such an experiment was conducted in 1994 by Mantegna et al. [Mantegna:1994]. The results of their experiment

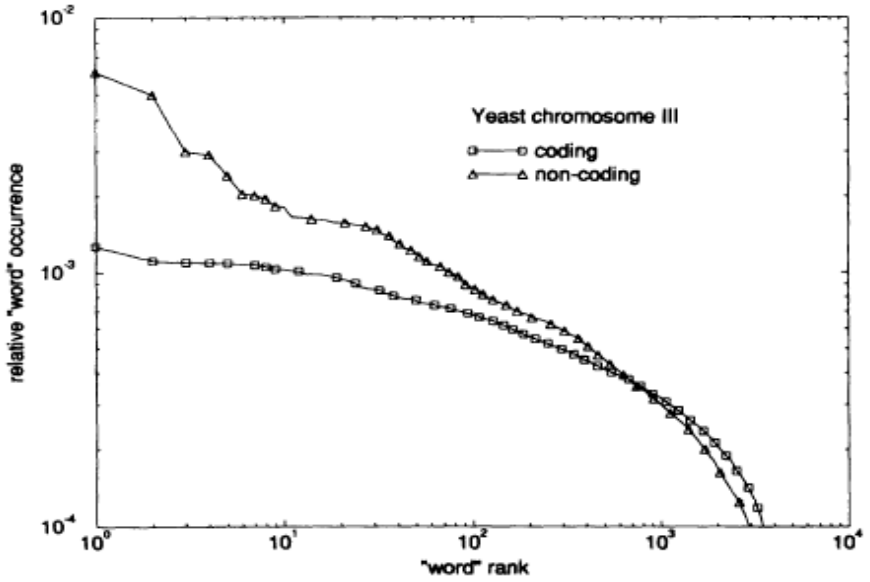


Figure 3.3: Separate Zipf plots for non-coding and putative coding regions of a DNA sequence of yeast chromosome III.

shows that although the coding parts of the DNA does not follow Zipf's law the non-coding part does (figure 3.3). From these results they concluded that non-coding DNA more strongly resemblance natural language than coding DNA does. However, as argued by Niyogi and Berwick [Niyogi:1995] just because a certain phenomenon follows Zipf's law does not necessarily mean it resembles a natural language more closely than something that does not follow Zipf's law. After all, Zipf's law also holds for many things apart from natural languages and therefore one should be careful in assuming that it can be used to measure the degree in which a phenomenon is similar to a natural language, when it is not entirely clear if what is being researched is a language at all.

[2]Both are concepts in government and binding theory which applies to natural language. See `http://www.criticism.com/linguistics/govt-binding-basics1.php` for more information.

## 3.4 SETI

The SETI (Search for Extra-Terrestrial Intelligence) project is a program which searches for intelligent life on other planets, not by visiting those planets, but by analyzing signals from outer space [SETI]. These signals are collected using large radio telescopes and subsequently analyzed to determine wether the signal is just noise or an actual message from an extra-terrestrial intelligence. Because the galaxy is so large (there are more stars than there are grains of sand on all of Earth's beaches) and the SETI project is limited by the number of radio telescopes and amount of computer processing time available, the researchers at SETI have made certain assumptions. The first group of assumptions narrow down the number of stars that the telescopes are pointed at to collect signals. SETI scientists assume that any alien life form will be somewhat similar to the life forms found on earth, they will probably be based on some form of carbon chemistry and will require the presence of liquid water. Furthermore planets around suns similar to our own sun have a higher chance of hosting intelligent life. Large suns have a relatively short lifespan so the chance of intelligent life developing around does suns is smaller. Small suns emit a relatively small amount of heat and energy, which means that planets in those solar systems will be colder and less likely to have liquid water. The 2nd assumption that is made at the SETI project is that aliens will use a radio signal within the useful radio spectrum to communicate. Although the assumptions described above narrow down the search space significantly it is still extremely large. An added problem for the SETI project is that no one knows exactly what the alien signal will look like, the possibility exists that a signal from an alien intelligence was already received, but that it was not recognized. Recently researchers at the SETI institute have been studying quantitative measures from information theory like entropy and zipf to see if these might be useful in their search for extra-terrestrial intelligence. Using the "language" of whales and dolphins they are attempting to find a better understanding of intelligence as a evolutionary adaption and possibly also new methods to analyze possible signals from intelligence on distant worlds[Hanser:2002][Richards:2004].

## 3.5 Some notes on the collection of corpora

The following section will describe some issues that arise in collecting different types of corpora, and what attempts have been made to collect these types of corpora. In order to determine whether or not methods used in

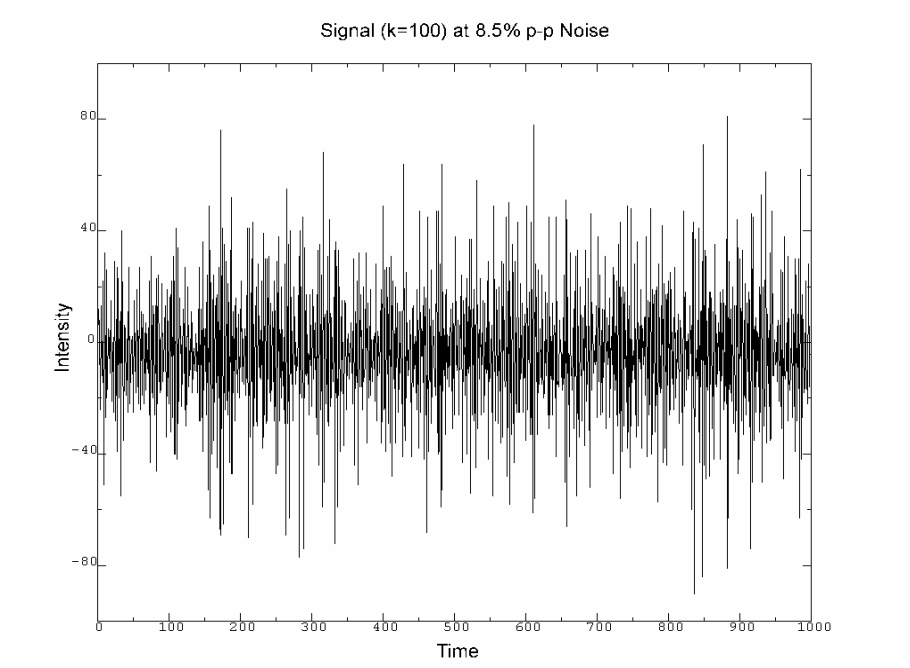


Figure 3.4: An example of a radio signal received by SETI, in this case the signal turned out to be nothing more than background noise.

linguistical analysis can also be used in other forms of communication and data, a first step has to be the collection of corpora containing such data. Preferably these corpora should be in a form which makes it easy to apply the methods described in the earlier chapters. The basic feature these methods all have in common is the fact that they can be applied to human natural language texts, with tokens being the basic elements which make up the texts. Because of this a corpus should preferably be plain text with clear rules as to what a basic element is. Another thing that the methods described earlier have in common is that they produce more accurate results given larger quantities of data. Having not enough data will produce results which could be very inaccurate.

### 3.5.1 Problems with animal corpora

A major problem when one wants to perform quantitative research on animal communication is the lack or inaccessibility of animal corpora of sufficient

size. Collecting a corpora of animal communication is generally a slow and expensive process. Specialized equipment is often needed to record animal sounds or gestures and after the process of collection is completed the collected data has to be analyzed by experts who should then convert it to a workable form. This is the main reason for the lack of large animal corpora; the corpora that *do* exist contain small quantities of data usually from a small group of animals.

Another problem with corpora containing animal communication is determining from which elements animal communication is made up. If a method is to be used which requires a word frequency list of a human corpus then it should be possible to define a similar structure from the animal data. For dolphins, several methods of classifying their whistles into categories exist. These methods vary from letting human experts classify the whistles to letting a computer distinguish them. Unfortunately the different methods do not always classify the same whistles into the same categories, one of the reasons again seems to be the small size of the corpora [Janik:1999]. A possible solution to the lack of objective comprehensive repertoire models is using self-organizing neural networks to handle the classification [Murray:1998].

### 3.5.2 DNA

Collecting a corpora of DNA is relatively easy because the human genome and complete DNA analysis of other creates have been made publicly available[3]. It would therefor be possible to apply the linguistic methods which were discussed earlier to DNA.

### 3.5.3 Artificial language corpora

#### Universal Decimal Classification system

The universal decimal classification system (UDC) is a multilingual classification scheme that can be used to index and later retrieve documents on any subject. It was derived from the decimal classification system of Melvil Dewey[4] at the beginning of the 20th century [UDC:2001]. UDC was used extensively in libraries to classify the contents of books. UDC can be seen as an artificial language, it describes the contents of a document using a set of numbers.

[3]The human genome is available at project Gutenberg; http://www.gutenberg.org/.
[4]The Dewey Decimal classification system was first published in 1876

### Computer languages

Languages created by humans to program computers are called computer languages. These languages vary from very low-level machine language and assembly to higher level languages like C++ and Delphi. As many as 2500 dialects, variants, versions and implementations of computer languages have been cataloged [ComputerLanguageList:1991]. Collecting corpora of computer languages varies in difficulty, depending on the language being collected; some languages have been widely used while others were only used in one single project. One easy way of creating a corpus is by collecting source code published under the GNU license [GNUProject:2004]. The source code of software published under the GNU licensed is always included with the program, sources for operating systems like Linux, and C compilers are all available. Because of the large diversity in computer languages it would be interesting to see what differences exist between different languages, especially between older, relatively low level languages like Fortran and newer, higher level languages like C++. A problem that does arise when one wants to collect different kinds of programming languages is the fact that not all of them are available in the same quantity. While popular languages like C which are still used a lot are widely availably, source code written in Fortran or assembly are much harder to find. This can be largely be explained due to the fact that when languages like Fortran were commonly used the internet was still in it's early stages of development and much less software was made available to the general public.

### Other artificial languages

A natural language is defined as being a language which has been learned by humans as a mother language; learned in the critical period roughly between birth and puberty when language acquisition occurs naturally. Besides computer languages humans have also designed other none natural languages. Examples of these are Klingon[5] and the Elvish language[6]. Another example of an artificial language is the language called simplified English which was used by the Fokker company in manuals for airplanes. Simplified English is based on English however, simpler words than in the original manuals were used, and complex structures like passive sentences were avoided so mechanics with only basic English skills could read and understand the manuals.

[5]Created by fans of the television show Startrek
[6]Created by fans of the books by J.R.R. Tolkien

# Chapter 4

# The data sets

This chapter describes the steps that were taken to collect and prepare the corpora for the experiments that will be described later on in this thesis. Additionally the properties of the corpora will be explored and the reasons why certain choices were made during the preparation of the different corpora will be explained.

## 4.1 English corpus

To use as an example to determine the minimum amount of tokens needed to create a accurate Zipf plot the book War and Peace by Leo Tolstoy was selected to serve as a corpus for the English language. A natural language was chosen because it is known that the Zipf phenomena occurs in languages like English. For the other experiments involving natural language the book Moby Dick was by Herman Melville was used, as this book was also used by other scientists for Zipf related research[Ferrer:2002].

## 4.2 C++ corpus

### 4.2.1 Description of the corpus

The C++ corpus that was collected to be used in the various experiments consists of all C and C++ files[1] that make up the source code for GCC[2]. GCC is a open source compiler for C and C++ and is widely used across

[1] .c, .h and .cpp files
[2] Version 3.2.1

the world. The corpus consists of 6938 different files with a total size of 37.2MB.

### 4.2.2 Preparation of the C++ corpus

The first step that has to be taken to prepare the C++ corpus for use in our experiments is the removal of all the comments in the source code. These sections of commentary usually consist of natural language and are also removed by the compiler when the program is translated into machine language. Comments are not considered to be a part of the programming language. The next step in the preparation process is the tokenization of the source code. In C++ a code fragment like: `for(x=0;x<y;x++);` has the same meaning as: `for ( x = 0 ; x < y ; x++ ) ;` Therefor it is important that things like brackets and semicolons are identified as separate tokens. For our experiments all brackets ( `'('`,`')'`, `'<'`, `'>'`, `'['`, `']'` ), and the C/C++ operators are seen as separate tokens.

### 4.2.3 Identifiers and Variables

In most programming languages, including C++, programmers have the option to name their variables and functions so the source code becomes more readable. For example a variable that may be used to iterate trough a loop could be called 'Iterator'. Often these variables have names that are derived from natural languages. Because of this it could be argued that these identifiers should be removed from the list of C++ "words" or at the very least given a unique name avoid natural language polluting the programming language, and interfering with the experiment. However, unlike comments, identifiers and variables are an integral part of the programming language, and although they are renamed internally by the compiler when the translation to machine code is made, excluding them would result in a loss of information. In natural language we refer to objects everyday. For example when we use "John" or "Mary" in a conversation we are referring to a certain person. If we use the word John in a completely different situation we might be referring to a completely different person. Therefor, when calculating the Zipf slope or Entropy of a natural language we do not replace every reference to a object by a unique identifier, we are interested in words that are being used and not the objects that are being referred. Similarly in programming languages variables with the same name in different modules could be referring to the same piece of data and therefor it would mean a loss of data if the variables were to be renamed. Although it is true that

variables like "i" are often used to describe different variables that do not relate to each other in any way, the same is also true for natural languages. For example the word "the man" could be used to refer to billions of different objects but when calculating entropy or the Zipf curve or entropy for a language is still counted as one word.

### 4.2.4 Result of Source Code Preparation

After the source code has been prepared by removing comments and tokenizing whatever is left over the final result consists of 32.0MB of C/C++ code, containing 126142 types of tokens.

## 4.3 Fortran Corpus

Because the Fortran programming language is not as popular as it used to be it is much harder to find Fortran source code in sufficient amounts. Because Fortran code that had a similar function as the code in the C++ corpus (Compiling code) other types of programs were collected from various source. After tokenization of the complete Fortran corpus, with the exclusion of any comments, the Corpus has a total size of 29.2MB containing 34246 token types.

## 4.4 Assembly and Bitcode Corpora

The assembly corpus consists of a total of 7.01MB of compiler related assembly code. To prepare the corpus for the experiments described later on in this thesis only that comments had to be removed. This yielded a final corpus size of 4.81MB. The bitcode corpus consists of all the binary files included in the Borland Delphi 7 compiler and IDE. No further preparations had to be made for this corpus.

## 4.5 Size of Corpora

### 4.5.1 Determining the minimum corpus size for a Zipf plot

The Zipf measure strongly depends on the amount of data available. As the Zipf plot actually depends on the frequency of words in a corpus, selecting a corpus that is too small will generate inaccurate results. To make sure that the corpora which were collected for the various experiments described

Table 4.1: Corpora sizes

| Name | Size | Types | Tokens | Percentage of Total Corpus |
|---|---|---|---|---|
| C++ Complete | 32.0MB | 126142 | 5778504 | 100% |
| C++ Keywords | - | 250 | 3257590 | 56.37% |
| C++ No Keywords | - | 125892 | 2520914 | 43.63% |
| Fortran Complete | 29.2MB | 34246 | 2357688 | 100% |
| Fortran Keywords | - | 259 | 1409842 | 59.80% |
| Fortran no Keywords | - | 33987 | 947846 | 40.20% |
| Assembly Complete | 4.81MB | 18634 | 574770 | 100% |
| Bit code | 39.5MB | 256 | 13513044 | 100% |
| War and Peace | 3.11MB | 17503 | 572180 | 100% |
| Moby Dick | 2.29MB | 16983 | 523286 | 100% |
| Moby Dick Function Words | - | 215 | 123231 | 23.55% |
| Moby Dick no Function Words | - | 16768 | 400055 | 76.45% |

in this thesis are of sufficient size a Zipf plot was created of a corpus of English, in this case the book War and Peace. Furthermore, Zipf plots were created of progressively smaller parts of the book. When a certain plot deviates strongly from the complete corpus plot this is an indication that the selected part of the corpus has become to small too yield accurate results. The sizes of the various corpora used in the experiments which will be described in the next chapters can be viewed in table 4.1.

# Chapter 5

# Experiments

## 5.1 Natural language, Zipf curve bulge and two groups of words

To compare the artificial programming languages to a natural language, the obvious procedure is to perform the same actions on a the text as on the programming languages. We shall use the book Moby Dick in these first experiments, the same book was used in other Zipf related experiments by other researchers [Ferrer:2002]. As the words in Moby Dick have to be separated into two different groups, the contents of each group has to be decided. As the first group should contain words that are used on a daily basis it would be a logical choice to select function words and other frequently used supporting words into this group. The words that were selected to make up the A lexicon, can be separated in several categories[Leech:2001][Ferrer:2001b].

- Pronouns and determiners, like "ourselves", "your" and "such".
- Prepositions, like "during", "before" and "depending on".
- Conjunctions, like "and", "though" and "albeit".
- Interjections and discourse particles, like "yes", "goodbye" and "oh".
- Commonly used verbs, like "have" and "be".[1]

[1]These words lists can be downloaded from the companion website for Word Frequencies in Written and Spoken English: based on the British National Corpus[Leech:2001]: `http://www.comp.lancs.ac.uk/computing/research/ucrel/bncfreq/`

## 5.2 Programming languages, Zipf curve bulges and two groups of words

Before the existence of two distinctly different types of words can be assumed in a natural language corpora, by means of comparing it to a programming language, is showing that a bulge also exists in the programming language's Zipf plots. The second part of the experiment will be to separate the Zipf plots into the A and B lexicon. The experiments described here will be performed on the Fortran and C++ corpora. These two corpora were selected because they both have a large number of distinguishable keywords and operators and because they offer sufficient possibilities for a programmer to add custom variable and function names. For the C++ corpus the keywords are defined as all functions described in the C99 standard[C9X:2000] and the standard library; the Ansi Fortran 90 standard was used to extract the keywords and functions from the Fortran corpus[Chivers:2005]. One could argue that the keywords make up the largest part of the most frequently occurring words in the corpus, so simply selecting the 200 most frequent words would have the same effect. However, this is not the case: Many keywords are not frequent at all, while certain other words like the variables "X" and "foo" are extremely frequent. The C++ function words are to the C++ corpus what the frequently used verbs like have and be are for the English corpus. Because a clear separation into an A and B lexicon for the assembly and bitcode corpora is not easily done only the properties of the full corpora were examined in this thesis.

## 5.3 Zipf in random corpora

Intuitively it is obvious that a random text does not contain two different type of word groups. After all, all words in the text are randomly generated and therefor they are all equal. To prove that this is indeed the case and the subsequent bulge in the Zipf plot are only inherent to data with meaning, randomly generated corpora will be created. Zipf plots of these corpora will be made and then analyzed for the existence or absence of such a bulge. To prevent a inaccurate result due to word length variations in the random and other corpora, the random corpora will consist of the same number of words, with the same word length, as their non-random counterparts [Ferrer:2002].

### 5.3.1 Determining the rough minimum corpora size for a Zipf plot

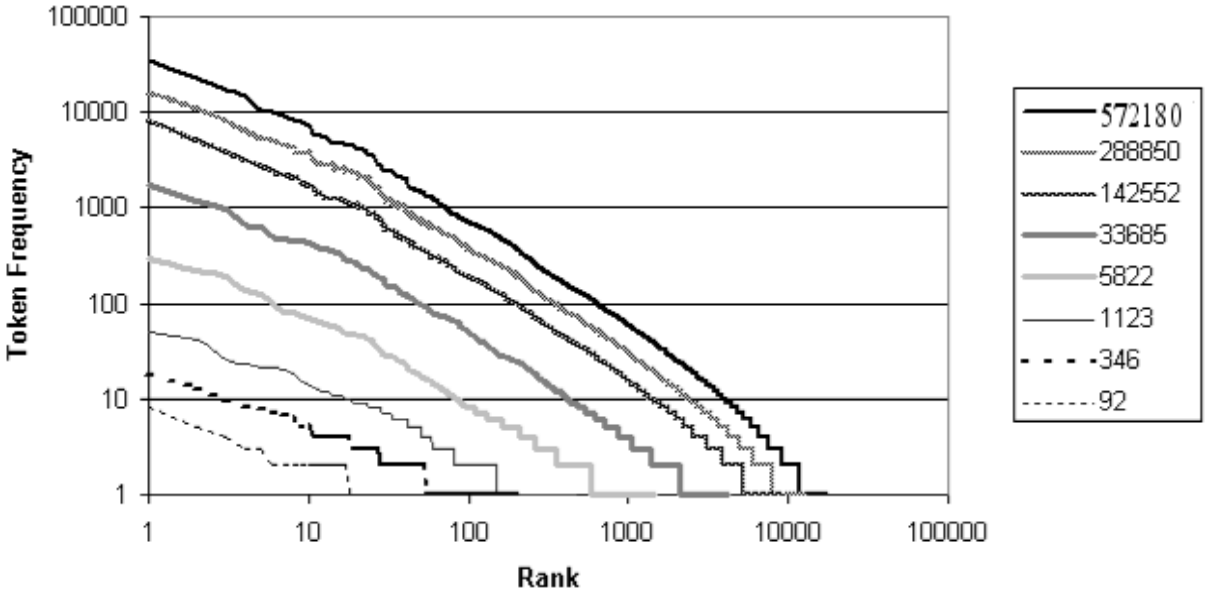


Figure 5.1: Zipf curves for selecting increasingly smaller numbers of words from War And Peace.

Figure 5.1 shows the result for the experiment that was done to determine the minimum size of a corpora needed to draw a accurate Zipf plot. Although a clear threshold is not visible in the diagrams the smaller corpora sizes progressively produce more erratic Zipf plots. However, all plots of 142552 words and more do not show any noteworthy differences from each other, and therefore it can be safely assumed that selecting a corpus of at least 150000 words should be sufficient to create accurate Zipf plots.

### 5.3.2 Zipf curve bulge and two groups of words in English

As can be seen in figure 5.2A the presence of two separate word groups is not immediately apparent. This is mostly due to the fact that the word 'the' occurs very frequently when compared to the other words in the top 10 of most frequently occurring words. The high frequency of this single word causes the sharp angle right at the start of the plot. If one were to remove that word from the frequency list then the presence of two slopes

and a bulge in the Zipf plot would become much more visible. Figure 5.2B shows the plot for Moby Dick with all function words removed. As can be seen the plot follows an ideal Zipf distribution quite nicely. Finally, figure 5.2C shows that when only using function words in a plot, that plot will not resemble a Zipf distribution anymore.

### 5.3.3 Zipf curve bulge and two groups of words in C++

As can been seen in figure 5.3A the C++ corpus shows a clear bulge in its Zipf plot. The first part of the graph is almost horizontal, then suddenly drops and then continues on a more steady line. The more abrupt change in this plot when compared to the plot of natural languages can be explained, if we assume the presence of two word groups, by the fact that the division in an programming language in an A and B lexicon is much more clear than it would be in a natural language. Figures 5.3B and 5.3C show what happens if you split the C++ keywords from the rest of the corpus. The plot of the keywords clearly no longer shows any resemblance to a standard Zipf plot with a slope of -1. However, the Zipf plot of the corpus where the C++ keywords had been removed has become an almost perfect, straight line following the slope that Zipf's first law predicts.

### 5.3.4 Zipf curve bulge and two groups of words in Fortran

In the three fortran Zipf plots in figure 5.4 the same effects as in the plots for the C++ corpus which were discussed in the previous section can be seen. As was the case with the C++ corpus the Zipf plot of the Fortran corpus where the keywords were removed adheres much closer to Zipf's law, while the plot showing only the keywords no longer resembles a straight line.

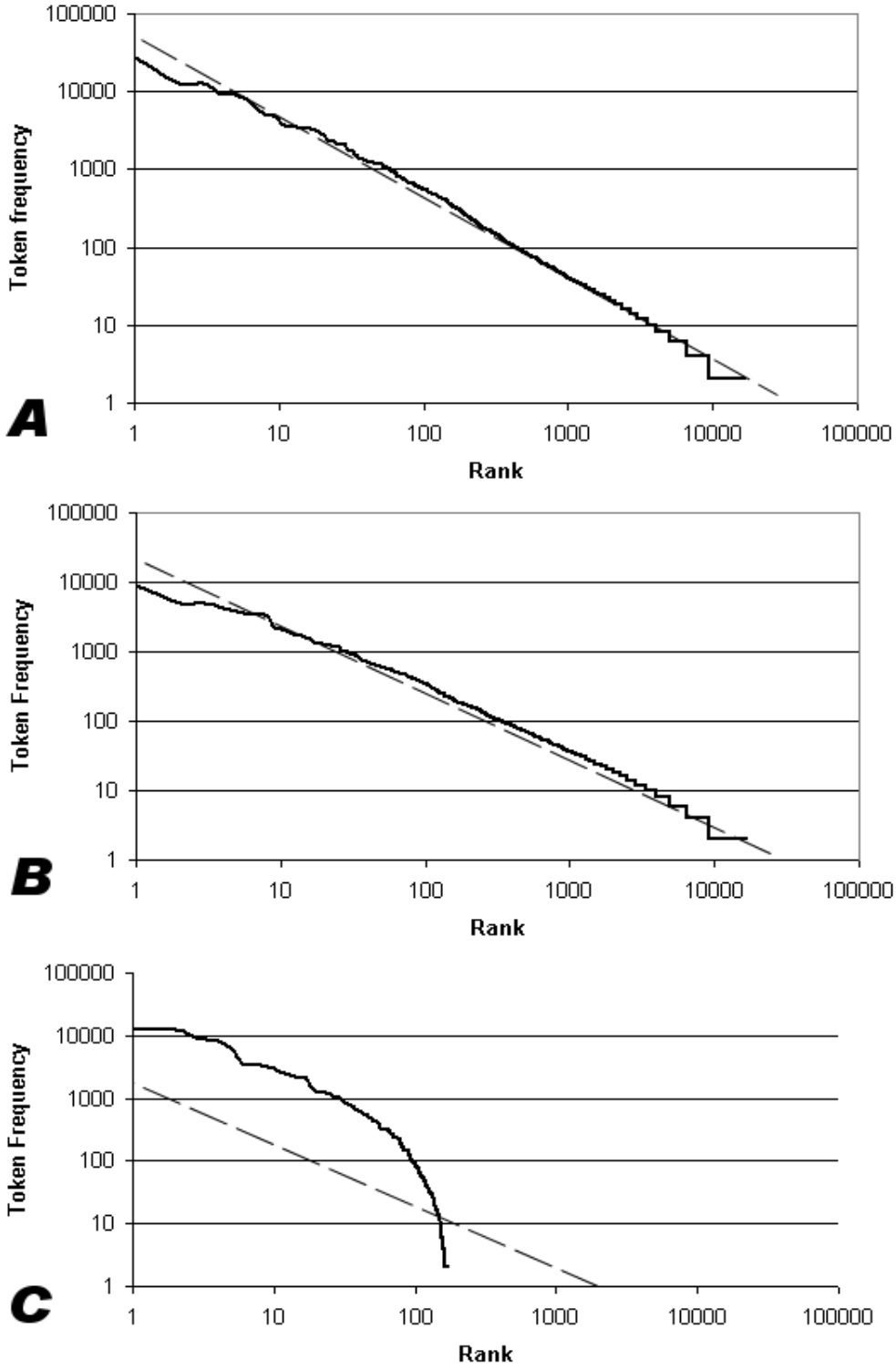


Figure 5.2: A) Word frequency distribution on a log-log scale for English. The straight line shows an ideal Zipf distribution. B) Word frequency distribution for the same corpus, but with all function words removed. C) Word frequency distribution on a log-log scale for the function words in English.

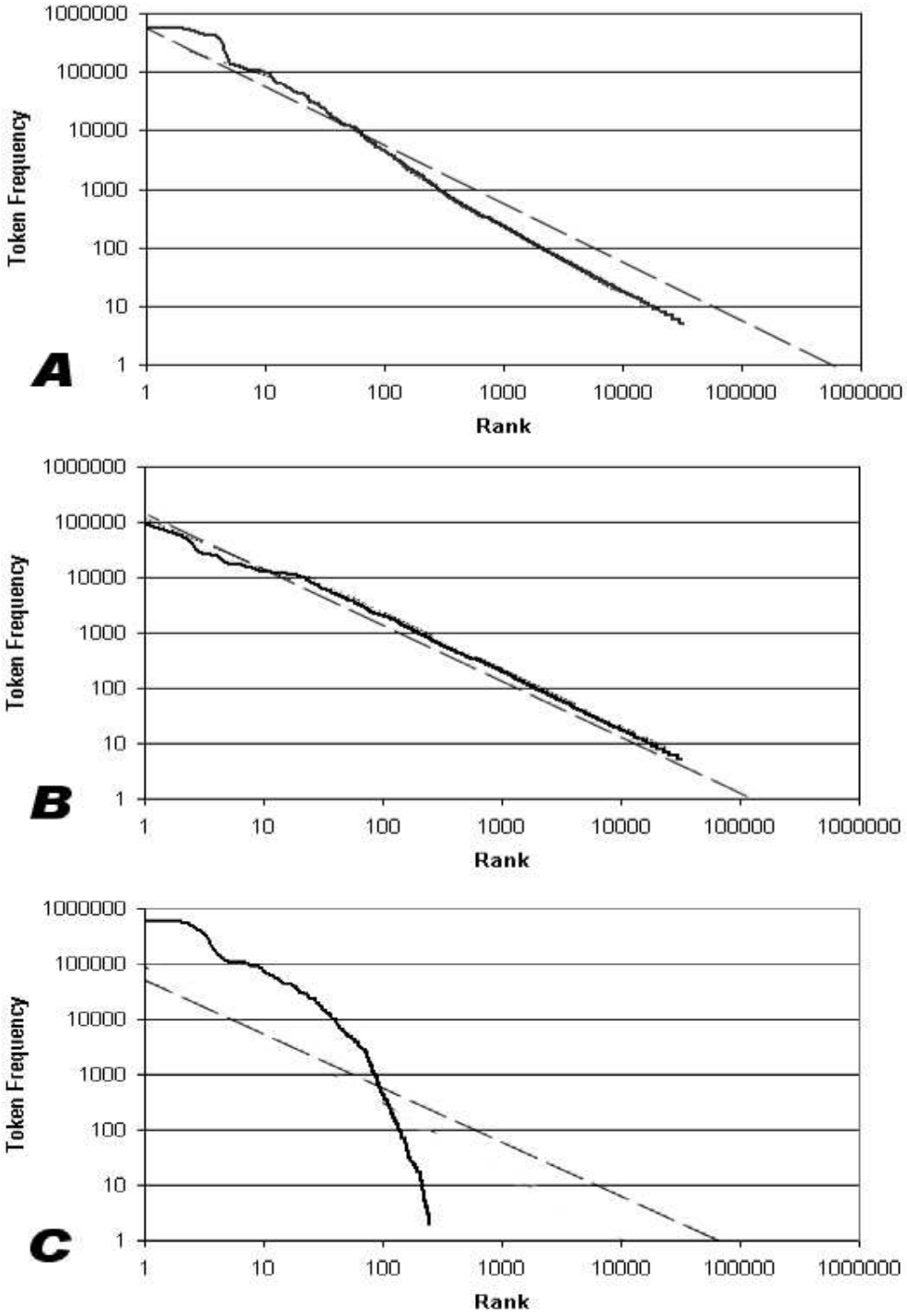


Figure 5.3: A) Word frequency distribution on a log-log scale for the C++ corpus. The straight line shows an ideal Zipf distribution. B)Word frequency distribution on a log-log scale for the C++ corpus, with all C++ keywords removed. The straight line shows an ideal Zipf distribution. C)Word frequency distribution on a log-log scale for the C++ corpus, with everything but the C++ keywords removed. The straight line shows an ideal Zipf distribution.

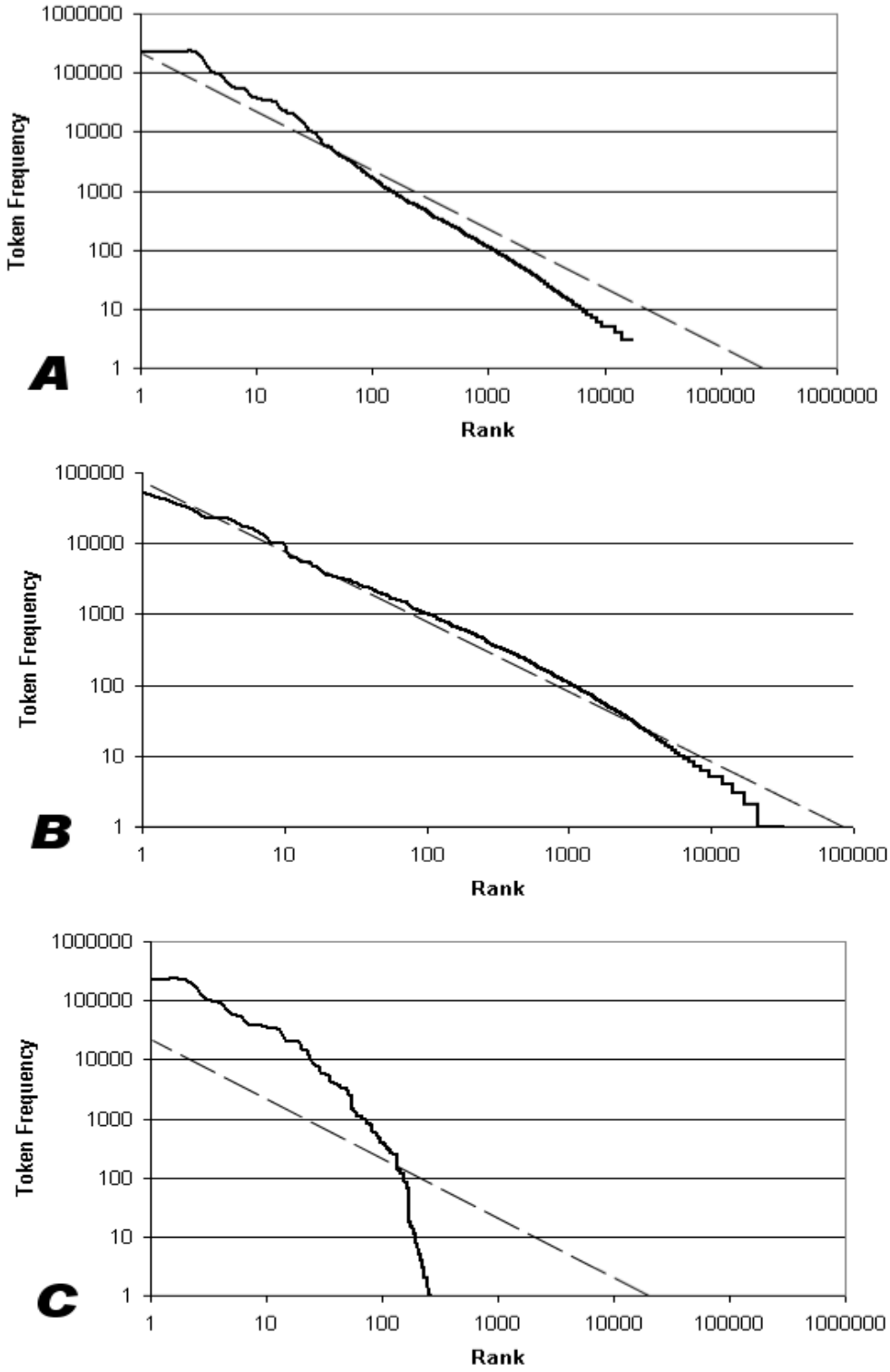


Figure 5.4: A:Word frequency distribution on a log-log scale for the Fortran corpus. The straight line shows an ideal Zipf distribution. B:Word frequency distribution on a log-log scale for the Fortran corpus, with all Fortran keywords removed. The straight line shows an ideal Zipf distribution. C:Word frequency distribution on a log-log scale for the Fortran corpus, with only the Fortran keywords. The straight line shows an ideal Zipf distribution.

Table 5.1: Average Zipf plot slope for the various corpora

| Name | Rank Start | Rank End | Average Slope |
|---|---|---|---|
| Moby Dick Total | 1 | 16983 | -0.95 |
| Moby Dick Total Head | 1 | 250 | -0.68 |
| Moby Dick Total Tail | 250 | 16983 | -1.08 |
| Moby Dick Lexicon B | 1 | 16768 | -0.77 |
| CPP Total | 1 | 126142 | -1.15 |
| CPP Total Head | 1 | 90 | -0.65 |
| CPP Total Tail | 90 | 126142 | -1.21 |
| CPP Lexicon B | 1 | 125892 | -0.92 |
| Fortran Total | 1 | 34246 | -1.22 |
| Fortran Total Head | 1 | 85 | -0.62 |
| Fortran Total Tail | 85 | 34246 | -1.47 |
| Fortran Lexicon B | 1 | 33987 | -0.97 |

### 5.3.5 The bulge examined more closely

Table 5.1 shows the various angles of inclination for the corpora that were separated into an A and a B corpus. The large differences in the inclination rates of the heads (the part of the graph before the bulge) and tails (the part after the bulge) of the plots for the complete corpora are especially striking. The differences in inclines between the two parts of the graphs for the Moby Dick, C++ and Fortran corpora are -0.40, -0.50 and -0.60 respectively. These differences in slope clearly show that a bulge is present in all three diagrams.

### 5.3.6 Zipf in random corpora

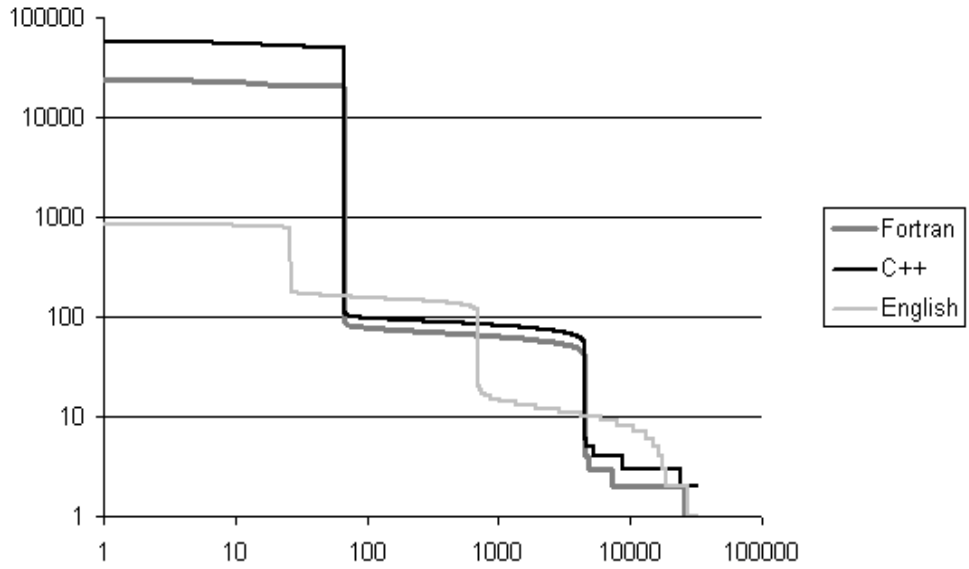


Figure 5.5: Zipf plots for three randomly generated corpora.

As can be seen in figure 5.5 the plots of the corpora that were created from the English corpus and two of the programming language corpora don't show a bulge. Instead they all show step-like behavior, which can be explained by the random nature of the corpus. Because character frequencies in the different corpora are totally random, and all characters have the same chance of occurring words of the same length have the same frequency. The first part of the Zipf plots all consist of words of one 'letter' the second step are those words consisting out of two letters and so on. The reason that the 'steps' of the plot for the English corpus are less wide can be explained by the fact that the programming languages use a larger collection of characters, so their random corpora have a larger vocabulary. For example characters like "(" or "=" would not occur in a novel like War and Peace while they'de be seen frequently in a piece of source code.

### 5.3.7 Zipf in low-level programming languages

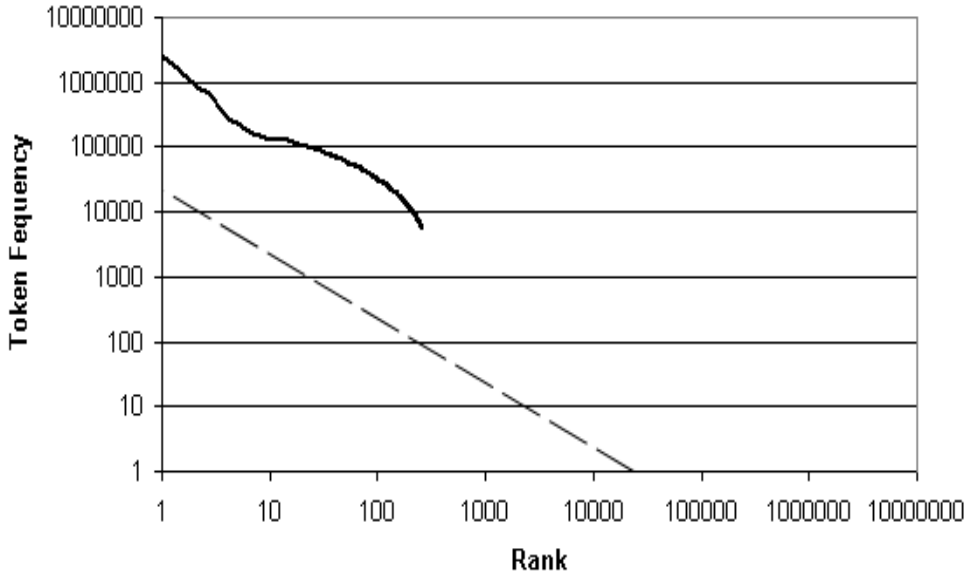


Figure 5.6: Word frequency distribution on a log-log scale for the 32-bit corpus.

As can be seen in figure 5.6 the Zipf plot of the bitcode corpus does not resemble those of the other programming languages even though it resembles a straight line. One of the main causes of this is that the bitcode has a very limited lexicon. The assembly corpus however, already resembles greatly the other programming languages. A difference that can be seen in the Zipf plot for the assembly corpus though is that it seems to have multiple bulges. Starting with a straight line the plot continues less steeply after the first bulge, only to bulge once more to continue on in slightly steeper fashion.

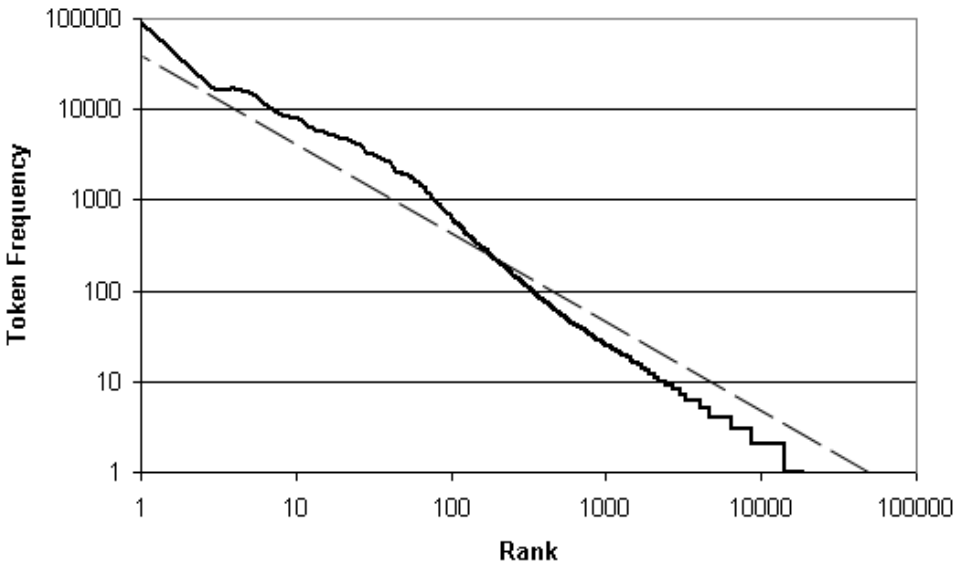


Figure 5.7: Word frequency distribution on a log-log scale for the Assembly corpus.

# Chapter 6

# Conclusion

The various Zipf plots of the different programming languages discussed in this thesis show that programming languages also adhere to Zipf's first law. The exception is the binary corpus, but this is most likely due to the limited size of the binary lexicon. When comparing the plots in figures 5.3A, 5.4A and 5.7 there does not seem to be a discernable difference between the three. This indicates that the change or progress in programming languages as they have developed over the years cannot be shown by the use of Zipfian measures. When separating the function words from the other words in the English text it is clear that the function words by themselves do not show Zipfian behavior.

The remaining words in the English text however, are even more Zipfian. The same is true for the C++ and Fortran programming languages, when the keywords are split from the rest of the corpus the keywords no longer follow Zipf, while the rest of the corpus does. For the programming language corpora it is also clear that after the Keywords are removed the rank-frequency plot more closely follows a slope of -1. This effect although present is less clear in the natural language text. One possible reason to explain this is the fact that it is much harder to define what an actual function word is and what is not, for programming languages the difference between the two groups can be easier to distinguish, if one follows the standards that were created for programming languages to select the different word groups. Secondly the text that was used in the experiment, Moby Dick was written by one author about one subject. Words like 'whale' are very frequent in the book, so frequent that it approaches the levels of high frequency function words. These two issues could interfere with getting a clear result. A major difference between random texts and natural and programming languages is

the fact that random texts do not have different word groups.

To conclude, we have established that the results of the experiments described in this thesis show that it is indeed likely that the different word groups in both natural, and artificial languages like programming languages reveal themselves as a slight curve in the plot of a Zipf diagram for a complete text. From the above it can be concluded that the original hypothesis as defined in the introduction still holds true. It does indeed seem likely that communicative texts, both of natural and artificial nature, consists of two different lexica. These lexica in turn cause the bulge in Zipf plots. When separating the lexica the first group shows no resemblance to a Zipf plot while the second group does.

In the section about DNA an experiment was described were Zipf plots were created for DNA that was also split into two groups; coding and non-coding DNA. Like in the experiments described here earlier one part of DNA did follow Zipf's first law while the second group did not (see figure 3.3). This raises the question if a Zipf plot could be used to determine wether or not an unclassified piece of data is random noise, or actually contains some form of meaning. Although this thesis does not claim to hold the answer to that question the experiments described here do show that texts containing communicative data could be separated from random texts by using Zipfian measures.

Finally lets consider figure 2.1 again. Although the AOL user data is neither a language or even a communication environment the bulge in the Zipf plot is still present. This leads to the hypothesis that other collections of data cited in articles about Zipf's law, also consist of two 'lexica'. For the AOL data this could for example be the presence of two user groups in the data. A possible next step for future researchers would be to examine wether the phenomena of a bulge in a Zipf plot in other types of data can also be explained by the presence of two lexica, or two types of data equivalent to lexica.

# Bibliography

[Adamic:2002] Lada A. Adamic. Zipf, power-laws, and pareto - a ranking tutorial. `http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html`, 2002. Last visited: November 22, 2004.

[Aristotle:330BC] Aristotle. Historia animalium, 330 BC.

[Baayen:1993] Harald Baayen. Statistical models for word frequency distributions: A linguistic evaluation. *Computers and the Humanities*, 26:347–363, 1993.

[C9X:2000] Thomas Wolf. The new iso standard for c (c9x). `http://home.tiscalinet.ch/t_wolf/tw/c/c9x_changes.html`, Augustus 2000. Last visited 12 January, 2005.

[Carol:1967] John B. Carroll. On sampling from a lognormal model of word-frequency distribution. *Computational Analysis of Present-Day American English*, pages 406–424, 1967.

[Chivers:2005] Ian Chivers and Jane Sleightholme. Fortran 90, 95 and 2003 home page. `http://www.kcl.ac.uk/kis/support/cit/fortran/f90home.html`, January 2005. Last visited 17 January, 2005.

[Cohen:1996] A. Cohen, R.N. Mantegna, and S. Havlin. Can zipf analyses and entropy distinguish between artificial and natural language texts?, January 1996.

[ComputerLanguageList:1991] Bill Kinnersley. The language list. `http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm`, 1991. Last visited: April 18, 2004.

[Ferrer:2001a] Ramon Ferrer i Cancho and Ricard V. Sole. Two regimes in the frequency of words and the origins of complex lexicons: Zipfs

law revisited. *Journal of Quantitative Linguistics*, 8(3):165–173, 2001.

[Ferrer:2001b] Ramon Ferrer i Cancho and Ricard V. Sole. The small world of human language. In *Proceedings Of The Royal Society Of London Series B Biological Sciences*, volume 268, pages 2261–2265, November 2001.

[Ferrer:2002] Ramon Ferrer i Cancho and Ricard V. Sol. Zipf's law and random texts. *Advances in Complex Systems*, 5(1):1–6, 2002.

[Ferrer:2003] Ramon Ferrer i Cancho and Ricard V. Sole. Least effort and the origins of scaling in human language. In KennethW.Wachter, editor, *PNAS*, volume 100, pages 788–791. National Academy of Sciences, February 2003.

[Ford:2002] John K.B. Ford. *Encyclopedia of Marine Mammals*, chapter Killer Whale, page 669675. Academic Press, 2002.

[Frisch:1967] Karl von Frisch. *The Dance Language and Orientation of Bees.* Belknap Press of Harvard University Press, 1967.

[GNUProject:2004] Richard Stallman. The gnu project. `http://www.gnu.org/gnu/thegnuproject.html`, 2004. Last visited: April 18, 2004.

[Hanser:2002] Sean F. Hanser, Laurance Doyle, Jon Jenkins, and Brenda McCowan. Information theory applied to animal communication systems and its possible application to seti. Bioastronomy 7, 2002.

[Janik:1999] Vincent M. Janik. Pitfalls in the categorization of behaviour: a comparison of dolphin whistle classification methods. *Animal Behaviour*, 57:133–143, 1999.

[Laherrere:1997] Jean Laherrere. Multi-hubbert modeling. `http://www.oilcrisis.com/laherrere/multihub.htm`, 1997. Last visited 15 February, 2005.

[Leech:2001] Geoffrey Leech, Paul Rayson, and Andrew Wilson. *Word Frequencies in Written and Spoken English: based on the British National Corpus.* Longman, London, 2001.

[Li:1999] Wentian Li. Zipf's law. `http://linkage.rockefeller.edu/wli/zipf/`, January 1999. Last visited 15 February, 2004.

[Mantegna:1994] R.N. Mantegna, S.V. Buldyrev, A.L. Goldberger, S. Havlin, C.-K. Peng, M. Simons, and H.E. Stanley. Linguistic features of non-coding dna sequences. *Physical Review Letters*, 73(2):3169–3172, December 1994.

[McCowan:1999] Brenda McCowan, Sean F. Hanser, and Laurence R. Doyle. Quantitive tools for comparing animal communication systems: information theory applied to bottlenose dolphin whistle repertoires. *Animal Behaviour*, 57(2):409–419, 1999.

[McCowan:2002] Brenda McCowan, Sean F. Hanser, and Laurence R. Doyle. Using information theory to assess the diversity, complexity, and development of communicative repertoires. *Journal of Comparative Psychology*, 166(2):166–172, 2002.

[Miller:1954] G.A. Miller. Communication. *Annual Review of Psychology*, 5:401–420, 1954.

[Murray:1998] Scott O. Murray, Eduardo Mercado, and Herbert L. Roitblat. The neural network classification of false killer whale (pseudorca crassidens) vocalizations. *Journal of the Acoustical Society of America*, 104(6), December 1998.

[Niyogi:1995] Partha Niyogi and Robert C. Berwick. A note on zipf's law, natural languages, and noncoding dna regions. Technical report, Massachusetts Institute Of Technology and Center for Biological and Computational Learning, March 1995.

[Nollman:1999] Jim Nollman. *The Charged Border: Where Whales and Humans Meet*. Henry Holt, 115 West 18th Street, New York, 1st edition edition, 1999.

[Paijmans:2004] J.J. Paijmans. Building a linguistic corpus from bee dance data. *Proceedings of the first international congres of bioinformatics*, June 2004.

[Popescu:2003] Ioan-Iovitz Popescu. On a zipfs law extension to impact factors. `http://alpha2.infim.ro/~ltpd/Zipf_Law.html`, June 2003. Last visited 22 November, 2004.

[Richards:2004] Diane Richards. Unlocking language in space and on earth. `http://www.space.com/searchforlife/seti_richards_doyle_040422.html`, April 2004. Last visited 23 November, 2004.

[Rossie:2002] James Rossie. Saimiri sciureus, squirrel monkey. `http://digimorph.org/specimens/Saimiri_sciureus/338948/`, 2002. Carnegie Museum of Natural History, Last visited 13 September, 2004.

[SETI] SETI Institute. Seti website. `http://www.seti.org`. Last visited 23 November, 2004.

[Sebeok:1990] Thomas A. Sebeok. *Essays in Zoosemiotics*. Toronto Semiotic Circle, 1990.

[Seife:1999] Charles Seife. Deep message. *New Scientist*, 161(2175):24, February 1999.

[Shannon:1948] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:pp. 379–423 and 623–656, July and October 1948.

[Sichel:1986] H. A. Sichel. word frequency distributions and type-token characteristics. *Mathematical Scientist*, 11:45–72, 1986.

[Sinha:2001] Chris Sinha. The epigenesis of symbolization. *LundUniversity Cognitive Studies*, 85:85–94, 2001.

[UDC:2001] UDC Consortium. About universal decimal classification and the udc consortium. `http://www.udcc.org/about.htm`, 2001. Last visited 4 October, 2004.

[Vides:1992] Julio Collado-Vides. Grammatical model of the regulation of gene expression. In *Proceedings of the National Academy of Sciences*, volume 89, pages 9405–9409. Department of Biology, MIT, National Academy of Sciences, October 1992.

[Wenner:1967] A.M. Wenner. Honey bees: do they use the distance information contained in their dance maneuver? *Science*, 155:847–849, 1967.

[Wenner:2002] A.M. Wenner. The elusive bee dance "language" hypothesis. *Journal of insect behavior*, 15(6):859–878, November 2002.

[WikiDNA:2004] Wikipedia. Dna. `http://en.wikipedia.org/wiki/DNA`, November 2004. Last visited 22 November, 2004.

[WikiPoisson:2005] Wikipedia. Poisson distribution. `http://en.wikipedia.org/wiki/Poisson_distribution`, January 2005. Last visited 15 February, 2005.

[Zipf:1949] G. K. Zipf. *Human Behaviour and the Principle of Least Effort.* Addison-Wesley Press, 1949.