TEXT MINING: ALGORITHMS AND APPLICATIONS

Kenneth Cosh

Faculty of Engineering, CMU

Preface

The quantity of text being produced daily in our modern world is unparalleled throughout history. Not only is the global literacy rate higher than ever, but modern communication technologies afford the ability for almost anyone to create text and then share it across the world. The internet has become our primary source of information and whilst there is now plenty of rich multimedia, much of the core of the web is text, and virtually every interaction we have with social media creates more text in the form of user generated content.

Given that the quantity of text grows on a scale that would be impossible for anyone to read or capture knowledge from, it is important to turn to technology to help with unlocking the wealth of information, ideas and knowledge contained within. Text Mining offers a valuable means to extract assets from text, and this book intends to provide ideas and examples of how text mining can provide an asset, whether for business intelligence, or for personal interest.

The book is essentially broken into two main parts – firstly Natural Language Processing is introduced, along with discussing some of the key algorithms that can be used to mine information from text. The second part of the book examines a collection of applications of those algorithms, which could be used in a variety of different fields, whether text mining academic research, providing data for an e-commerce platform, or creating a virtual travel agent.

Dr. Kenneth Cosh

Table of Contents

Chapter 1 – Natural Language Processing **1**
1.1 Introduction 1
1.2 Basic NLP Tasks 8
1.2.1 Tokenization 8
1.2.2 Part-Of-Speech Tagging 11
1.2.3 Chunking / Parse Trees 13
1.2.4 Stemming and Lemmatization 14
1.2.5 Stop Lists 15
1.2.6 N-Grams 17
1.2.7 Terminology Extraction 19
1.2.8 Named Entity Recognition 20
References 21

Chapter 2 – Information Retrieval and Searching **23**
2.1 Introduction 23
2.2 Structured, Unstructured and Semi-structured Data 24
2.3 Precision and Recall 27
2.4 Search Engine Algorithms 30
References 35

Chapter 3 – Document Classification and Clustering **36**
3.1 Introduction 36
3.2 Content Based vs Request Based Approaches 36
3.3 Automatic Document Classification 37
3.4 Feature Extraction 38
3.4.1 Term Frequency – Inverse Document Frequency (tf-idf) 39
3.4.2 Log Likelihood 43
3.5 Comparing Documents 46
3.6 Clustering Documents 47
3.6.1 K-Means Clustering 48
3.6.2 Agglomerative Hierarchical Clustering 50
References 53

Chapter 4 – Content Clouds: A Simple Visualisation Tool **54**
4.1 Web 2.0 54
4.2 Content Clouds 58
4.3 Weblogs 61
References 64

Chapter 5 – Destination Discovery **66**
5.1 Wikipedia 66
5.2 Destination Content Clouds 68
5.3 Comparing Destinations 70
5.4 Clustering Destination 73
5.5 User Driven Comparisons 76
References 79

Chapter 6 – Recommender Systems **80**
6.1 Collaborative Filtering 80
6.2 Content-Based Recommender Systems 83
6.3 Hybrid and Mobile Recommender Systems 87
6.4 Sentiment Analysis 87
References 93

Chapter 7 – Towards Automatic Ontology Creation **94**
7.1 Concept Categorisation 95
7.2 Identifying Relationships between Categories 102
7.3 Identifying Relationships between Pages 103
References 107

Chapter 8 – Bibliometric Analysis **108**
8.1 Software Engineering 108
8.2 Current Research Themes 109
8.3 Evolving Research Themes 115
8.4 Conclusions 124
References 125

The quantity of text being produced daily in our modern world is unparalleled throughout history. Modern communication technologies afford anyone to create text and share it across the world. The internet has become our primary source of information and much of the core of the web is text. Virtually every interaction we have with social media creates more text in the form of user generated content.

Given that the quantity of text grows on a scale that would be impossible for anyone to read or capture knowledge from, it is important to turn to technology to help with unlocking the wealth of information, ideas and knowledge contained within. Text Mining offers a valuable means to extract assets from text, and this book intends to provide ideas and examples of how text mining can provide an asset, whether for business intelligence, or for personal interest.

CHIANG MAI
UNIVERSITY PRESS

ISBN: 978-616-398-452-4
9 786163 984524
ราคา 400 บาท

Chapter 1
Natural Language Processing

1.1 Introduction

Natural Language Processing (NLP) is an ongoing research discipline concerned with situations where machines interact with human languages and are required to handle the complexities of natural languages such as English. It encompasses a broad spectrum of challenging problems, including information retrieval, speech recognition, automatic translation, sentiment analysis and document classification. It is a discipline which combines knowledge from Linguistics and Computer Science, notably Artificial Intelligence (AI) and Machine Learning. The key challenges are Natural Language Generation (NLG) and Natural Language Understanding (NLU). This book is focused on the techniques and applications of extracting understanding from text, otherwise known as text mining.

Natural Language Processing faces a variety of non-trivial challenges as languages are full of complexities that have evolved in disparate ways over thousands of years with different natural languages presenting distinct complications. A natural language contrasts with an artificial language or formal language, such as a programming language, where precise rules are in place to govern the language. While a natural language may be supervised by rules such as syntax and grammar, its use is flexible and adaptable, and often encouraged to be so in order to afford elaborate communications. Its use varies due to ambiguity, nuance, dialect, metaphor, semantics, synonyms, pragmatics, amongst others and its continuing evolution is unplanned and uncontrolled.

Natural Language is obviously intended as a communication system, but even as intelligent humans we regularly suffer from misunderstandings and communication breakdowns. The causes are diverse, from simply missing vocabulary to misinterpreting the intended sense of an expression. Real time verbal communication is supported by

myriad of back channels, such as gesture, body language, tone of voice, etc. facilitating the expression of the affective state and illocutionary force of the message. When dealing with written text, some of these back channels are lost, which adds to the chance of miscommunication. With human conversation, breakdowns are often easily fixed, as the expression can be repeated or clarified (Dix, et al. 2003).

The field of Semiotics examines how signs and symbols (including linguistic signs) are used for communication, created by a sender, and then in turn interpreted by the recipient (Chandler, 2002). Ferdinand de Saussure in the early 20th century proposed a dyadic model for understanding sign systems, indicating a separation between a (linguistic) sign, and it's meaning (Komatsu & Harris, 1993). His model consists of a signifier and a signified, where the signifier is the form that a sign may take, while the signified is the concept being represented, as seen in Figure 1.1. This example, this demonstrates a separation between the textual sign "CAR" (signifier), and the object that it is referring to.

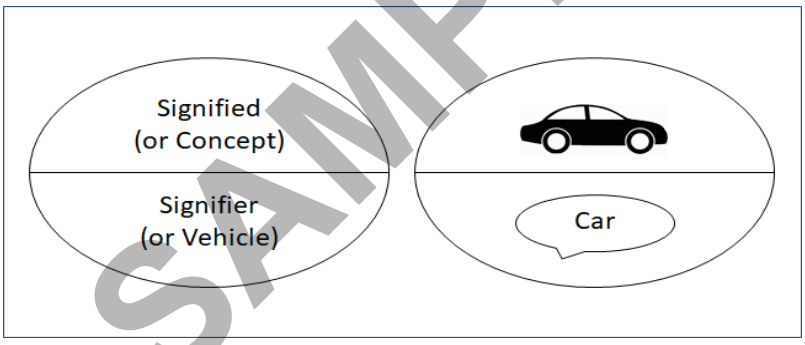

Figure 1.1: Saussure's Dyadic Model of a Sign

Meanwhile Charles Sanders Peirce, independently, but at a similar time, was investigating the field of Semiology, and suggested a triadic model, separating the sign, from what it represents, and thirdly how it is interpreted (Peirce, 1931-1958). Here the model has three parts; the Representamen, or form that the sign takes, the Interpretant, or sense made from the sign, and the Object, that was being referred to. Figure 1.2 shows a version of the Semiotic Triangle, which adds the sense taken, or the understanding taken from the natural language.

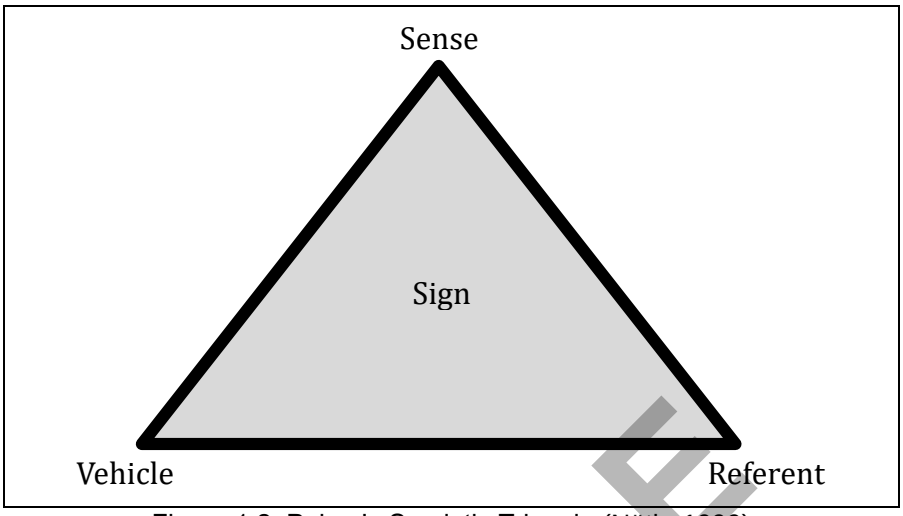

Figure 1.2: Peirce's Semiotic Triangle (Nöth, 1990)

Whilst the signs examined in semiotics are more than just linguistic or textual identifiers, it is clear that the intricacies of natural language have presented a complex challenge for a long time. Given the difficulty intelligent humans can have with processing linguistic signs, automating parts of this task is not trivial.

Stamper proposed a Semiotic Ladder to identify different levels in which communication is passed, or carried, by signs (Stamper, 1973). Initially there are two distinct levels to his ladder; the human information functions and the IT platform, as shown in Figure 1.3 (Cordeiro & Filipe, 2004).

Human Information Functions	**SOCIAL WORLD:** beliefs, expectations, commitments, contracts, law, culture…
	PRAGMATICS: intentions, communications, conversations, negotiations…
	SEMANTICS: meanings, propositions, validity, truth, signification, denotations…
The IT Platform	**SYNTACTICS:** formal structure, language, logic, data, records, deduction, software, files…
	EMPIRICS: pattern, variety, noise, entropy, channel capacity, redundancy, efficiency, codes…
	PHYSICAL WORLD: signals, traces, physical distinctions, hardware, component density, speed, economics…

Figure 1.3: Stamper's Semiotic Ladder

The implication here is that certain levels of communication can be handled by the technology platform, while other levels require human functions. The base level of the ladder is concerned with the Physical World, which could rely on sound waves in the case of verbal communication, or with digital communication this would involve the management of the cables required to make physical connections between the sender and receiver of a message. Clearly information technology has this level under control, whether considering the most primitive technology of fire beacons or modern communication systems. Communication will be effective if the physical world layer works as expected, and it only fails in situations such as impaired hearing or when cables are broken.

The next level of the ladder is the Empirics level, which governs the encoding of messages and their transmission across a communication channel. In the form of verbal communication, this would be concerned with sound waves, while as data is transmitted digitally empirics is concerned with the binary encoding of messages. Once again, largely this doesn't need to be considered by most human communicators who will assume that the empirics are functioning as expected.

The Syntactics layer is largely concerned with grammar and the formal structure of language. Once again, this is placed within the IT Platform layer, indicating that the formal structure of language can be managed by the machine, but higher levels of the ladder may not be. At this stage it is worth considering the utterance "Hungry I Am". For this message to be correctly understood the lower levels of the ladder need to function – the Physical World layer needs to correctly convey the message to the receiver, using the Empirics layer for encoding. In this case the grammar of the message would be questioned for its accuracy, and while the machine might reject the grammar, a human would probably understand the intention of the utterance. Whilst grammar is important, successful communication is more important.

The next layer is concerned with Semantics, or the meanings of the words used in an utterance. This layer is the first to be placed in the Human Information Functions layer, and this book will consider in several places how the IT platform may be able extend into the Semantics layer. The Semantic Web is a good example of where the IT platform is extending into understanding the Semantic layer. Semantics is concerned

with the meanings of the individual words within an utterance. In order to understand “Hungry I Am”, one needs to understand the desire for food, the concept of the self, and possession. If one does not understand the word ‘hungry’, then the statement makes no sense.

Pragmatics deals with the intention of the complete communication. The statement “Hungry I Am”, (ignoring the grammatical errors), could be interpreted as conveying the fact that the speaker needs to eat some food. At a pragmatic level, the statement is more likely to convey an invitation to go to have lunch together. Whilst semantics deals with the meanings of individual words in a statement, pragmatics deals with the sense of a combined utterance. The pragmatic layer is currently beyond the capabilities of modern information technology.

The upper layer of Stamper’s ladder is the Social World, reflecting the bottom layer of the Physical World. To illustrate the Social World layer, I recall when I shared an office with a few colleagues. As a new team, initially (for a few days), one of us would go to the colleague’s cubicle around 11:30 (due partly to working schedules) and ask something like “Hungry?”. After a few days of this pattern there was no need for any utterance – around 11:30 one of the office would walk to the middle of the room, and the rest of the colleagues would stand up and set off to the canteen. This behaviour occurred consistently without any need for explicit communication due to the development of a cultural level of communication.

For communication to be effective, all levels of the Semiotic Ladder need to be considered, without a successful physical world connection a message will be lost, while at the other end of the ladder, without the social world understanding, a new member of the team could be bemused as to why the rest of the team stood up and left the room at the same time. The base levels are fairly trivial for technology to handle, but the interesting modern challenges involve helping the machine with the upper layers.

Natural language is clearly used as a two-way communication system, and as in all conversations there are two key components; the ability to produce a meaningful contribution to the conversation, and the ability to make sense of the contributions of others. Regardless of the form of the communication, (written, spoken, formal, informal), these two components are pivotal to research in Natural Language Processing; in

other words, Natural Language Generation and Natural Language Understanding.

Natural Language Generation (NLG) concerns tasks where the machine needs to produce human readable language, such as producing summaries of long documents, or product descriptions, or responses for an online chat bot. A basic NLG task involves analysing a data set, deciding which features to include in a summary, and constructing text that appears natural, for example taking meteorological data and producing a weather report from it where there has been some success since the early 1990s (Goldberg et al., 1994). The techniques for NLG go beyond simple reflex-based rule matching.

Natural Language Understanding (NLU) is concerned with the other side of a conversation - enabling the machine to understand human language. The scope of NLU can vary greatly, from allowing a robotic response to a short command, to the challenge of fully understanding a complex novel. Depending on the scope of the problem, NLU tasks can range from trivial to very hard (AI Hard). Applications of NLU are diverse including text classification and organisation, translation, question answering and content analysis. Generally, NLP algorithms are applied to preprocess text, with NLU intended to make use of the results.

Along with NLG and NLU, Speech Recognition is often listed as a third core challenge within the field of Natural Language Processing. Here the goal is to recognise spoken language and convert it into text. Speech Recognition can scale between simply identifying specific, isolated instructions, such as “Call Home”, to enabling a speaker to input large amounts of text without typing. The field has demonstrated several significant breakthroughs, beginning in the 1990s and 2000s where the process would begin with a training set, where the user would need to read a selection of prepared statements so that the machine could learn their voice. More recently the ability of the machine to correctly identify speech without needing training, allowing speaker independence. A further application of speech recognition is referred to as Voice Recognition or Speaker Recognition, where the machine is able to determine who is speaking, which could be used for verification within a security process. Ironically, the abilities of some speech recognition applications (such as Alexa) have raised security fears themselves. Speech Recognition is an area of Natural Language Processing that has produced significant

progress, most notably with the application of Hidden Markov Models, Deep Learning and Artificial Neural Networks.

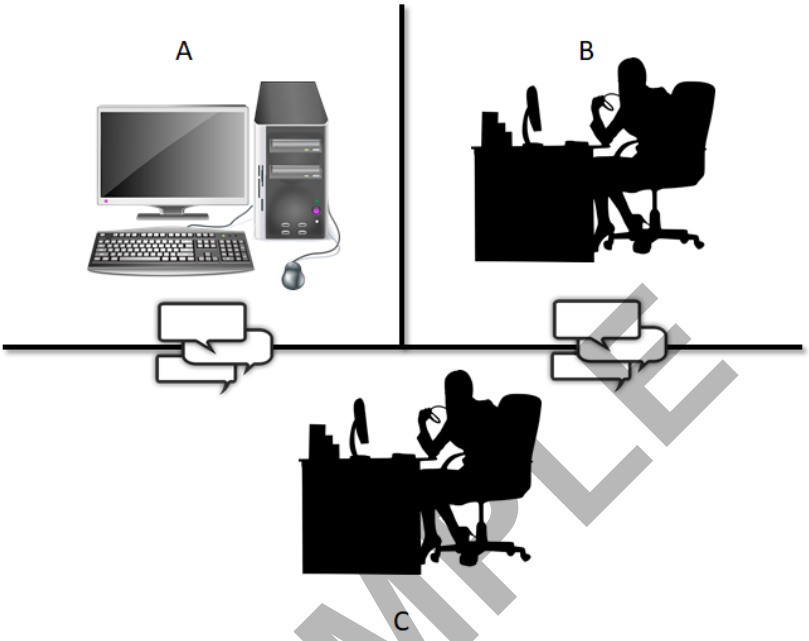

Figure 1.4: Model of the Turing Test

Early endeavours into the field of Natural Language Processing were inspired by the Turing test as proposed by Alan Turing in 1950 (Turing, 1950). The task became the quintessential test for a machine to demonstrate intelligent behaviour, in other words artificial intelligence. The test is to create a machine with which one can communicate, such that an intelligent human evaluator is unable to distinguish between the human and the machine. When asked questions in a conversation, the machine should be able to give responses that are human like in a natural language. The standard interpretation of the game is that the human player C has a conversation with a computer player A and another human player B and is left with the challenge of determining which is which as shown in Figure 1.4.

There appeared to be some early successes in Natural Language Processing, such as with some limited, basic automatic translation, however the challenges involved quickly emerged to be much more complex than initially anticipated and real progress was slow. Initially the

field turned to linguists to produce comprehensive sets of rules, such as grammatical rules, and if-then rules. Formal definitions of grammatical rules for structuring sentences were for many years highly influential, notably Chomsky's theories of linguistics. With linguists supporting the generation of language rules, syntax was closely examined. The earliest approaches applying machine learning and artificial intelligence involved decision trees, where clearly defined "If-Then" rules could be specified by an expert.

Since the 1980s, largely due to the improvements in computational processing power and storage, statistical methods have become more popular, being applied to develop a probabilistic approach to analysing large corpora. This revolution in the early 1990s led to the development of "Corpus Linguistics", where large corpora (large collections of texts) are processed automatically by the machine to make inferences and decisions based on statistically generated models, and probabilities.

1.2 Basic NLP Tasks

This section introduces some of the common NLP tasks which are often performed as pre-processing before text can really be mined. Most of these tasks are concerned with syntax, or grammar pre-processing.

1.2.1 Tokenization

Tokenization refers to the task of breaking a string of text down into tokens, and while this could concern breaking a text into paragraphs, or sentences, generally the concern is word level segmentation. Initially this appears as though it should be a relatively trivial task. If we take the text "The Quick Brown Fox Jumps Over The Lazy Dog", this consists of 35 characters, and 8 spaces. Using the spaces as a delimiter would return 9 tokens as follows; "The", "Quick", "Brown", "Fox", "Jumps", "Over", "The", "Lazy" and "Dog". Depending on what the tokens will be used for it could be represented in a variety of ways, for example as XML.

```
<sentence>
  <word>The</word>
  <word>Quick</word>
  <word>Brown</word>
```

```
  <word>Fox</word>
  <word>Jumps</word>
  <word>Over</word>
  <word>The</word>
  <word>Lazy</word>
  <word>Dog</word>
</sentence>
```

Alternatively it could be stored as a tree like s-expression.

```
(sentence
   (word The)
   (word quick)
   (word brown)
   (word fox)
   (word jumps)
   (word over)
   (word the)
   (word lazy)
   (word dog))
```

Or simply as an array of words.

Array ([0] => The [1] => quick [2] => brown [3] => fox [4] => jumps [5] => over [6] => the [7] => lazy [8] => dog.)

In English, a space is generally a good approximation of the separator between words, and some programming languages have existing functions to split a string based on a specified delimiter, such as the explode() function in PHP, and the split() function in Java. Some basic parsing may then be needed to format the results as required. However, while simply splitting text up based on spaces gives decent results, it may not be sufficient, and tokenization may need to consider further cases. An initial complication of this approach comes from punctuation. If we consider the token “I’m”, there are choices about how it might be broken down;

a) I am
b) Im
c) I m

To tokenize “I’m” into 2 separate words (as in option a) requires some further sophistication in understanding the reason behind the apostrophe. Option b) is achieved by simply removing punctuation before parsing the text, while option c) is achieved by replacing punctuation with a space which is then removed later, neither of which is particularly satisfactory. There is however a tradeoff between the processing required to handle each piece of punctuation, versus the added information provided by words such as ‘am’, which are often later removed as stop words.

A second punctuation issue comes from dealing with abbreviations, such as “Dr.” for Doctor, or “Sun.” for Sunday. In the first case, “Dr.” could be simply an alias for “Doctor”, while “Sun”, as a token, becomes ambiguous between a day of the week, or the star at the center of our solar system. Hyphenation offers a further issue, as tokens such as “trade-off” could be tokenized as “trade” and “off”, or “tradeoff”, depending on how punctuation is handled. Punctuation such as this is often inconsistently used throughout texts, so for most of the applications discussed in this text, the principle of “KISS” (keep it simple stupid) is applied, with punctuation and capitalisation removed.

In English, words are conveniently segmented with a space, but this isn’t the case in all languages, which presents further tokenization challenges. In languages such as Thai, Japanese, Korean or Chinese there are no spaces to mark word separation, so a tokenizer needs to identify different word tokens. Consider the following expression in Thai;

ฉันอยากไปกินข้าว

With no spaces to mark word segmentation, further efforts are needed to separate each token.

ฉัน - I

อยาก - Want

ไป - Go

กิน - Eat