THE BEAUTY OF MATHEMATICS IN COMPUTER SCIENCE

Jun Wu

A Chapman & Hall Book

CRC Press
Taylor & Francis Group

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper

International Standard Book Number-13: 978-1-138-04960-4 (Paperback)
978-1-138-04967-3 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Wu, Jun, 1967- author.
Title: The beauty of mathematics in computer science / Jun Wu.
Description: Boca Raton, FL : Taylor & Francis Group, 2019.
Identifiers: LCCN 2018035719| ISBN 9781138049604 paperback | ISBN 9781138049673 hardback
Subjects: LCSH: Computer science--Mathematics. | Machine learning.
Classification: LCC QA76.9.M35 W84 2019 | DDC 004.01/51--dc23
LC record available at https://lccn.loc.gov/2018035719

Visit the Taylor & Francis Web site at

Contents

Foreword

A few years ago, I wrote the forewords for Dr. Jun Wu's Chinese editions of *On Top of Tides* and *The Beauty of Mathematics*. Since, I'm happy to learn that *The Beauty of Mathematics* was awarded the prestigious Wenjin Prize.

The Beauty of Mathematics in Computer Science, with its new name in the English edition, originates from a series of Google China blog articles by Google's senior staff research scientist, Jun Wu. Initially, the blog's editors were afraid that the subject matter would bore readers—that it was too abstract and esoteric. That fear was quickly dispelled. Through vivid and compelling language, *The Beauty of Mathematics in Computer Science* connects the history of mathematics to the emergence of its practical applications. Wu systematically introduces the reader to important mathematical bases from which modern science and technology arise. Complex ideas are made accessible to a wide audience, especially those interested in science and technology.

As I wrote in the preface to *On Top of Tides*, Wu is one of a rare kind, with both strong narrative abilities and deep insight into the development of modern technology. Among the researchers and engineers I know, Wu stands in a unique position to share effectively his knowledge with the general public. In *The Beauty of Mathematics in Computer Science*, Wu proves this point once again. From years of accumulated knowledge, Wu excels in his grasp of mathematics and information processing, introduced here as professional disciplines ranging from speech recognition and natural language processing to information search. From the origins of digital computing, to the mathematics behind search engines, and clever mathematical applications to search, Wu brings the essence of mathematics to life. In his writing, mathematics is not a bag of boring, abstruse symbols; rather, it drives fascinating technologies in our everyday lives. Indeed, through Wu's tale, the reader will inevitably discover the hidden beauty of mathematics.

Galileo once said that "mathematics is the language with which God has written the universe." Along the same line of thought, Einstein wrote, in an obituary to Amalie Emmy Noether:

> Pure mathematics is, in its way, the poetry of logical ideas ... In this effort toward logical beauty spiritual formulas are discovered necessary for the deeper penetration into the laws of nature.

From years of personal study in information processing and speech recognition, I deeply appreciate the fundamental role mathematics plays in all fields of science.

In the fifth century AD, Greek philosopher Proclus Diadochus quipped that "wherever there is number, there is beauty." Today, I sincerely recommend *The Beauty of Mathematics in Computer Science* to any friend interested in the natural sciences, scientific research, or life in general. Whether you study liberal arts or engineering, Wu's exposition on mathematics will help you appreciate the elegance and sublimity of our universe. The value of this book lies in both the author's familiarity with the subject matter and his active role in developing such technologies as a career. Wu not only explains why simple mathematical models can solve complex engineering problems, but also reveals the thought processes behind his colleagues' and his own work. Without experience in application, most scholars of pure mathematics cannot achieve the latter point.

From the original Google China blog articles to the publication of *The Beauty of Mathematics in Computer Science*, Wu has spent much time and effort on this book. During his spare time from work, he rigorously rewrote most of the articles, so that an ordinary reader could understand and enjoy the material, but an expert would still learn much from its depth. Since the first version, Wu has encapsulated two more years of research at Google into two new chapters. In this edition, I hope that the reader will further appreciate the beauty of mathematics.

Sometimes, I find that today's world is under a lot of pressure to be practical, and in the process, has lost some of its curiosity about the natural world. In this respect, Wu's book is an excellent remedy. I very much hope that in the future, Wu will continue to write such books, elucidating complex ideas in simple terms. These ideas are the best gifts he could bestow upon society and the younger generation.

Kai-Fu Lee

Preface

The word “mathematics” stems from the Greek word, μα´ϑημα, which means “wisdom gained from learning.” In that sense, early mathematics encompassed a wider field of study and related more closely to people’s everyday lives.

Early mathematics was less mysterious and more practical. Like any field, mathematics undergoes constant change, becoming deeper and more theoretical in the process. In fact, the evolution of mathematics corresponds to the constant abstraction of concrete experiences and their effects on our lives. Today, several millennia have stripped mathematics down to numbers, symbols, equations, and theorems—quite detached from our everyday lives. People might use arithmetic to calculate the amount to tip at a restaurant, but beyond those basics, many see little value in mathematics, especially pure mathematics. After graduation, most college students will never encounter advanced mathematics again, so after a few years, they forget most of what they have learned. As such, many question the point of learning mathematics in the first place.

Even worse, many mathematicians have difficulty making a living from pure mathematics, both in the United States and China. A common stereotype portrays all mathematicians as hopeless “nerds,” with thick-rimmed glasses and poor social skills. To the layperson, why doom yourself to the socially unattractive label of “mathematician”? Thus, neither the abstruse numbers and symbols, nor the peculiar folk who study them, evoke common signs of beauty.

On the contrary, mathematics is far more prevalent in our lives than we may think. Initially, we might consider only scientists as direct consumers of mathematics—that is, of course studying atomic energies or aerodynamics would require mathematical knowledge! If we look deeper, though, technologies we use every day are all constructed from mathematical building blocks. When you ask Siri for today’s weather, mathematical gears whir into action. As a researcher for over twenty years, I have often marveled at the ways mathematics can be applied to practical problems, which are solved in a way I can only describe as magic. Therefore, I hope to share some of this magic with you.

In ancient times, the most important skill people developed, besides knowledge of the natural world, was to exchange ideas, or broadly, to communicate. Accordingly, I have selected *communication* as the starting point of this book. Communication is an excellent field to illustrate the beauty of mathematics, for not only does it abundantly adopt mathematics as a tool, but it also ties closely to our everyday lives.

Since the Industrial Revolution, communication has occupied a large percentage of time in people's lives. Moreover, after the advent of electricity, communication has both closed the distance between people and accelerated the world's economic growth. Today, it is commonplace to spend copious amounts of time lounging in front of a TV or watching Netflix, browsing the latest social media or posting photos from a smart phone. These are all offshoots of modern communication. Even activities that traditionally involved physically traveling somewhere, like shopping, have been overtaken by e-commerce sites—again, modern communication technologies. From century-old inventions, like Morse's telegraph and Bell's phone, to today's mobile phones and Internet, all modern implementations of communication have adhered to information theory principles, which are rooted in mathematics. If we search further back, even the development of language and writing were based on mathematical foundations.

Consider some of our everyday technologies: omnipotent web search, locating the tastiest restaurants and your local DMV; speech recognition on a smart phone, setting a reminder for laundry; or even online translation services, preferably not for your child's foreign language homework. To an everyday user, it is not immediately evident that mathematics drives all these seemingly magical features. Upon further inspection, however, these diverse applications can all be described by simple mathematical models. When engineers find the appropriate mathematical tool for some hairy problem, they will often bask in the elegance of their solution. For instance, although there are hundreds of human languages, from English to Swahili, the underlying mathematical models for translating them are the same, or nearly so. In this simplicity lies beauty. This book will introduce some of these models and demonstrate how they process information, especially to bring about the technological products we use today.

There is often an aura of mystery about mathematics, but its essence is uncomplicated and straightforward. English philosopher, Francis Bacon, once quipped, "Virtue is like a rich stone, best plain set." Mathematics is precisely this type of virtue. Thus throughout this book, I will attempt to portray that simplicity is beauty.

Finally, I provide a brief explanation for the book's extensive treatment of natural language processing ideas and in particular, its experts. These world-class scholars hail from a diverse set of nationalities or backgrounds, but they share a common love for mathematics and apply its methods towards practical problems. By recounting their lives and daily work, I hope the reader can better understand these individuals—understand their ordinariness and excellence; grasp their reasons for success; and most of all, sense that those who discover the beauty in mathematics live more fulfilling lives.

Jun Wu

Acknowledgments

I would like to thank my wife, Yan Zhang, for her longtime support and generous help in my career, and my father Jiaqing and my mother Xiuzhen Zhu, who brought me to the world of mathematics when I was very young.

I would like to thank many of advisors, colleagues, and friends, who gave me their generous and wise advice and suggestions during my career, especially to Prof. Zuoying Wang of Tsinghua University, Prof. Sanjeev Khudanpur, Prof. Fred Jelinek, and Prof. David Yarowsky of Johns Hopkins University, Dr. Eric Brill of eBay, Prof. Michael Collins of Columbia University, Dr. Amit Singhal, ex-senior VP of Google, Mr. Matt Cutts, Dr. Peter Norvig, Dr. Kai-Fu Lee, Dr. Franz Och, Ms. Dandan Wu, Ms. Jin Cui, and Dr. Pei Cao of Google. I am also grateful to Prof. Wing H. Wong of Stanford University and Mr. John Kimmel of Chapman & Hall, who helped me to publish this book.

Special thanks to Rachel Wu and Yuxi Wang for translating the book from Chinese, and Ms. Teena Lawrence and Ms. Michele Dimont, the project managers of the book.

Chapter 1

Words and languages, numbers and information

Words and numbers are close kin; both building blocks of information, they are as intricately linked as elements of nature. Language and mathematics are thereby connected by their shared purpose of recording and transmitting information. However, people only realized this commonality after Claude E. Shannon proposed the field of information theory, seven decades ago.

Since ancient times, the development of mathematics has been closely tied to the ever-growing human understanding of the world. Many fields—including astronomy, engineering, economics, physics, and even biology—depended on mathematics, and in turn, provided new grounds for mathematics to advance. In the past, however, it was quite unheard of for linguistics to draw from mathematics, or vice versa. Many famous mathematicians were also physicists or astronomers, but very few were also linguists. Until recently, the two fields appeared incompatible.

Most of this book tells the story of the past half century or so, but in this chapter, we will venture back to ancient history, when writing and numbers were first invented.

1.1 Information

Before our *Homo sapiens* ancestors developed technologies or started looking like modern humans, they could convey information to each other. Just as zoo animals make unintelligible animal noises, early humans made unintelligible "human" sounds. Initially, maybe the sounds had little meaning beyond exercising available muscles, but gradually, they began to carry messages. For example, some series of grunts may signify, "there's a bear!". To which a companion may grunt "yuh" in acknowledgment, or another series of sounds that signify, "let's go pelt it with rocks." See Figures 1.1 and 1.2.

FIGURE 1.1: Earliest forms of communication for humankind

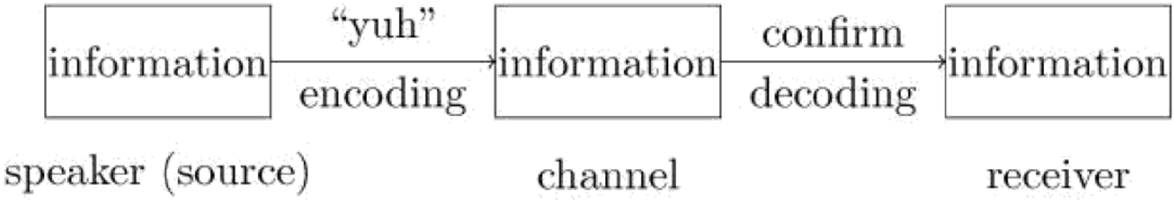


FIGURE 1.2: Same communication model beneath primordial grunts and modern information transfer.

In principle, there is little difference between these primordial grunts and the latest methods of information transmission, reception, and response. We will fill in the details of communication models in later chapters, but note here that simple models capture both ancient and modern communication.

Early humans did not understand enough about the world to communicate much, so they had no need for language or numbers. However, as humankind progressed and civilization developed, there was a growing need to express more and more information. A few simple noises could no longer fulfill humans' communication needs, and thus language was invented. Stories of daily life, which can be considered a specific type of data, were actually the most valuable artifacts from that time. Transmitted through oral tradition, these stories were passed down the generations, through each cycle of offspring. When humans began to accumulate material goods or food surpluses, the concepts of "more" and "less" emerged. In these days, counting had not yet been invented, since there was no need to count.

1.2 Words and numbers

Our ancestors quickly learned more about their world, and their language became richer, more abstract in the process. Elements that were often described, including objects, numbers, and actions, were abstracted into words of their own, precursors to the vast vocabularies of today. When language and vocabulary grew to a certain size, they could no longer fit inside a single human brain—just as no one today can remember all the wisdom of mankind. A need for efficient recording of information arose, and its solution was writing.

Archeologists today can verify when writing (including numbers) first appeared. Many readers of *On Top of Tides* have asked me why that book primarily discusses companies in the United States. The reason is simple: the past hundred years of technological revolutions have been set almost completely there. Likewise, to study the information revolutions of 5,000 to 10,000 years ago, we must return to the continent of Africa, where our human ancestors first walked out, the cradle of human civilization.

A few thousand years before the oldest (discovered) Chinese oracle bones were carved, the Nile River Valley was already nurturing an advanced civilization. Ancient Egyptians were not only excellent farmers and architects, but they were also the earliest inventors of ideographs.* These were the famed hieroglyphics. Ancient Egyptian left many hieroglyphical scrolls describing their lives and religions. One of the most famous hieroglyphical scrolls is "Book of the Dead (Papyrus of Ani.)", which resides permanently in the British Museum. The scroll consists of over 20 meters of painted papyrus, with over 60 paintings and pictographs. This cultural relic portrays an all-encompassing record of Egyptian civilization, 3,300-3,400 ago.

In early days of Egyptian civilization, the number of existing hieroglyphics corresponded directly to the amount of information to be documented. The earliest engravings of hieroglyphics, dating back to the 32nd century BC, utilized only around 500 characters. By the fifth century BC (the classical Greco-Roman era), this number had increased 5,000 characters, approximately the number of commonly used Chinese characters. However, as civilization continued to develop, the number of hieroglyphics did not increase with the production of information. There is a finite limit to the number of characters any one person can remember, so instead of inventing more characters, ancient civilizations began to generalize and categorize concepts. For example, the ancient Chinese ideograph for "day" represented both the sun itself, as well as the time between sunrise and sunset. In ancient Egyptian hieroglyphics, a single symbol could also convey multiple meanings.

This idea of clustering ideas into single characters is similar to today's concept of "clustering" in natural language processing or machine learning. Ancient Egyptians required thousands of years to consolidate multiple meanings into a single word. Today, computers may take several hours, or even minutes, to accomplish the same task.

When a single word can take on many meanings, ambiguities inevitably emerge. Given different environments, a word can dance from one meaning to another. Disambiguation, or determining the specific meaning, has not changed from traditional linguists to modern computers: we must examine the context. In most cases, the context will tell us the answer, but of course, there are always outliers. Consider your favorite religious text. Many scholars, students, or other theologians have expounded on these texts, but there exists no interpretation without controversy. Each scholar will use his or her own understanding to eliminate ambiguity, but none have flawlessly succeeded thus far, otherwise many a bloody war would have been averted. Thus is the inconclusive nature of human language. In our case, the situation is similar. Even the most successful probabilistic models will fail at some point.

After the advent of writing, lifetimes of experience could be handed down from generation to generation. As long as a civilization is not exterminated, and there exist those who understand its language, this information can persist forever. Such is the case of the Chinese or, with a stretch, the Egyptians. Certainly, it is more difficult to unlock ancient records without living knowledge of the language, but it is not impossible.

Isolated civilizations, whether due to geographic, cultural, or historical reasons, adopted different languages. As civilizations grew, however, and the earth refused to expand, they came into contact, peaceful or otherwise. These interactions spawned a need for communication and with that, translation.

Translation itself is only possible because despite differences among languages, the underlying information is of the same type. Furthermore, language is only a *carrier* of information, not the information itself. With these two observations in mind, you might wonder, if we abstract out "language" and replace it with another medium, such as numbers, can we still encode the same information? Why yes, this is the basis of modern communication. If we are lucky, different civilizations might even communicate the same information in the same language, in which case we have a key to unlock their unknown secrets.

Around the seventh century BC, the Hellenic sphere of influence extended to Egypt, whose culture was gradually impacted by the Greek. After the Greeks (including Macedonians) and Romans became the rulers of Egypt, the Egyptian language was eventually latinized. Hieroglyphics phased out of usage, into the backstage of history. Only temple priests now learned the pictographs, which served only for record keeping. In the fourth century AD, emperor Diocletian eradicated all non-Christianity religions in Egypt, where the knowledge of hieroglyphics ceased to be taught.

Not until 1400 years later, in 1798, did the meaning of hieroglyphics come into light once more. When Napoleon led his expedition to Egypt, he brought along hundreds of scholars. On a lucky day, lieutenant Pierre-Francois Bouchard discovered an ancient Egyptian relic in a placed called

Rosetta (Figure 1.3). Atop were inscribed the same message in three languages: ancient Egyptian hieroglyphics, Demotic script (ancient phonetic Egyptian), and ancient Greek. Bouchard immediately realized the importance of his discovery to cracking the hieroglyphic code, so he handed the Rosetta Stone to accompanying scientist Jean-Joseph Marcel. Marcel copied the writings and brought them back to France for further study. In 1801, France was defeated in Egypt, and the physical tablet transferred to British hands, but Marcel's prints were circulated through Europe. Twenty-one years later in 1822, French linguist Jean-Francois Champollion finally decoded the hieroglyphics on the Rosetta Stone. Here, we see that the carrier of those writings, stone or paper, was unimportant. Instead, the writings themselves, the information, were the key.

FIGURE 1.3: Rosetta Stone.

Rosetta Stone deciphered, the entire history of ancient Egypt, dating back to the 32nd century BC, was suddenly at the disposal of historians and linguists. Thanks to this trove of information, modern day historians know much more about the Egyptians of five thousand years ago than the Mayans of only one thousand years ago. The Egyptians recorded the most important aspects of their lives in writing, so the information lives on. As a natural language processing researcher, I extract two guiding principles from the Rosetta Stone story.

First, redundancy vastly improves the chances that information can be communicated or stored without corruption. The Rosetta Stone repeated the same content three times, so if at least one version remains readable, the stone is decipherable. Fortunately, 2,000 years ago, someone had the foresight to copy Ptolemy's imperial edict in three languages. This concept of redundancy extends to information encoding across noisy channels (for instance, wireless networks). Consider naively that we send the same message twice. Then, it is more likely our recipient will receive at least one of them.

Second, large amounts of bilingual or multilingual language data, known as a corpus, are essential to translation. These data are the bases of machine translation. In this aspect, we do not require more knowledge than Champollion possessed, for the Rosetta Stone. We simply own more computers and apply mathematical tools to speed up the process.

With Rosetta Stone's importance in mind, we should not be surprised that so many translation services and software are all named after Rosetta. These services include Google's own machine translation service, as well as the best-selling (or at least much advertised) software for foreign languages, Rosetta.

We have seen that the emergence of writing was induced by an ancient "information revolution," when people knew more than they could remember. Similarly, the concept of numbers arose when people owned too many possessions to keep track of otherwise. A famous American physicist, George Gamow tells of such a primitive tribe in his book, "One, Two, Three... Infinity." The story goes that two tribal leaders were competing in who could name the largest number. One chief thought for some time and named, "three." After considering for some time, the other chief admitted defeat. Today, a nerdy middle school student might have named a googol, and the second, a googol to the googol, but consider the times. In primitive tribes, all objects were extremely scarce. Beyond three, the tribal chiefs knew only of "many" or "uncountable." By this reasoning, early humans could not have developed a complete counting system.

When our ancestors developed a need for numbers beyond three, when "five" and "eight" became distinguishable, counting systems were invented. Numbers, naturally, are the bases of these counting systems. Like words, numbers were born first in concept, then in writing. With ten convenient fingers to count on, early humans set their numeric systems in base ten. Without a doubt, if we all had twelve fingers instead of ten, we would probably be counting in base twelve right now.

To remember numbers, early humans also carved out scratches on wood, bone, or other portable objects. In the 1970s, archeologists unearthed several baboon leg bones from the Lebombo Mountains, between Swaziland and South Africa. These 42,000-year-old bones featured such scratches, and scientists believe that they are the earliest evidence of counting.

Characters with numeric meaning appeared around the same time as hieroglyphics, thousands of years ago. Nearly all ancient civilizations recorded “one,” “two,” and “three” in some form of line—horizontal like the Chinese, vertical like the Romans, or wedge-shaped, like the Mesopotamians (Figure 1.4). Early numbers simply recorded information, without any abstract meaning.

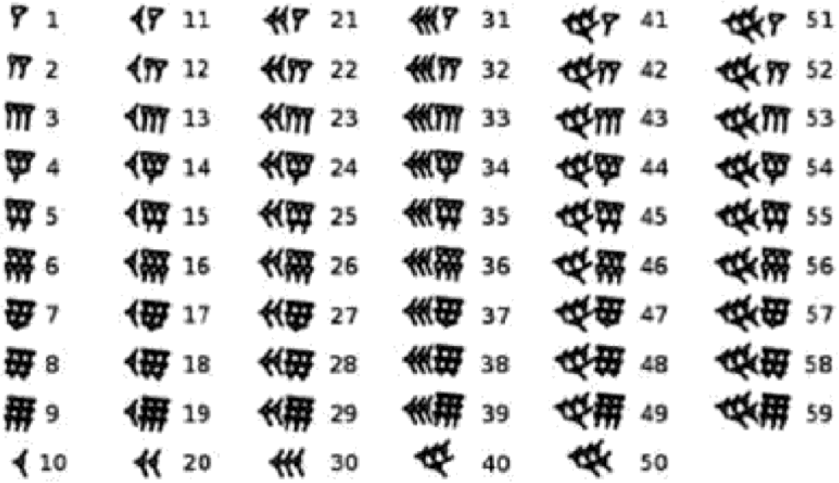


FIGURE 1.4: Cuneiform from ancient Mesopotamia.

Gradually, our ancestors accumulated so much wealth that ten fingers were no longer enough to count their possessions. The easiest method would have been to count on fingers and toes, but then what, grow another set of appendages?

Our ancestors invented far more sophisticated methods, though of course, there may have existed some extinct Eurasian tribe that *did* count on toes. They developed the “carry method” in base ten. This system was a great leap forward for mankind: for the first time, man had developed a form of numeric encoding, where different symbols represented different amounts.

Nearly all civilizations adopted the base ten system, but did there exist any civilizations who counted in base twenty—that is, took full advantage of all ten toes before switching to the carry system? The ancient Mayans did so. Their equivalent to our “century” was the sun cycle, 400 years long each. In 2012, the Mayans’ last sun cycle ended, and in 2013, the cycle began anew. This I learned from a Mayan culture professor, when I visited Mexico. Somewhere along the way, 2012’s “end of the sun cycle” became synonymous with “end of the world.” In perspective, imagine if the turn of each century meant apocalypse for us! Of course, we digress.

Compared to the decimal system, a base twenty system comes with nontrivial inconveniences. Even a child, without sophisticated language or vocabulary, can memorize a times tables, up to 9 times 9 equals 81. In base twenty, we would have to memorize a 19 by 19 table, equivalent to a Go board you could say, with 19 times 19 equals 361 entries. Even in burgeoning human civilizations, around 1 AD, no one but scholars would have the mind to study such numbers. The base twenty numeric system, coupled with a painfully difficult writing system, may have significantly slowed the development of Mayan society. Within a single tribe, very few were fully literate.

With respect to the different digits in base ten numbers, the Chinese and the Romans developed distinct units of orders of magnitude. In the Chinese language, there are words for ten, hundred, thousand, 10^4, 10^8, and 10^{12}. In contrast, the Romans denote 1 as I, 5 as V, 10 as X, 50 as L, 100 as C, 500 as D, and 1,000 as M, which is the maximum. These two representations unknowingly captured elements of information encoding. First, different characters represent different numerical concepts; and second, both imply an algorithm for decoding. In ancient China, the rule of decoding was multiplication. Two million is written as two hundred “ten-thousands,” or 2 × 100 × 10000. On the other hand, in ancient Rome the rules were addition and subtraction. Smaller numbers to the left meant subtraction, as IV signifies 5 - 1 = 4. The same numbers to the right meant addition, as VI signifies 5 + 1 = 6. Unfortunately, the Roman numeric system does not scale well to large numbers. If we wanted to express one million, we would need to write MMMM... and cover up an entire wall (Figure 1.5). Later the Romans invented M with a horizontal bar on top to represent “a thousand times a thousand,” but to express one billion, we would still need to cover an entire wall. Therefore, from an efficiency standpoint, the Chinese mathematicians were more clever.

FIGURE 1.5: A Roman mathematician tried to write "one million" on the board.

However, the most efficient numeric system came from ancient India, where today's universal "Arabic numerals" actually originated. This system included the concept of "zero," and it was more abstract (hence flexible) than that of both the Romans and the Chinese. As a result, "Arabic numerals" were popularized throughout Europe, which learned of them through Arabic scholars. This system's success lay not only in its simplicity, but also in its separation of numbers and words. While a convenience for traders, this detachment led to a divide between natural language and mathematics for thousands of years.

1.3 The mathematics behind language

While language and mathematics grew farther apart as disciplines, their internal similarities did not fade, unaffected by the growingly disparate crowds who studied them. Natural language inevitably follows the principles of information theory.

When mankind established its second civilization in the Fertile Crescent, a new type of cuneiform letter was born. Archeologists first uncovered these symbols on clay tablets, which looked esoterically similar to Egyptian tablets, so they mistook these symbols for pictograms. Soon, however, they realized that these wedge-shaped symbols were actually phonetic, where each symbol stood for a different letter. These constituted the world's earliest phonetic language.* The British Museum owns tens of thousands of these slate and clay tablets, carved with cuneiform letters. These engravings, along with Assyrian reliefs, are among the most valuable Babylonian relics.

This alphabetic language was developed by the Phoenicians and introduced to the east coast of Syria, in the west of Mesopotamia. Businessmen and traders, the Phoenicians preferred not to carve intricate wedge letters, so they designed an alphabet of 22 symbols. This alphabet spread with the Phoenicians' business interests, reaching the Aegean islands (including Crete) and the ancient Greeks. Upon reaching the Greeks, the alphabet was transformed into a fully developed alphabet, with no more ties to the Babylonian cuneiform script. Spelling and pronunciation were more closely linked, and the Greek alphabet was easier to learn. In the next few centuries, accompanying Macedonian and Roman conquests, these languages with at most a few dozen characters were embraced by much of Eurasia. Today, we refer to many Western phonetic languages as "Romance languages" for the Romans' role in linguistic dissemination.

Human language took a large leap from hieroglyphics to phonetic languages. An object's description transformed from its outward appearance to an abstraction of its concept, while humans subconsciously encoded these words as combinations of letters. Furthermore, our ancestors chose very reasonable encodings for their languages. For the Romans, common words were often short, and obscure words long. In writing, common words required fewer strokes than uncommon ones. Although our ancestors did not understand information theory, their ideas were fully compliant with the principle of making an encoding as short as possible. The resulting advantage is that writing saves time and material.

Before Cai Lun invented paper, writing was neither easy nor cheap. For example, in the Eastern Han dynasty (around first century AD), text was often inscribed on materials including turtle shell, stone, and bamboo. Since the process was so arduous, every single character was treated as if written with gold. Consequently, classical Chinese writing was painfully concise, while contemporary spoken language was much more verbose and colloquial, not much different from today's Chinese. In fact, the Lingnan Hakka peoples of southern China closely retain the ancient spoken language, with vocabulary and mannerisms of the late Qing dynasty.

This concept of language compression aligns with basic ideas in information theory. When a communication channel is wide, information can be sent directly; but if a communication channel is narrow, then information must be compressed as much as possible before delivery and recovered upon receiving. In ancient times, two people could speak quickly (wide channel), so no compression was needed. On the other hand, writing was slow and expensive (narrow channel), so scholars must first distill daily vernacular into exquisitely crafted poetry. Converting everyday speech into fine writing was a form of compression, and interpreting classic Chinese today is a form of decompression.

We also see this phenomenon in action when streaming video. Broadband provides high bandwidth, so we can watch high-definition videos with sharp resolution. Mobile data plans enforce heavy limits, and data is sent over unreliable networks, so latency is much higher, and resolution is a few magnitudes lower. Though a few thousand years ago there was no information theory, classic Chinese writing adhered to its principles.

Around the time of the late Babylonians, two historical works were produced: one Chinese, one Jewish. Chinese historian Sima Qian wrote a 530,000-word account of Chinese history in the classical style, and the ancient Jews began documenting their history in the Middle East, under Babylonian rule. This latter body of work consisted of Moses' teachings, and we refer to them collectively as the Torah. Its straightforward prose is similar to Sima Qian's writings, but unlike the Chinese manuscript, the Torah was taken into the Bible, whose writing spanned many centuries. Later scribes worked from manuscripts that were hundreds of years old themselves, so copying errors were unavoidable. Scholars say that today, only Oxford University owns an errorless copy of the ancient Bible.

Ancient Jewish scholars copied the Bible with utmost devotion and propriety, washing their hands to pray before writing the words "God" or "Lord." However, copy errors would undeniably emerge, so the scholars devised an error detection method, similar to that used in computers today. They assigned each Hebrew letter to a number, so that every row and column summed to a known value. After coping each page, a scholar would verify that the sums on the new page were the same as those on the old or conclude that he erred in the transcription. Each incorrect sum for a row (or column) signified at least one error on that row, so errors could be easily found and eliminated (see Figure 1.6). Like the ancient Hebrews, modern computers also use the idea of checksums to determine whether data is valid or corrupted.

FIGURE 1.6: Ancient Jewish scholars check every row and sum to verify that they copied the Bible correctly.

From ancient times to now, language has become more accurate and rich, largely due to advances in grammar. I am not a historian of languages, but I would guess that with high probability, grammar started taking shape in ancient Greek. If we consider morphology (constructing words from letters) as the encoding rules for words, then grammar captures the encoding and decoding for languages. However, while we can enumerate all words in a finite collection, the set of possible sentences is infinite. That is, a few tomes worth of dictionary can list all the words in the English language, but no one can compile all English writings ever to exist. Mathematically speaking, while the former can be completely described by a finite set of rules (trivially, we can enumerate all words), the latter cannot.

Every language has its niche usages that grammar rules do not cover, but these exceptions (or "inaccuracies") give language its color. Occasional dogmatic linguists treat these exceptions as "sick sentences." They spend their lives trying to eliminate the linguistic disease and purify the language through new grammar rules, but their work is futile. Take Shakespeare, for instance. Classics now and popular in his time, Shakespeare's works often contained famous, yet ungrammatical phrases. Many attempts were made to correct (or rather, tamper with) his writings, but while these attempts have been long forgotten, Shakespeare's "incorrect" writings persisted. Shakespearean brilliance, taught in schools around the world, originates from grammatical "mistakes."

Grammatical deviancy in literature leads to a controversy: do we consider our existing bodies of text (corpus) as the true expression of language, or should we designate a set of rules as correct usage? After three or four decades of debate, natural language processing scientists converged on the former, that existing data is truth. We will cover this period of history in Chapter 2.

1.4 Summary

In this chapter, we traced the history of words, numbers, and language to pique the reader's appetite for the mathematics intrinsic to our lives. Many of the topics introduced here are the focus of later chapters, including the following.

- Principle of communication and the model of information dissemination
- Encoding, and shortest codes
- Decoding rules and syntax of language
- Clustering
- Checksums (error detection and correction)
- Bilingual texts and corpuses, useful in machine translation
- Ambiguity, and the importance of context in eliminating ambiguity

Modern natural language processing researchers are guided by the same principles as our ancestors who designed language, though the latter's choices were mostly spontaneous and unintentional. Mathematics is the underlying thread through past and present.

*Images that represent objects or ideas.

*If we treat each stroke of a Chinese character as a "letter," we could consider Chinese as "alphabetic" as well, but only in two dimensions.

Chapter 2

Natural language processing—From rules to statistics

In the previous chapter, we introduced that language emerged as the medium of information exchange between humans. Any language is an encoding of information: it is composed of individual units—letters, words, or Arabic numerals—and an encoding algorithm—a collection of grammar rules. When we communicate an idea, our brains apply grammar encodings on these units to extract comprehensible sentences from abstract thought. If the audience understands the same language, they can use the same rules to decode a string of words back into abstract thought. Thus, human language can be described in terms of information theory and mathematics. While many animals have means of communication, only mankind uses language to encode information.

By 1946, modern computers were becoming increasingly accessible, and computers began to outperform humans at many tasks. However, computers could only understand a limited set of machine-oriented commands, not at all like the English we speak. In this context, a simple question arose: do computers have the potential to understand natural language? Scientists have contemplated this idea since the computer's earliest days, and there are two cognitive aspects to it. First, is a computer powerful enough to understand natural language? Second, does a computer process and learn natural language the same ways a human would? This chapter explores these two questions in depth, but on a high level, the answer to both is a resounding yes.

2.1 Machine intelligence

The earliest proponent of machine intelligence was the father of computer science, Alan Turing. In 1950, he published the seminal paper, "Computing machinery and intelligence," in the journal *Mind*. Rather than detailing new research methods or results, this paper was the first to provide a test that determined whether a machine could be considered "intelligent." Suppose a human and machine were to communicate, both acting as if they were human (Figure 2.1). Then, the machine is intelligent if the human cannot discern whether he is talking to a machine or another human. This procedure is known as the eponymous Turing Test. Though Turing left behind an open question, rather than an answer, we generally trace the history of natural language processing back to that era, 60 years ago.

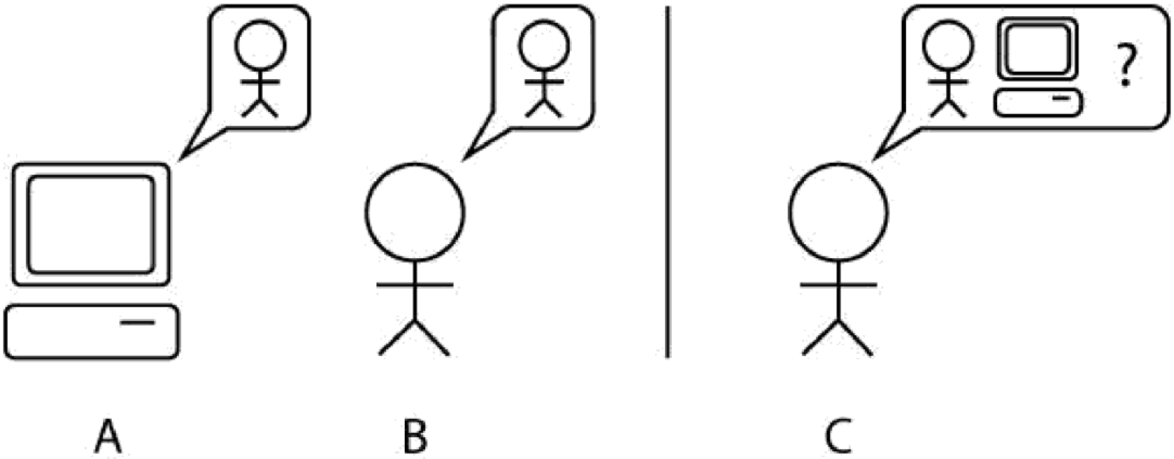


FIGURE 2.1: The human cannot tell whether he is talking to a human or machine, behind that wall.

These 60 years of history can be roughly divided into two phases. The first two decades, from the 1950s to the 1970s, were spent in vain but well-intentioned efforts. Scientists around the world wrongly assumed that machines learned the same way humans did, so they spent some nearly fruitless 20 years trying to replicate human thought processes in machines. Only in the 1970s did scientists reassess their approach towards natural language processing, which entered its second phase of research. In the past 40 years, their mathematical models and statistical methods have proven successful. Nowadays, natural language processing is integrated into many consumer products, such as voice assistants on smartphones or automated phone call agents. There are few remaining contributions by the early natural language processing scientists, but their work has been essential to understanding the success of modern methods and avoiding the same pitfalls. In this section, we recount the history of early machine intelligence efforts and in the next, we follow the transition to statistical methods.

Tracing the history of artificial intelligence, we come to the summer of 1956. Four young visionaries—28-year-olds John McCarthy and Marvin Minsky, 37-year-old Nathaniel Rochester, and 40-year-old Claude Shannon—proposed a summer workshop on artificial intelligence at Dartmouth College, where McCarthy was teaching at the time. Joining them were six additional scientists, among which were 40-year-old Herbert Simon and 28-year-old Allen Newell. At these seminars, they discussed unsolved problems in computer science, including artificial intelligence, natural language processing, and neural networks. It was here that the concept of artificial intelligence was first formulated. Other than Shannon, these ten scientists were of little fame or import at that time, but in years to come, four would win Turing Awards (McCarthy, Minsky, Simon, and Newell). Though he received no Turing Award himself, Shannon has an equivalent position to Turing in the history of computer science as the "father of information theory"; in fact, the highest award in information theory is named for Shannon.

These ten scientists were later hailed as the top computer scientists of the 20th century, having initiated a multitude of research areas that are still active today, many of which have directly improved our lives. Unfortunately, the bright minds gathered at Dartmouth that summer produced few results of merit during that month. In fact, their understanding of natural language processing would have been inferior to that of a PhD student today. This is because the scientific community had made a gross misunderstanding about the nature of natural language processing research.

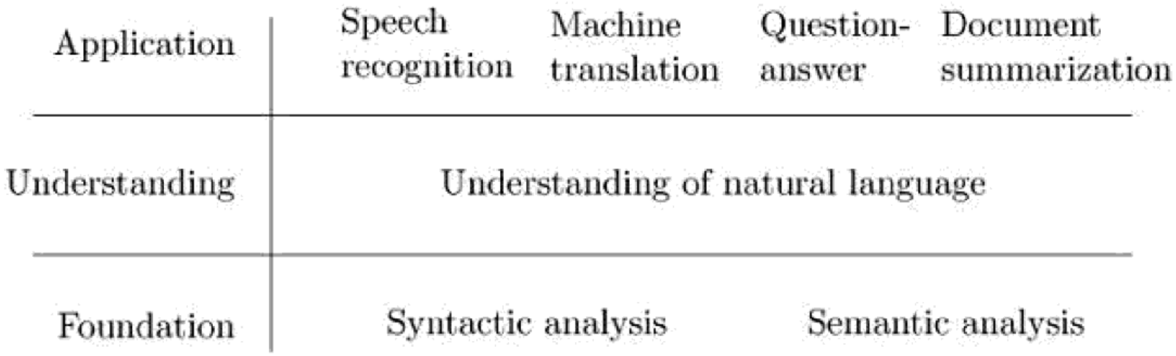
At that time, scientists presumed that in order to complete advanced tasks like translation, a computer must first understand natural language and in turn, possess human-level intelligence. Today, very few scientists insist on this point, but the general public still believes, to some extent, that computers require human-like intelligence to carry out intelligent tasks. In perspective, this assumption is not without reason. For example, we take for granted that a Chinese-English translator is fluent in both tongues. To humans, this is an intuitive deduction. However, this assumption falls into the "flying bird" fallacy. That is, by simply observing birds, we might design an airplane to flap wings in the proper manner, without needing to understand aerodynamics. In reality, the Wright brothers invented the airplane through aerodynamics, not bionics. Exactly replicating mechanisms in the natural world may not be the optimal way to construct competent machines. Understanding natural language processing from a computer's perspective is akin to designing an airplane from aerodynamic principles. For translation, a computer does not need to understand the languages it translates between; it simply translates.

Today, speech recognition and translation are widely adapted technologies, with billions of users, but few understand their underlying mechanisms. Many mistakenly assume that computers understand language, when in reality, these services are all rooted in mathematics, and more specifically, statistics.

In the 1960s, the primary problem scientists encountered was to teach computers to understand natural language. Prevailing beliefs decomposed natural language processing into two tasks: syntactic analysis and semantics extraction. That is, scientists believed that computers could understand natural language by uncovering the grammatical structure of text and looking up the meaning of words. Unfortunately, these goals were misguided, rooted in centuries-old presumptions. Linguistic and language studies have been well established in European universities since the Middle Ages, often forming the core of their curricula. By the 16th century, standardized grammar was becoming widespread, a byproduct of the Bible's introduction outside Europe. By the 18th and 19th centuries, Western linguists had formalized the study of various languages. The large corpus of papers produced by these scholars led to an all-encompassing linguistic system.

In the study of Western languages, we are guaranteed to encounter grammar rules, parts of speech, and word formation patterns (morphology). These rules provide a methodical way to learn foreign languages, and they are straightforward to describe to a computer. As a result, scientists had high hopes for applying traditional linguistics to syntactic analysis.

Compared to syntactic analysis, semantics were much harder to encapsulate and convey to a computer. Until the 1970s, most results about semantic analysis were mediocre at best.* Despite limited successes, semantics are indispensable to our understanding of language, so governments do fund both syntactic and semantic analysis research. The history of bringing natural language processing research to application is described in Figure 2.2.



Application	Speech recognition	Machine translation	Question-answer	Document summarization
Understanding	Understanding of natural language			
Foundation	Syntactic analysis		Semantic analysis	

FIGURE 2.2: Early attitudes towards natural language processing.

Let us illustrate an example of syntactic analysis. Consider the simple sentence, "Romeo loves Juliet." We can separate the sentence into three parts: the subject, predicate, and punctuation. Each part can be further incorporated into the following syntactic parse tree.

Computer scientists and linguists often denote these sentence analysis rules as "rewrite rules." The rules used above include:

- sentence → noun phrase + verb phrase + punctuation
- noun phrase → noun
- verb phrase → verb + noun phrase
- noun phrases → nouns

- noun → “Romeo”
- verb → “loves”
- noun → “Juliet”
- punctuation → “.”

Before the 1980s, grammar rules for natural language processing were manually created, which is quite different from modern statistical methods. In fact, up until 2000, many companies including the then well-known SysTran still relied primarily on large sets of grammar rules. Today, although Google pursues a statistical approach, there are still vestiges of grammar rules in some products like Google Now (the precursor to Google Assistant).

In the 1960s, compiler technologies were propelled forward by Chomsky’s formal language theory. High-level programming languages utilized context-free grammars, which could be compiled in polynomial time (see appendix, Polynomial problem). These high-level programming languages appeared conceptually similar to natural language, so scientists developed some simple natural language parsers in the same spirit. These parsers supported a vocabulary of a few hundred words and allowed simple clauses (sentences with a single digit number of words).

Of course, natural language is far more elaborate than simple sentences, but scientists assumed that computers’ explosively increasing computation power would gradually make up for these complexities. On the contrary, more computation power could not solve natural language. As we see in Figure 2.3, syntactic analysis of even a three-word sentence is quite complex. We require a two-dimensional tree structure and eight rules to cover three words. To a computer, these computations are negligibly fast—until we evaluate the growth in complexity with respect to the text. Consider the following sentence from a *Wall Street Journal* excerpt:

> “The FED Chairman Ben Bernanke told the media yesterday that $700B bailout funds would be lended to hundreds of banks, insurance companies, and automakers.”

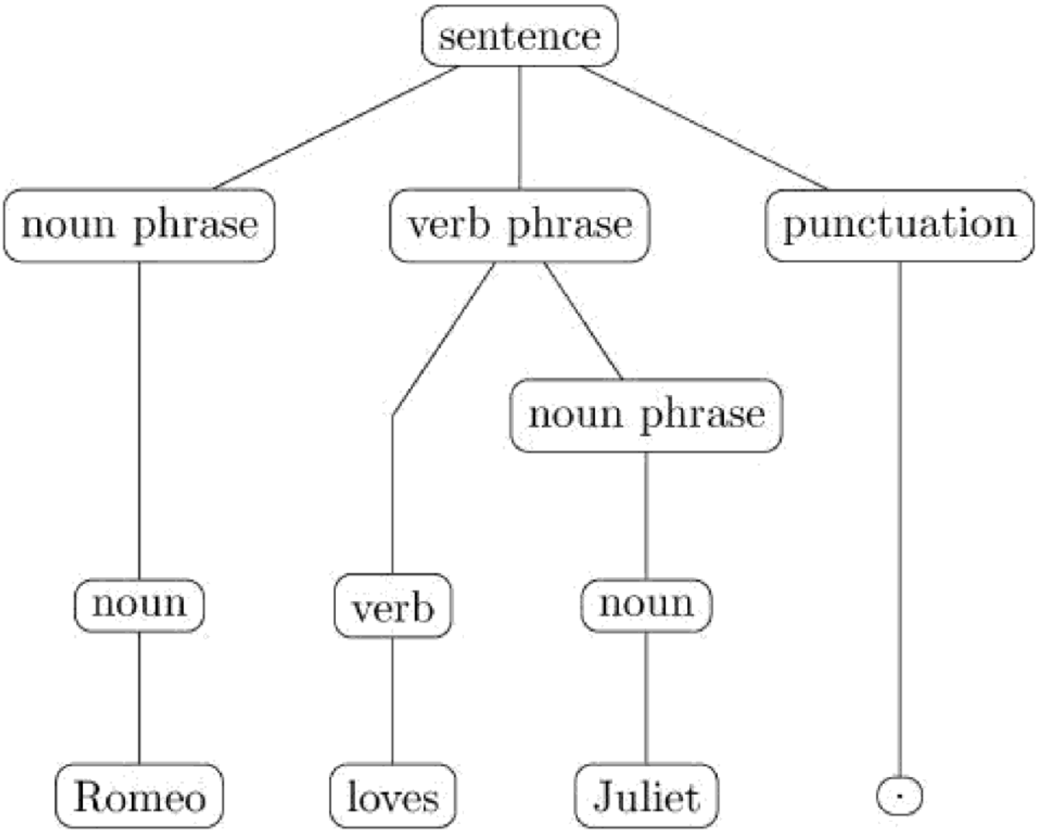


FIGURE 2.3: Syntactic parse tree for "Romeo loves Juliet."

This sentence still follows the "subject, predicate, punctuation" pattern,

> Noun Phrase [The FED Chairman Ben Bernanke] —— Verb Phrase [told the media... automakers] —— Punctuation [.]

The noun phrase at top level can be further divided into two noun phrases, "The FED Chairman" and "Ben Barnanke," where the former acts as a modifier to the latter. We can similarly decompose the predicate, as we can any linear statement. That said, the resultant two-dimensional parse tree would become exceedingly complicated, very fast. Furthermore, the eight rules provided for the "Romeo loves Juliet" example cannot adequately parse this new sentence. This complex example not only has more elements to parse, but also requires more rules to cover its structure.

In the general case, there are two barriers to analyzing all natural language using a deterministic set of rules. First, there is an exponential growth in the number of rules (not including part-of-speech tagging rules) required to cover additional grammatical structures. To cover a mere 20% of all statement requires tens of thousands of rules. Linguists simply cannot produce enough rules fast enough, and more specific rules are often contradictory, so context is required to resolve conflicts. If we want to cover over 50% of all statements, then every additional sentence requires many new grammatical rules.

This phenomenon is similar to an adult's reduced capability of picking up foreign languages. Children build language schemas as their brains develop, directly integrating language into the way they think. In contrast, adults learn foreign languages by studying vocabulary and grammar rules. An intelligent adult who learns English as a second language may find it difficult to speak as fluently as a native speaker or perform well on the GRE, even after 10 years of immersion. That is because rules cannot exhaustively describe a language's nuances.

Even if we obtained a full set of rules governing a language, we cannot efficiently implement them as algorithms. Thus, the second barrier to using deterministic rules for natural language processing is computational intractability. Recall that aforementioned compiler technologies of the 1960s parsed context-free grammars, which could be evaluated in polynomial time. These grammars are inherently different from natural language's context-dependent grammar. In natural language, meaning is often derived from surrounding words or phrases. Programming languages are artificially designed to be context-free, so they are much faster to evaluate. Since natural language grammars depend on context, they can become very slow to parse.

The computational complexity (see appendix) required to parse such grammars was formalized by Turing Award winner, Donald Knuth. Context-free grammars could be parsed in $O(n^2)$ time, where n is the length of the statement. On the other hand, context-dependent grammars require at least $O(n^6)$ time. In other words, if we had a sentence of 10 words, then a context-dependent grammar would be ten thousand times slower to parse than context-free grammar. As sentence lengths grow, the difference in running time explodes. Even today, a very fast computer (Intel i7 quad-core processor) takes a minute or two to analyze a sentence of 20-30 words using rule-based methods. Therefore in the 1970s, IBM, with its latest mainframe technologies, could not analyze useful statements using grammar rules.

2.2 From rules to statistics

Rule-based syntactic analysis (for both grammar and semantics) came to an end in the 1970s. Scientists began to realize that semantic analysis was even more difficult with a rule-based system: context, common sense, and "world knowledge" often contribute to meaning, but are difficult to teach a computer. In 1968, Minsky highlighted the limitations of the then artificial "intelligence" in semantic information processing with a simple example by Bar-Hillel. Consider the two sentences, "the pen is in the box" and "the box is in the pen." The first sentence is easy to understand, and a foreign student who has studied half a year of English can comprehend its meaning. The second sentence may cause confusion for such students—how can a large box fit inside a pen? For a native speaker, the second sentence makes perfect sense alone: the box fits inside a fenced area, a pig pen or similar. Whether "pen" refers to a writing implement or farm structure requires a degree of common sense to determine. Humans develop these sensibilities from real-world experience; unfortunately for scientists, computers do not live" in the real world. This is a simple example, but it clearly illustrates the challenges of analyzing semantics with computers.

Around this time, interest in artificial intelligence waned, partly due to the obstacles inflicted by misguided research. Minsky was no longer an unknown young fellow, but one of the world's leading artificial intelligence experts; his views then significantly impacted the US government's policies on science and technology. The National Science Foundation and other departments were disappointed by the lack of progress in natural language processing, and combined with Minsky's uncertainty, funding in this field greatly shrunk over time. It can be said that until the late 1970s, artificial intelligence research was more or less a failure.

In the 1970s, however, the emergence of statistical linguistics brought new life to natural language processing and led to the remarkable achievements we see today. The key figures in this shift were Frederick Jelinek and his group in IBM's T.J. Watson Research Center. Initially they did not aim to solve natural language, but rather the problem of speech recognition. Using statistical methods, IBM's speech recognition accuracy increased from roughly 70% to about 90%, while the number of words supported increased from hundreds to tens of thousands. These breakthroughs were the first steps towards practical applications of this research, outside the laboratory. We will pick up Jelinek's story later in this chapter.

IBM Watson Laboratories methods and achievements led to a momentous shift in natural language processing. A major figure in these developments was Dr. Alfred Spector, who took roles of VP of research at IBM and Google and held a professorship at Carnegie Mellon University. After Spector joined Google in 2008, I chatted with him about that shift in natural language processing those years ago. He said that while Carnegie Mellon University had delved very deep into traditional artificial intelligence, the researchers there encountered a few insurmountable obstacles. After visiting IBM Watson Laboratories later that year, he realized the power of statistical methods, and even as a professor of the traditional system, he could sense great change on the horizon. Among his students were Kai-Fu Lee, who was among the first to switch from traditional natural language processing methods to statistics. Along this line of research, Kai-Fu Lee's and Hsiao-Wuen Hon's excellent work also helped their thesis advisor, Raj Reddy, win the Turing Award.

As the head of research in two of the world's top technology companies, Spector's keen sense of the future of artificial intelligence was unsurprising. However, this was not a unanimous recognition; the schism between rule-based and statistical-based natural language processing lasted another 15 years, until the early 1990s. During this period, subscribers to each viewpoint organized their own conferences. In mutually attended conferences, each sector held their respective sub-venues. By the 1990s, the number of scientists adhering to the former method steadily decreased, and attendees of their conferences gradually switched over. As such, the prevailing viewpoint slowly converged to statistics.

In perspective, 15 years is a long time for a scientist; anyone who began their doctoral research following traditional methods and insisted on that path may have emerged to realize that their life's work had no more value. So why did this controversy last for 15 years? First, it takes time for a new branch of research to mature. Early on, the core of the statistical approaches lay in communication models and their underlying hidden Markov models (these mathematics are described in more detail later in the book). The input and output of this system were one-dimensional sequences of symbols, whose ordering is preserved. The earliest success of this system was speech recognition, followed by part-of-speech tagging.

Note that this model's output differs significantly from that of traditional methods. Syntactic analysis required a one-dimensional sentence as input but output a two-dimensional analysis tree. Traditional machine translation output a one-dimensional sequence (in another language), but did not preserve semantic order, so the output had little practical value.

In 1988, IBM's Peter Brown proposed a statistical-based machine translation approach, but IBM had neither enough data nor a strong enough model to produce useful results. IBM was unable to account for the variations in sentence structure across languages. For example, in English we say "a red apple," but in Spanish, it would become "una manzana roja," or "an apple red." With these limitations, few studied or made progress in statistical machine translation. Eventually, even Brown joined Renaissance Technologies to make a fortune in investment. Twenty years later, the paper of Brown and his colleagues became popular and highly cited.

On the technical side, syntactic analysis's difficulty lies in the fact that grammatical components are often interspersed within a sentence, rather than simply adjacent to one another. Only a statistical model based on directed graphs can model complex syntaxes, and it was difficult to envision how those would scale. For a long time, traditional artificial intelligence scientists harped on this point: they claimed that statistical methods could only tackle shallow natural language processing problems and would be useless against larger, deeper problems.

In the past three decades, from the late 1980s to the present, the ever-increasing amounts of computing power and data have made the daunting task of statistical natural language processing attainable. By the late 1990s, statistical methods produced syntactic analysis results more convincing than those of the linguists. In 2005, the last bastion of rule-based translation, SysTran, was surpassed by Google's statistics-based methods. Google had achieved a more comprehensive and accurate translation system, all through mathematical models. That is why we can say that mathematics will answer all of natural language processing problems.

Recall that there were two reasons it took traditional natural language processing 15 years to die out. The first was purely technical—models required maturity through time—but the second was practical—scientific progress awaited the retirement of the old linguists. This is a frequent occurrence in the history of science. Qian Zhongshu, in the novel *Besieged City*, remarked that even if scientists in their prime are not physically old, they may hold tightly to old ideas. In this case, we must patiently wait for them to retire and relinquish their seats in the halls of science. After all, not everyone is willing to change their points of view, right or wrong. So the faster these people retire, the faster science can advance. Therefore, I often remind myself to retire before I become too confused and stubborn.

Two groups contributed heavily to the transition between the old and new generations of natural language processing scientists. Other than Jelinek's own IBM-Johns Hopkins collaboration (which included myself), the University of Pennsylvania, led by Mitch Marcus, also played a big role. Marcus managed to obtain support from the National Science Foundation to set up the LCD project, which amassed the world's largest major-language corpus and trained a group of world-class researchers. Scientists from these two groups joined the world's leading research institutions, forming a de facto school of thought and shifting academia's predominant viewpoint.

At the same time, applications of natural language processing have also changed tremendously in the past three decades. For example, the demand for automatic question-answer services has been replaced with web search and data mining. As new applications relied more on data and shallow natural language processing work, the shift towards statistics-based systems was expedited.

Today, there are no remaining defenders of the traditional rule-based approach. At the same time, natural language processing has shifted from simple syntactic analysis and semantic understanding to practical applications, including machine translation, speech recognition, data mining, knowledge acquisition, and so on.

2.3 Summary

With respect to mathematics, natural language processing is equivalent to communication models. Communication models were the missing link between language as an encoding of information and natural language processing, but it took scientists many decades to arrive at this realization.

*It is worth mentioning that ancient Chinese linguists primarily focused on semantics, rather than grammar. Many ancient monographs, such as “Shuowen Jiezi” (explaining graphs and analyzing characters), were the results of such research.

Chapter 3

Statistical language model

Again and again, we have seen that natural language is a *contextual* encoding for expressing and transmitting information. For computers to understand natural language, mathematical models must first capture context. A model that accomplishes this—also the most commonly used model in natural language processing—is known as the statistical language model. This model is the basis of all natural language processing today, with applications including machine translation, speech recognition, handwriting recognition, autocorrect, and literature query.

3.1 Describing language through mathematics

The statistical language model was created to solve the problem of speech recognition. In speech recognition, a computer must decide whether a sequence of words forms a comprehensible sentence, and if so, return the result to the user.

Let us return to the example from the previous chapter:

> The Fed Chair Ben Bernanke told media yesterday that $700B bailout funds would be lended to hundreds of banks, insurance companies and auto-makers.

This sentence reads smoothly and its meaning is clear. Now suppose we changed the order of some words in the sentence, so that the sentence becomes:

> Ben Bernanke Federal Reserve Chairman of $700 billion told the media yesterday that would be lent to banks, insurance companies, and car companies hundreds of.

The sentence's meaning is no longer clear, but the reader can still infer its meaning, through the numbers and nouns. Now we scramble the words in the sentence to produce:

> the media Ben $700B of told Fed companies that lended yesterday insurance and banks, of auto-makers The and Chair, hundreds would be Bernanke the.

This final sentence is beyond comprehension.

If we ask a layperson to distinguish the differences between the three versions, he might say that the first follows proper grammar and is easy to understand; the second, while not grammatical, still retains meaning through words; the third obfuscates the words and any remaining meaning. Before the last century, scientists would have agreed. They would have tried to determine whether a text was grammatical, and if so, whether it conveyed any meaning. As we discussed in the previous chapter, this methodology led to a dead end. Near the turn of the century, Jelinek changed the prevailing perspective with an elegant statistical model.

Jelinek assumed that a sentence is meaningful if it is likely to appear, where this likeliness is measured with probabilities. We return to the three sentences again. The probability the first sentence appears is about 10^{-20}, while second and third range from 10^{-25} and 10^{-70}, respectively. While these probabilities are all extremely small, the first sentence is actually 100,000 more likely to appear than the second, and billions of billions more likely to appear than the third.

For the mathematically rigorous, let sentence S=w1,w2,...,wn represent an ordered sequence of individual words of length n. We would like to determine the probability $P(S)$ that S appears in any body of text we can find. Naively, we could compute this probability as follows. Enumerate all sentences ever uttered in the entirety of human history, and count the number of times S appears. Unfortunately, even a fool could see the impossibility of such an approach. Since we cannot determine the true probability that any sentence occurs, we require a mathematical model for approximating this value. S=w1,w2,...,wn, so we can expand $P(S)$ as

$$P(S)=P(w1,w2,\ldots,wn). \tag{3.1}$$

By conditional probability, it follows that the probability w_i occurs in the sequence is equal to the probability w_i appears alone, multiplied by the probability that w_i appears, given the existing sequence w1,w2,...,wi-1. We thus expand $P(S)$ to

$$P(S)=P(w1,w2,\ldots,wn)=P(w1)\cdot P(w2|w1)\cdot P(w3|w1,w2)\ldots P(wn|w1,w2,\ldots,wn-1), \tag{3.2}$$

where each word's appearance depends on the previous words.*

From a calculation standpoint, $P(w_1)$ is easy to find, and $P(w_2|w_1)$ is not hard either. However, P(w3|w1,w2) provides some difficulty because it involves three variables (words), w_1, w_2, and w_3. To calculate each variable's probability, we require a table the size of a language dictionary, and the size of these tables for conditional probabilities increases exponentially with the number of variables involved. When we reach the final word w_n, we realize that P(wn|w1,w2,...,wn-1) is computationally infeasible to calculate. So we return to our mathematical model—is there any way to approximate this probability without these expensive computations?

Russian mathematician Andrey Markov (1856-1922) proposed a lazy but effective method for resolving this quandary. Whenever we want to determine the probability of word w_i in sentence S, we no longer consider all previous words w1,w2,...,wi-1. Rather, we reduce our model to only consider the previous word $w_{i\text{-}1}$. Today we call these models Markov chains.* Now, the probability that sentence S appears becomes

$$P(S)=P(w1)\cdot P(w2|w1)\cdot P(w3|w2)\ldots P(wn|wn-1). \tag{3.3}$$

Equation 3.3 corresponds to the statistical language model known as the bigram model. Of course, we may also consider the n-1 words preceding word w_n, and such grams are known as N-gram models, introduced in the next section.

After applying the Markov assumption to $P(S)$, we encounter the next problem of determining conditional probabilities. By definition,

$$P(wi|wi-1)=P(wi-1,wi)P(wi-1). \tag{3.4}$$

Nowadays, researchers can easily calculate the joint probability $P(w_{i\text{-}1}, w_i)$ and the marginal probability $P(w_i)$. Extensive amounts of digital text (corpuses) allow us to count occurrences of word pairs $(w_{i\text{-}1}, w_i)$ and individual words w_i. Suppose we have a corpus of size N words. Let #(wi-1,wi) be the number of times $w_{i\text{-}1}$, w_i appear consecutively, and #(wi-1) be the number of times $w_{i\text{-}1}$ appears alone. We now obtain relative frequencies of occurrences in our corpus.

$$f(wi-1,wi)=\#(wi-1,wi)N \tag{3.5}$$

$$f(wi-1)=\#(wi-1)N \tag{3.6}$$

By the Law of Large Numbers, as long as our sample size *N* is large enough, then relative frequencies of events approach their true probabilities.

$$P(wi-1,wi)\approx\#(wi-1,wi)N \tag{3.7}$$

$$P(wi-1)\approx\#(wi-1)N \tag{3.8}$$

Returning to conditional probability, we notice that the two above values share the denominator *N*, so P(wi|wi-1) simplifies to a ratio of counts.

$$P(wi|wi-1)=\#(wi-1,wi)\#(wi-1) \tag{3.9}$$

Now, some readers may begin to appreciate the beauty of mathematics, which converts complicated problems into elementary forms. This process may seem slightly unbelievable at first. Basic probabilities can accomplish feats including speech recognition and machine translation, in which complicated grammars and artificial intelligence rules stumble. If you are questioning the efficacy of the model we just presented, you are not alone. Many linguists also doubted this statistics-based approach. In the modern day, however, researchers have found that statistics trumps the most sophisticated rule-based systems. We provide three real-world applications of this model, for the disbelievers.

Thirty years ago, the ex-president of Google China, Kai-Fu Lee, meteorically rose to the preeminence of speech recognition. As a doctoral student, he built a continuous-speech, speaker-independent system using the same Markov models described above.

Now let us fast forward some years to the 21st century, when Google developed its machine translation system, Rosetta. Compared to many universities and research institutions, Google was late to the machine translation game. Prior to Rosetta, IBM, the University of Southern California, Johns Hopkins, and SysTran had already made much progress in the field. For many years, these veteran institutions had participated in the National Institute of Standards and Technology (NIST) machine translation evaluation, a widely recognized performance benchmark. Google's Rosetta system first entered the competition in 2005, after only two years of development. To everyone's astonishment, Rosetta came in first, by a wide margin. It surpassed all rule-based systems, upon which over a decade of research was focused. What's Google's secret weapon? It is datasets and mathematical models hundreds of times larger than those of its competitors.*

A few years after that, around 2012, we come to some of my own work at Google involving the automatic question-answer system. At the time, many scientists in laboratory settings could design computers that answered basic questions. That is, "what is the population of the United States," or "what year was Donald Trump born?" While factual questions had straightforward answers like 300 million or 1946, "why" and "how" questions required further explanation. Questions like "why is the sky blue" were beyond the scope of computers then. We could mine large datasets to compile an answer with all the right keywords, but in order to sound fluent, a computer required the statistical language model.

Since incorporating that model, this system has been released in English and other foreign languages. If you Google “why is the sky blue” today, you will find a well-prepared answer. In future chapters, we will delve into the details of this system’s mathematics, but now we return to the statistical language model.

From these three examples, we see that the statistical language model has become an indispensable part of any “intelligent” computer system. Nonetheless, there are still countless implementation details between the mathematics and the software. For example, if a word (or pair of words) we encounter does not appear in the corpus, or only appears rarely, then our estimated probabilities are skewed. Fortunately, Jelinek and his colleagues not only presented this model, but also filled in many of its corner cases. These cases will be outlined in the following section. If you do not work with the statistical language model or find this mathematics unpalatable, then worry not. We have already laid all the foundations for the statistical language model, and you do not need to read further to appreciate its brilliance. The beauty of mathematics lies in its potential to accomplish groundbreaking work with a simple model.

3.2 Extended reading: Implementation caveats

Suggested background knowledge: probability theory and statistics.

Most of this book’s chapters will include an extended reading section, tailored towards professionals and those who wish to further study behind the mathematics behind the presented topics. These sections may require additional background to fully comprehend, so to save the reader’s time, I have noted the suggested prerequisite knowledge at the start of each section. While not a hard requirement, familiarity with these subjects will improve the reading experience. Later chapters do not depend on these extended readings, so readers are free to choose whether to read or skip any such section.

3.2.1 Higher order language models

In Equation 3.3 of the previous section, we assumed that the probability of each word w_i is related to its immediately preceding word w_{i-1}, but unrelated to all other preceding words w_j, where $j < i - 1$. The reader might wonder whether such an assumption is slightly oversimplified. Indeed, it is easy to find such examples where word w_i depends on words other than w_{i-1}. Consider the phrase, “sweet blue lilies.” Here, “sweet” and “blue” both describe “lilies,” but only “blue” is considered the previous word. As a result, we should perhaps consider the previous two words. Generalizing even further, we may modify our assumption so that a word w_i depends on several preceding words.

We express this assumption in mathematical notation. Suppose word w_i depends on N-1 preceding words. We modify the conditional probability of w_i, given an existing sequence of words, to

$$P(wi|w1,w2,\ldots,wi-1)=P(wi|wi-N+1,wi-N+2,\ldots,wi-1). \tag{3.10}$$

Equation 3.10 corresponds to a higher order Markov chain, or in natural language processing terms, an n-gram model. Some special cases of the n-gram model include $n = 2$, the bigram model (Equation 3.3), and $n = 1$, the context-free unigram model. The unigram model assumes that each word’s appearance is unrelated to the appearance of nearby words. In practice, the most commonly used model is $N = 3$, the trigram model, and higher orders beyond $N = 3$ are rarely used.

After learning that context is key to language, we might ask, why limit the orders to such low orders? There are two reasons. First, N-gram models quickly become computationally intractable. Both the spatial and time complexity of a meta model undergo exponential growth as the number of dimensions increases. A language's words are usually contained in a dictionary V where $|V| = 10,000$ to $100,000$ words. Each additional dimension adds exponentially many word combinations, where order N would require $O(|V|^N)$ space and O(|V|N-1) time. The model significantly improves from $N = 1$ to $N = 2$ and slightly improves at $N = 3$, but the model experiences little additional benefits when increasing to $N = 4$. While the marginal benefit of increasing N approaches none, the resources consumed increases exponentially, so $N = 4$ is near the maximum order that anyone will use. Google's Rosetta translation and voice search systems use the 4-gram model, but the model requires over 500 servers to store.

Second, we cannot cover all language phenomena, even if we try to increase N to infinity. In natural language, the relevance of context may span paragraphs, or even sections. Literary analysis is a prime suspect of far-ranging context, given that symbols may appear throughout an entire novel. Even if we *could* increase the order of our model to $n = 100$, we could not encapsulate all context. Such is a limitation of the Markov assumption. In its place, we have tools that consider long distance dependencies, which we will discuss later in this book.

3.2.2 Training methods, zero-probability problems, and smoothing

As described in Equation 3.3, the statistical language model requires the knowledge of all conditional probabilities, which we denote as the model's parameters. Obtaining these parameters from a corpus's statistics is known as training. For example, the bigram model requires two numbers, $\#(w_{i-1}, w_i)$ and $\#(w_{i-1})$, to estimate the conditional probability P(wi|wi-1). Seemingly straightforward, the model fails if the pair (w_{i-1}, w_i) never appears. Does P(wi|wi-1) equal 0 for all new sentences? Likewise, if the two counts are the same, does P(wi|wi-1) really equal 1? These questions involve the reliability of our statistics.

So far, we have assumed that statistics observed on our sample (available corpus) are equivalent to those that could be observed on the population (entire language). As long as our datasets are large enough, the Law of Large Numbers guarantees that this assumption holds. For example, suppose we want to determine the demographics of customers at the local mall. On a Saturday afternoon, we count 550 women and 520 men, so we conclude that $550/(550 + 520) = 51.4\%$ are women, and 48.6% are men. However, suppose we are in a hurry, so we walk in and out of the mall on a Tuesday morning. There are only 5 people, 4 men and 1 woman. Is it reasonable to conclude that 20% of customers are women, and 80% are men? What if we only observe 3 people, all women? Do we dare conclude that not a single man shops at the mall? Of course not. Small samples produce statistics with high variance.

Many people would say the unreliability of small samples is common sense, but the same people often forget this fact when training language models. Instead of addressing the lack of data, they question the model's validity. Today, the statistical language model has withstood the test of time, and many digital communications applications are built atop similar models. Given that the theory is sound, the remaining challenge lies in proper training of the model.

The first solution that comes to mind is to directly increase our dataset size. With some quick estimations though, we will see that all the data in the world is not enough to compose an adequate sample. Take Chinese, for instance. A Chinese dictionary contains around 200,000 words, not all commonly used.* To train a trigram model, we require 200,0003=8·1015 unique parameters. If we obtain our corpus by crawling the Chinese web, there are around 10 billion web pages, each with an average of 1,000 words (overestimate).