FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY
FAKULTEIT INGENIEURSWESE, BOU-OMGEWING EN INLIGTINGTEGNOLOGIE

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

INDIVIDUAL ASSIGNMENT / *INDIVIDUELE WERKSOPDRAG*

Surname / *Van*	**Liu**							
Initials */ Voorletters*	**YY**							
Student Number / *Studentenommer*	**1**	**3**	**0**	**4**	**3**	**7**	**0**	**7**
Module Code / *Modulekode*				**INF :**	**8**	**4**	**0**	
Assignment number / *Opdrag nommer*	**Final Research Paper**							
Name of Lecturer / *Naam van Dosent*	**Dr JP van Deventer**							
Date of submission / *Datum ingehandig*	**01/12/2018**							
Declaration / *Verklaring*: I declare that this assignment, submitted by me, is my own work and that I have referenced all the sources that I have used. */ Ek verklaar dat hierdie opdrag wat deur my ingehandig word, my eie werk is en dat ek na al die bronne wat ek gebruik het, verwys het.*								
Signature of student / *Handtekening van student*	**Yi-Yu Liu**							

MARK */ PUNT*	

THE ADVANCES OF STEMMING ALGORITHMS IN TEXT ANALYSIS FROM 2013 TO 2018

by

Yi-Yu (Bruce) Liu

13043707

Submitted in partial fulfilment of the requirements for the degree

MCom in Informatics

in the

FACULTY OF ECONOMIC AND MANAGEMENT SCIENCES

at the

UNIVERSITY OF PRETORIA

Study leader:

Dr JP Van Deventer

Mr RM Kruger

Date of submission

21 May 2019

ACKNOWLEDGEMENTS

I would like to thank my sister and my parents.

I would like to thank my supervisors. There is a Chinese proverb that says “一日為師, 終生為父”. This means: “One day as a teacher; a lifetime as a father”. Respect and filial piety come with it.

Somehow, I believe this to be so true. I speak to my supervisors about everything in life. I learn from them and I will always respect them like my own parents. I will forever look up to them like my parents.

I would like to thank Coffee and Milk. I have had a strong relationship with you and we bonded quite well.

Lastly and most importantly, I need to thank God for giving me this opportunity to learn; giving me whatever result is gained from this experience. Everything that I have today, every glory that I reap, is the result of your blessing.

ABSTRACT

Stemming is an activity within the pre-processing step of Text Analysis. It plays a role in the Text Analysis results. It drives Data Mining in fields such as Business Information Systems. Eight percent of existing organisational data that contributes Big Data is in an unstructured format. One of the focus areas within the concept of "Big Data" is the complexity of processing the data and being able to represent the results in such a way that they are easily understood. This challenge has been taken up by researchers over time.

To determine the advances in Stemming Algorithm research, a systematic review was performed on articles on Stemming Algorithms published in journals from 2013 to 2018. Data was collected from accessible scholarly databases. The articles were then filtered by year and topic. The remaining articles were processed through a set of methodological quality criteria. The final articles were put through a bi-gram Text Analysis process to answer the research questions.

The results concluded that the research focus for Stemming Algorithms has started to decrease as it reaches the plateau of productivity. The results show an evident drop in the collected articles from 58 in 2017 to 19 in 2018. Results show that information retrieval is still a common field of application for Stemming Algorithms. A major unexpected set of themes revolves around artificial intelligence, based on an increase in interest in this topic. Results show that a focus on Stemming Algorithms has shifted away from its development and moved towards its application. There is also a high interest in social media as an application of Stemming Algorithms. Future research suggestions include designing a Stemming Algorithm that would automatically and responsively adapt to the historical and morphological changes of language text.

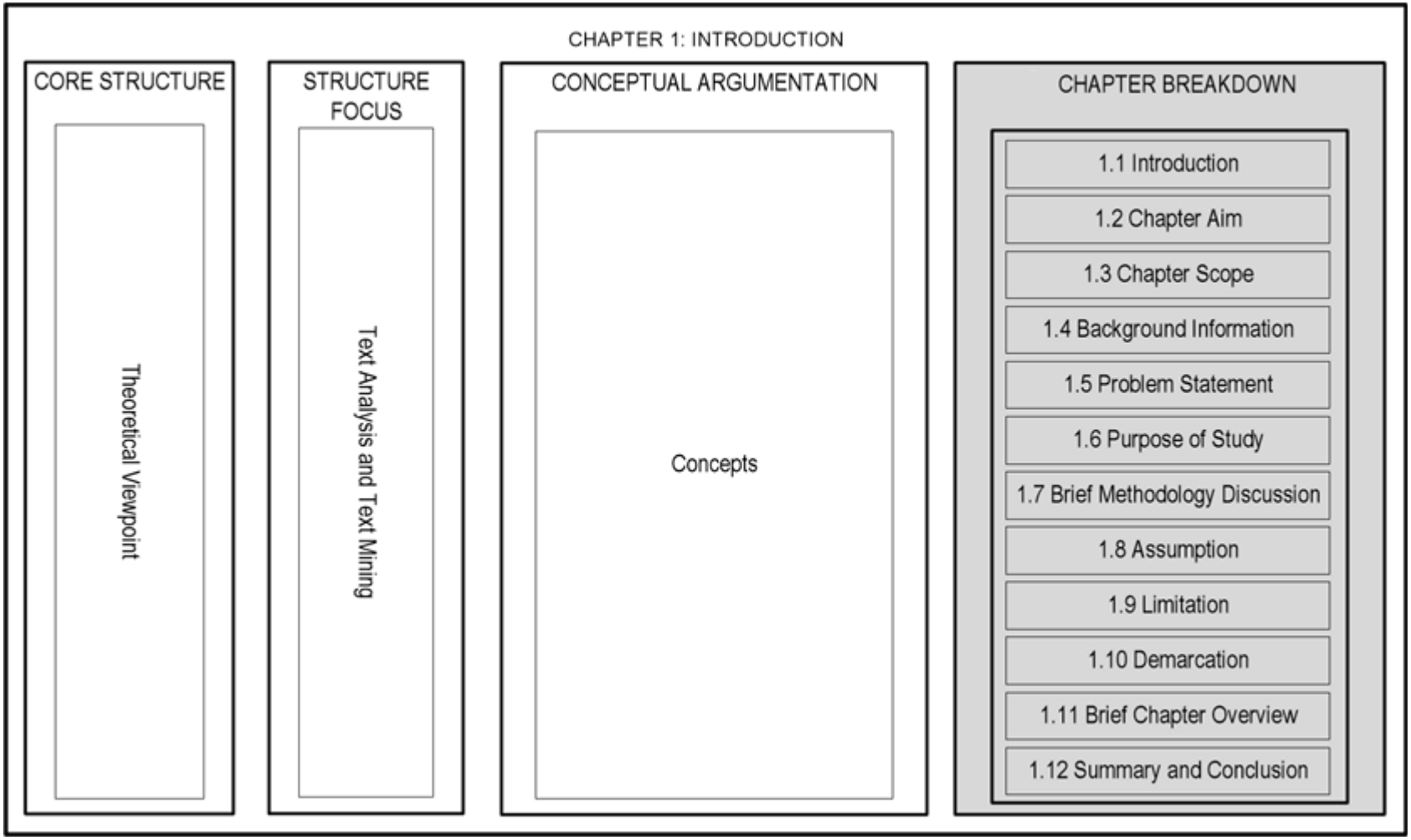
Keywords: Stemming Algorithm, Text Analysis, Big Data, systematic review

TABLE OF CONTENTS

LIST OF CHAPTER MAPS

LIST OF FIGURES

LIST OF TABLES



Chapter Map 1: Introduction

CHAPTER 1:
INTRODUCTION TO THE ADVANCES IN STEMMING ALGORITHMS OVER THE PAST SIX YEARS

1.1 INTRODUCTION

A Stemming Algorithm groups words of the same or similar meanings together (Porter, 1980). Examples of words that are similar in meaning are “managers”, “management” and “managerial”. These words all revolve around the conceptual understanding of the word “manage”, just in a different format. To understand why this matters, one needs to understand where the study of Stemming Algorithms fits into the pool of knowledge on the subject. The next few paragraphs will briefly contextualise Stemming Algorithms and discuss how Stemming Algorithms fit into businesses.

Businesses make use of information systems to enhance their information flow; thus, information systems can be regarded as a factor of business. Data Mining forms part of information systems, since Data Mining is a way to deal with data in a systematic way to retrieve useful information from the data (Shaw, Subramaniam, Tan & Welge, 2001). Text Mining forms part of Data Mining since text is a type of data. Text Mining is a process that is used to analyse a body of text to identify trends and patterns between words, entities and concepts of the text.

Text Mining is a way to understand the main concepts within a body of text (Feldman & Sanger, 2007). Understanding the main concepts within a body of text can assist a company with decision making (Porter & Cunningham, 2005). Text Mining encapsulates Text Analysis, where Text Analysis is a means of identifying trends and patterns. With Text Analysis, diagrams and graphs can be drawn to visualise trends and patterns (Feldman & Sanger, 2007). The Stemming Algorithm is therefore a step in the Text Mining process.

The purpose of this study is to aggregate the existing research that caters for the Stemming Algorithm as part of Text Mining or Text Analysis. In addition to grouping words of the same or similar meaning together, Stemming Algorithms also serve to improve Text Mining by indexing the variation of words as one word instead of different words. Indexing words from collected sources in such a way will speed up the Text Mining process. This also reduces the variety of words that are required to pull the results obtained from Text Mining, as the words have been included in the same groups (Moral, De Antonio, Imbert & Ramírez, 2014).

Various methods and algorithms have been established to perform Stemming Algorithms. Some common algorithms are Porters and Snowball (Berry, 2004). These are explained in more detail in Chapter 3. Porter (1980) published an article on a Stemming Algorithm in 1980, which was considered an early establishment. In 2018, Sugumar (2018) published an article describing how to improve the performance of an existing Stemming Algorithm. Since the paper of Porter (1980), there have been developments in research on the topic of Stemming Algorithms. To understand the changes that have taken place on the topic over time, the main purpose of this research can be phrased as follows:

What are the advances of Stemming Algorithm in Text Analysis?

The scope of this research is to determine trends in the development of Stemming Algorithms. Accomplishing this research serves the purpose of making Stemming Algorithms aggregated and reviewed, allowing the update of existing Text Mining applications, that incorporate Stemming Algorithms, to optimise their Text Mining practices.

1.2 AIM OF THE CHAPTER

This chapter introduces the research by establishing the problem. It then provides a set of research questions that are of interest, as well as the objectives of the study. Additionally, it states the aim of the chapter by providing the scope of the research and clearly defining the areas of interest related to the problem statement.

1.3 SCOPE OF THE CHAPTER

To achieve the aim of the chapter, background information is presented, followed by the problem statement, where the research question and objectives are discussed. To define the scope of the research, some of the required demarcations and limitations are defined, and the reasons for these demarcations and limitations are provided. Finally, the chapter gives a breakdown of the subsequent chapters in the dissertation.

1.4 BACKGROUND INFORMATION

To provide an example of how important language is, some statistics from Google and Facebook are discussed. In 2012, Facebook had 955 million active accounts over 70 different language platforms. Google can support different sets of services for clients. Firstly, it monitors 7.2 billion pages every day. Secondly, it processes 20 petabytes of data every day, while translating this into 66 different languages. Big Data is a term that was developed to describe a very large amount of data from different sources of a variety of types (Sagiroglu & Sinanc, 2013).

One of the focus areas within the concept of “Big Data” is the simple difficulty of processing the data and being able to represent the results in a way that can be easily understood. Organisations

in any industry have Big Data that can be used because value can be gathered from such data (Sagiroglu & Sinanc, 2013).

Organisations have started paying more attention to Big Data and have grown interested in using it for decision making (Labrinidis & Jagadish, 2012). Eighty per cent of existing organisational data that contributes to Big Data is in an unstructured format (Papadopoulos et al., 2017). Unstructured data is usually in the form of a body of text that can include email text, documents, social media comments and instant messages (Müller, Junglas, Debortoli & Vom Brocke, 2016).

The valuable information extracted from the sources has great potential for a single person or an organisation (Sagiroglu & Sinanc, 2013). The current concern (the main research question of this study) pertains to existing research that has been established to deal with the different languages within Big Data.

Manyika et al. (2011) discuss how McKinsey Global Institute specified that harnessing value from Big Data can potentially allow organisations to analyse in-store behaviour, create new business models and, in some cases, even influence human behaviour by customising actions for suitable products and services. The harnessing of value from Big Data ultimately assists with data-driven decision making (Manyika et al., 2011).

Due to the large volumes of available data, companies worldwide are extracting information from textual data for a competitive advantage (Provost & Fawcett, 2013). The process of extracting information from textual data is known as Text Analysis. Text Analysis falls within Text Mining[1] (Gaikwad, Chaugule & Patil, 2014; Tan, 1999). For Text Mining to take place, the first thing that needs to happen is that the person who is carrying out the Text Mining needs to collect data.

[1] Text Mining is a concept used to describe the extracting of trends and information from textual data, which forms part of Data Mining as a practice (Vijayarani, Ilamathi & Nithya, 2015).

Before the data is taken to be processed, some pre-processing activities need to take place to prepare the data. Different techniques can be used to process the data, based on the researcher's requirements. These will be discussed in Chapter 2. Once the data has been analysed, the results can be displayed to communicate valuable knowledge (Gaikwad, Chaugule & Patil, 2014).

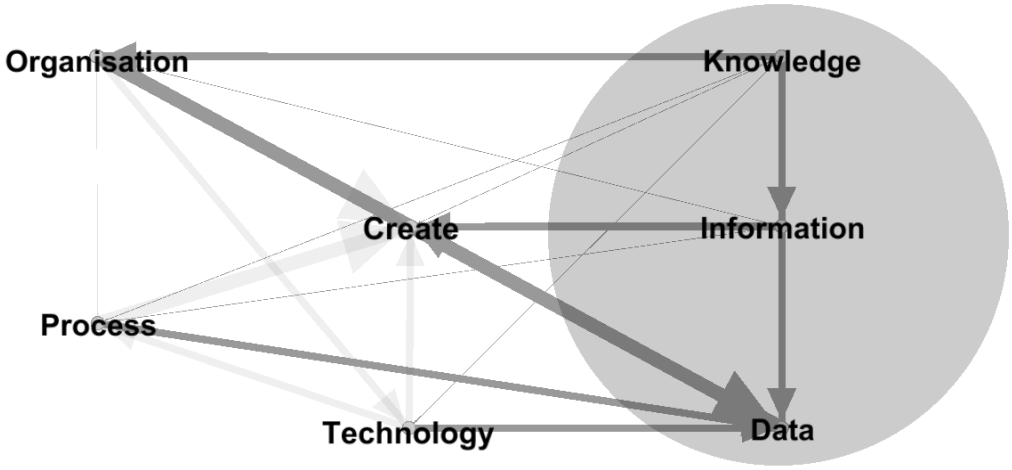


Figure 1: An example of Text Mining results (Van Deventer, 2014)

In Figure 1, an example of Text Mining results can be seen. The accuracy of the results does not matter, but rather the method according to which the results were gathered. Figure 1 presents the conceptual connections to the theme "knowledge".

In Figure 1, there are arrows and words. The words represent the "themes" that are presented. The arrows direct the relationship between the themes. The thickness of the arrows indicates the strength of the relationship between the themes. The darkness of the arrows' shading indicates interest from the author. The circle indicates themes of interest from the author.

From Figure 1, one can say that knowledge leads to information, which indicates the direction of the relationship. There is also a relationship leading from "knowledge" to "organisation" to "create" and, finally, to "information" or vice versa.

To produce such a result, the first step in the process is to identify the existence of such words in a text document (Feldman & Sanger, 2007). Details regarding the process will be discussed in Chapter 2. However, there is a typical step in the process that combines different forms of the same word into one. This is called stemming (Lovins, 1968). A given example in the diagram in Figure 1 would be combining the words “creating”, “creation” and “creative” into the root word: “create”. This is done because all three words serve the same purposes and act as sub-concepts for the defined main concept, “create.” This sub-process – stemming – of Text Mining, removes the prefixes and suffixes of a word to gain the “root” word. The root word holds the significant meaning behind all the variations of the word (Berry, 2004). The “stem” of the word is regarded as the naked word without a prefix or suffix (Moral et al., 2014). From the example, it can be seen that stemming is the grouping of words that contain the same concept. Therefore, we can say that the purpose of stemming is to consolidate ideas within the analysis to determine further understanding within the meaning of the text, such as the relationships between words or the intensity of concepts within context (Moral et al., 2014). Placement of stemming in the text mining process will be discussed in more detail in Chapter 2.

Without the assistance of stemming, a different understanding of what is known of the concept “create” would result. The associated ideas of “create” are not different from the associated ideas of “creating”. In Figure 1, both these words are conceptually categorised into the same understanding. Therefore, this is a required process.

There are three different types of stemming: the removal of the suffixes of words based on a guided set of rules to classify them based on their root word, using statistical formulae to synthesise the words and group them together, and comparing the words to an existing corpus of corresponding stems (Vijayarani et al., 2015). The different methods of stemming will be discussed in more detail in Chapter 3. Different types of stemming methods were created to tackle

the problems encountered in the stemming process. Some of the problems encountered relate to differences between languages. The difficulty with stemming words in different languages includes scripting differences, wording differences and structural or grammatical difference (Basiri, Ghasem-Aghaee, & Reza, 2017).

Within scripting differences, each language has its own way of documenting its communication on paper. There are languages that use the English alphabet to complete the task. Some languages, however, do not use the alphabet at all. Examples of such languages are Mandarin Chinese, Korean, Japanese and Arabic. The stemming process for these types of languages differs based on the character differences of the respective languages (Basiri, Ghasem-Aghaee & Reza, 2017).

The second difficulty lies in the way different phonetic languages spell words. In turn, this means that the suffixes and prefixes of words are different across different languages. This causes predefined rule-based Stemming Algorithms to experience difficulty in stemming different languages. Some languages are conceptual languages, and others are semantic languages, which brings about a different difficulty when it comes to stemming. Conceptual languages group words differently to semantic languages (Basiri et al., 2017).

Lastly, the sentence structure of different languages is different (Ameka, 2016). The Text Mining process will yield different results for different languages because of the differences in sentence structure. This causes difficulty for stemming as the words at the beginning of a sentence can mean something different to those at the end of a sentence (Basiri et al., 2017). Therefore, the question arises as to whether words should be stemmed and grouped together or kept separate since the meaning of the same word changes according to its position in the sentence. If they should be kept together, how would technology be able to determine the differences?

Having acknowledged the problems mentioned above, one would need to know whether anything has been done to solve these problems and what has been accomplished? This question leads to the problem statement of this research. The next sub-section will discuss the problem that is faced by providing an understanding of the objectives of the study.

1.5 PROBLEM STATEMENT

In the background information, the major advances in Stemming Algorithms to support different languages were discussed. In this section, the main purpose and concern of this research study are presented.

A simple search on Google Scholar on 22 February 2018 for articles on “Stemming Algorithms” published between 1900 and 2018 on the Open WorldCat database indicated 79 800 results. These are only the results that are on the Open WorldCat database. There are other databases that also contain articles about Stemming Algorithms. This shows that much research has been done on Stemming Algorithms since 1900.

With so much research on the topic of Stemming Algorithms, it is difficult to deliver a meticulous summary of all the available primary research in response to the existing research on Stemming Algorithms over the past six years.

The problems associated with Stemming Algorithms for different languages mentioned in the background information leads us to this research study. Based on Gartner’s hype cycle presented in July 2013, Text Analysis reaches its plateau of productivity in two to five years. This can be seen in Figure 2.

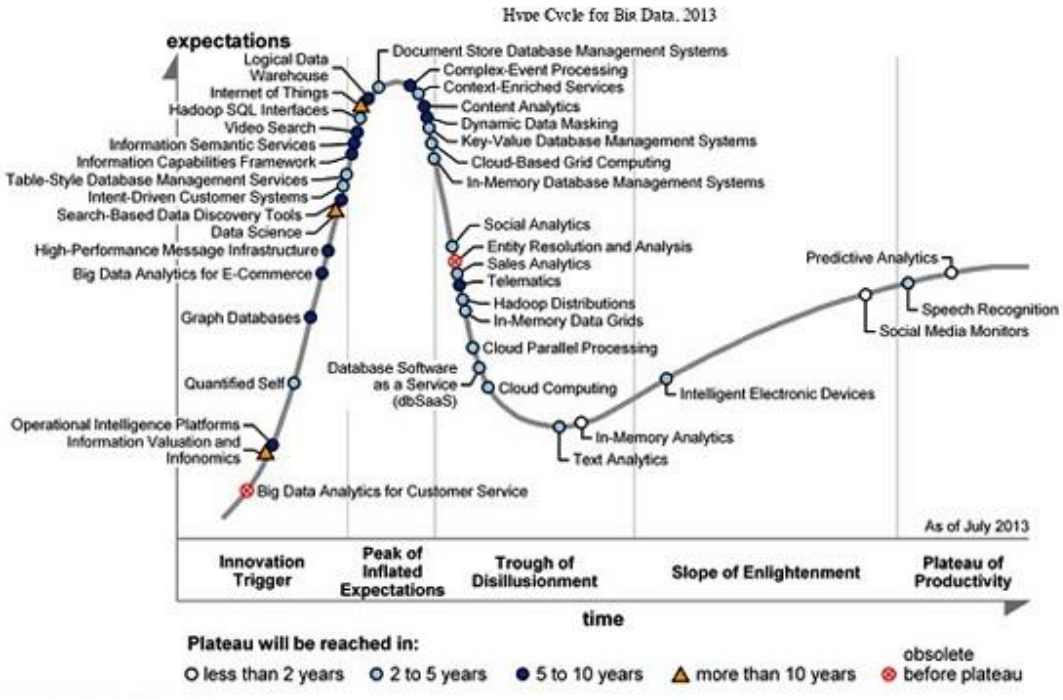


Figure 2: Gartner's hype cycle of 2013 (Heudecker, 2013)

Figure 2 shows that Text Analysis is in the stage of “trough of disillusionment”, which shows that it has a low interest during the specified years. The legend indicates that Text Analysis reaches a plateau in two to five years. This means that, during these two to five years, Text Analysis finds itself in the “slope of enlightenment”. This is where knowledge on the topic is building up. At the end of this period, the adoption of technology in business would have taken place. Analysing the knowledge established during this period provides a perspective of the direction in which that particular topic is advancing (Linden & Fenn, 2003). In the case of this research, the focus is on Stemming Algorithms as a sub-set of Text Analytics.

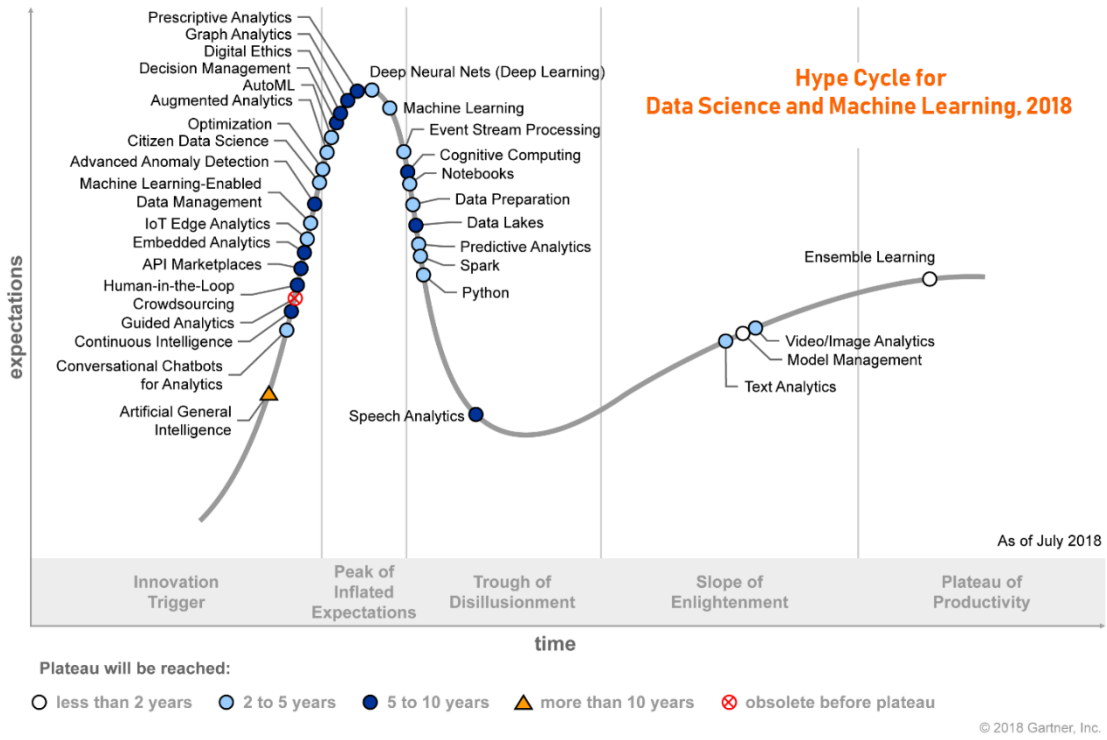


Figure 3: Gartner's hype cycle of 2018 for data science and machine learning (Krensky & Hare, 2018)

Figure 3 shows Gartner's hype cycle presented again in 2018. In Figure 3, it is evident that "Text Analytics" did not reach the "plateau of productivity" as predicted. Text Analytics is still in the "slope of enlightenment". Therefore, the years 2013 to 2018 were decided on for the research evaluation for Stemming Algorithms based on the gathered slope of enlightenment. The year 2019 has been removed from the study to avoid inconsistencies as it was incomplete at the time of this study.

In conclusion, the problem we are facing now is that there is no existing consolidated systematic review of the established Stemming Algorithms between 2013 and 2018 to understand the establishment of Stemming Algorithms.

1.6 PURPOSE OF THE STUDY/RESEARCH OBJECTIVES

This section discusses all the questions, sub-questions, goals and objectives of the study. The main question of this study is as follows:

What are the advances of Stemming Algorithms in Text Analysis over the past six years (2013 to 2018)?

To support this question, the following sub-questions have been established to break down the research question:

1. *What are the characteristics of Stemming Algorithms?*
2. *What are the applications of common Stemming Algorithms in Text Analysis?*
3. *What are the problems associated with Stemming Algorithms in text and Text Analysis?*
4. *How have the Stemming Algorithms been applied and changed in Text Mining over the past six years (2013 to 2018)?*

The aim of this research study is to consolidate the existing researched knowledge of Stemming Algorithms over the past six years into one research dissertation. The following objectives are determined to answer the questions stated above.

1. Identify the characteristics of Stemming Algorithms.
2. Consider the application of common Stemming Algorithms in Text Analysis.
3. Identify the problems with Stemming Algorithms in text and Text Analysis.
4. Identify how Stemming Algorithms have changed in Text Mining over the past six years (2013 to 2018).

The combined understanding of all the questions and sub-questions leads to a greater understanding of the problems faced with Stemming Algorithms over the past six years to achieve the aim stipulated above. The main goal of this study is to consolidate the existing knowledge of Stemming Algorithms over the past six years into one dissertation. This can assist in creating cohesion of understanding of the knowledge of Stemming Algorithms.

In summary, this section has discussed all the questions, sub-questions, goals and objectives that this research study intends to achieve.

1.7 METHODOLOGY

This section gives an overview of the methodology applied to this research and presented in this dissertation.

The systematic review methodology was used for this study. This was based on the version of the methodology presented by Okoli and Schabram (2010), accompanied by the knowledge and understanding of Attride-Stirling (2001). Within the methodology proposed by Okoli and Schabram (2010), there are four main stages: planning, selection, extraction and reporting.

To fulfil the steps required by Okoli and Schabram (2010), Google Scholar was chosen as a platform for the collection of data. The following specifications were provided on the databases: specify only empirical research (journal articles); specify only pdf files; specify only articles written in English; and specify only terms relating to the theme of the topic, such as Stemming Algorithms, stemmer, Text Mining and Text Analysis.

WebHarvy was used to search each Google Scholar result and download each respective article from the results. Themes were identified from the downloaded articles according to a Text

Analysis process. The Text Analysis conducted for this research was adapted from the research of Aryal, Gallivan and Tao (2015), whose research was used to identify themes from healthcare information systems research. The process included data collection and pre-processing, and a word list was ultimately produced. This data collection that formed part of the research differed from the WebHarvy approach and the presentation of the results. The data preparation and presentation were adopted from Van Deventer (2014). Finally, the calculation and interpretation of the results were adopted from Kerlinger and Lee (2000).

To analyse the results, the calculation and interpretation explained by Kerlinger and Lee (2000) were carried out, and the correlation between two factors compared to each other. This interpretation was then used to answer the questions set out in the problem statement.

1.8 ASSUMPTIONS

This sub-section reflects on the assumptions made while conducting the research. The assumptions listed below are not exhaustive and may include items of interest that have as yet not been considered. Based on Gartner's hype cycle provided in the problem statement, it is assumed that Stemming Algorithm trends can be extracted from articles published between 2013 and 2018. Further research can extend to beyond the six-year period. Furthermore, it is assumed that the data used for the research articles went through a detailed review process to ensure that it is valid and contributes to the body of knowledge on Stemming Algorithms and Text Mining.

It is trusted that the process of gathering literature, as discussed in Chapter 4, was carried out in good faith. Within the analysis of the results or the reporting stage of the methodology, the Pareto Principle is applied. This principle is used to eliminate the datasets that do not contribute to the main concepts of the results. It is assumed that the principle holds true and is applicable to this research.

1.9 LIMITATIONS

This section discusses the limitations of the research. These limitations include, but are not limited to, the fact that the articles are all empirical research articles and conference proceedings on Stemming Algorithms from 2013 to 2018. Furthermore, for practical reasons, articles to which the University of Pretoria has access are the only articles taken into consideration. In addition, only articles written in English are taken into consideration. Further research can include other types of research articles, articles with external access permissions and articles written in different languages.

Another limitation lies in the analysis of the results, as it may lead to subjective understandings of the author due to his limited knowledge and intellectual capability. However, the author has performed a literature review in an effort to combat such limitations.

With regard to data collection, there is a limited possibility to validate the results of the research articles within the data collection process that is applied within this research. Therefore, it is a limitation that the research results within each collected article are not validated.

The next section will discuss the demarcations for the research.

1.10 DEMARCATION

This section discusses the demarcations that will be applied to the rest of the dissertation. To fulfil the requirements of the problem statement, research articles published between 2013 and 2018 are collected. Articles published in 2019 are not included as this would cause instability in the sample data. Articles written in English are taken into consideration even if they are written for a different language. If the main body of the research is not scripted in English, it is excluded from the research study.

A set of criteria has been followed to eliminate articles that could not be integrated into this research study. This criteria includes non-empirical studies, non-scripted methodologies, incomprehensible analyses, and a misalignment between the results and the conclusions of the collected articles. Furthermore, the algorithms and mining techniques established within the literature are not tested.

Research was not done on the conceptual difficulties and differences of language connotations. The focus of the research is on the technology that has been used to work with the text. Therefore, language connotations are not a primary focus of the research. Additionally, this study does not focus on Eastern languages such as Mandarin, Japanese or Korean.

This study does not include an in-depth study of the different methods of data collection for Text Mining. Only some examples are provided, since data collection is not the main focus of the results and does not deliver the main contribution. Further research can evaluate the different methods of data collection for Text Mining and its influences on Stemming Algorithms.

This study does not include an in-depth study into statistical calculations, formulae and codes within the algorithms of each Stemming Algorithm mentioned in Chapter 3. Only high-level steps are provided as the minor details within the statistical calculations of each algorithm are not the main focus of the results, nor do they deliver the main contribution.

As a demarcation, the data collection method has limited the search to pdf files, since academic articles are published as pdf files in a downloadable format. However, future research can include content from non-academic articles, as well as different file formats.

Lastly, this study does not cover the topic of lemmatisation. The next section gives the brief chapter overview.

1.11 BRIEF CHAPTER OVERVIEW

This section discusses the structure of the dissertation, and discusses the content of each chapter. The dissertation maps a distinct flow in arguments following a particular construction of structures and arguments. Figure 4 gives a breakdown of the content of the chapters according to a core structure, structure focus and conceptual argument. The chapter breakdown diagram can be referenced throughout the process of reading the dissertation to pinpoint the paradigm and constructs of the referenced section.

As represented in Figure 4, the foundations of the research are segmented into three sections: theoretical viewpoint, practitioner's viewpoint and integration. The theoretical viewpoint considers the established knowledge from literature, whereas the practitioner's viewpoint executes tasks as an extension to the literature, and the integration ties up all the loose ends.

Within the theoretical viewpoint, discussions around the concepts related to Stemming Algorithms are carried out to identify the problems associated with Stemming Algorithms within the literature. From a practitioner's viewpoint, the procedures of the research are carried out, in addition to a descriptive analysis as a form of practice. Integration completes the pattern, matching a building blocks identification by further analysing the results of the research. The final section of integration provides a conclusion to the research by tying up all the loose ends and answering the research question.

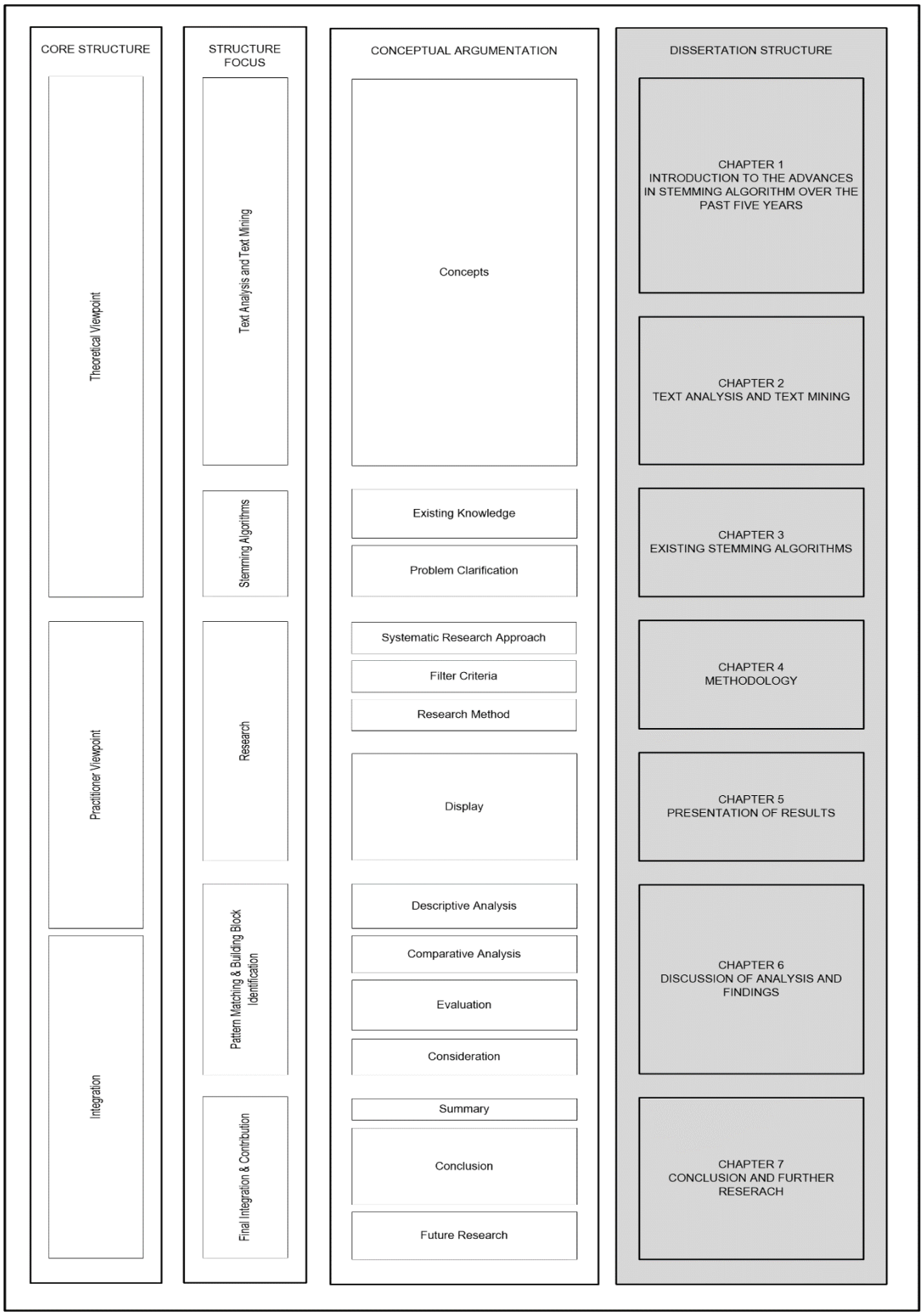


Figure 4: Chapter breakdown

Figure 4 illustrates the mapping of the different chapters to the core structure, focus and conceptual argument.

In Chapter 1, the research begins with a theoretical viewpoint to establish the field of study. As a structural focus, the established field of Stemming Algorithms is discussed to introduce the reading into problem areas and to clearly define the research objectives.

Chapter 2 discusses the concepts of Text Mining and Text Analysis in detail to establish the application of Stemming Algorithms and place them in the Text Mining or Text Analysis process. It includes a discussion of the steps and processes taken within Text Mining and Text Analysis respectively. It then focuses on the Stemming Algorithm step.

Chapter 3 presents the establishment of existing Stemming Algorithms over time. It draws a classification of the Stemming Algorithms into its respective categories and extends the algorithm to the research problem. This chapter provides the results for interpretation at a comprehensible level.

In Chapter 4, the practitioner's viewpoint comes into play. To carry out the research, practical activities take place. This chapter discusses what practical research needs to be carried out to fulfil the research goal. The research follows a systematic review with a set of specific filter criteria and steps. This chapter discusses these in detail.

Chapter 5 displays the research results carried out from the practitioner's viewpoint. The purpose of this chapter is to display the results and point out patterns and trends. The results will be discussed in the next chapter. This chapter presents the results of the research at a digestible level before moving on to the discussion.

Chapter 6 extracts points from the previous chapter. The focus of this chapter is on pattern matching and building block identification. The building blocks consist of knowledge areas that contribute to the objective of the study. The descriptive analysis is the practitioner's viewpoint, as it is a simple performance of analysis. The comparative analysis begins to tie theoretical concepts to the results. Therefore, the integration in the core structure begins. Finally, this chapter discusses the evolution of the analysis and provides considerations around the research.

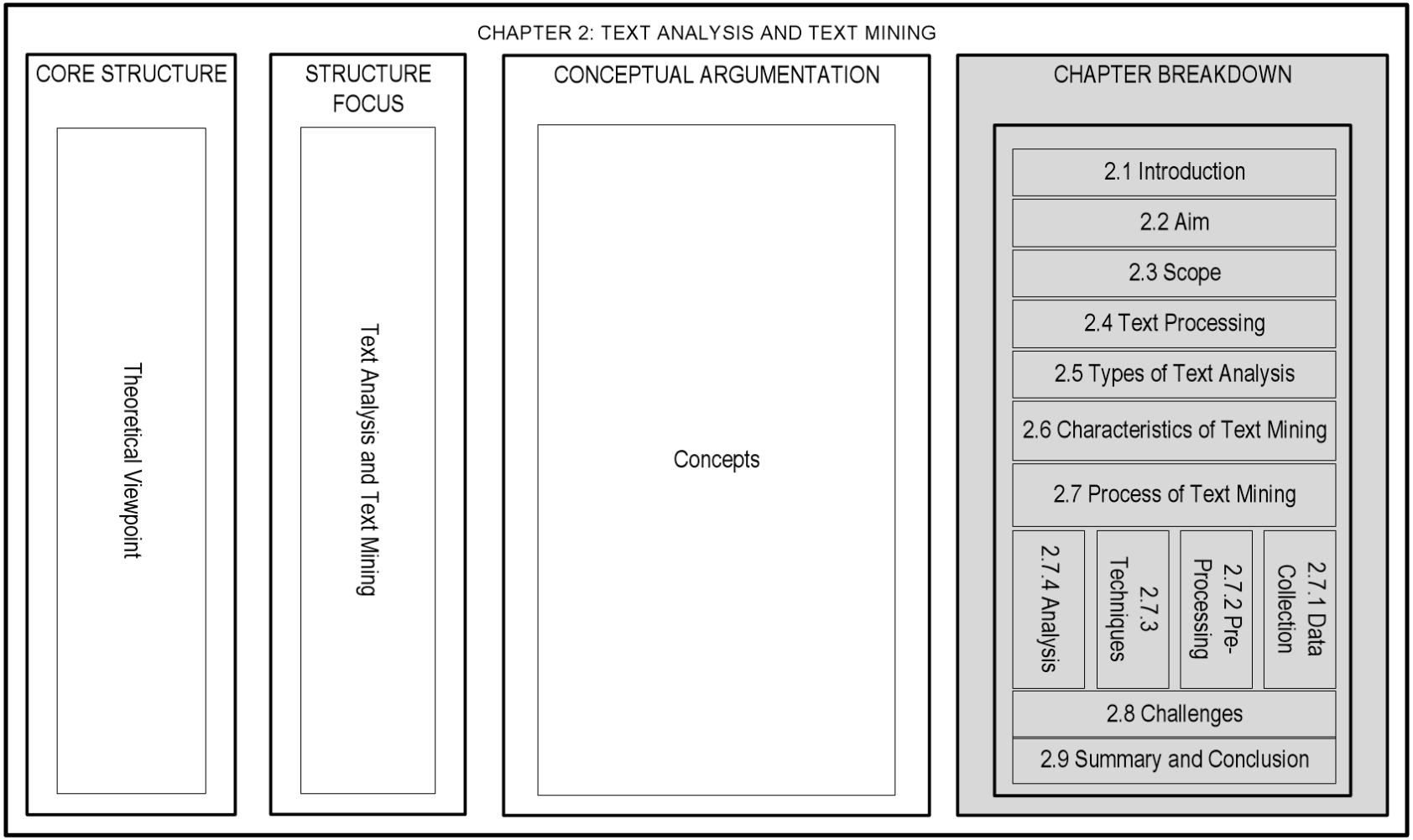
In Chapter 7, the conclusion to the research is drawn. It merges the theoretical and practitioner viewpoints. It outlines the findings of the research to answer the research problems and objectives provided in Chapter 1.

1.12 SUMMARY AND CONCLUSION

In Chapter 1, aspects associated with the background information are discussed. The problem statement, purpose of the study, assumptions and demarcations are given, and a brief chapter overview is provided. In the background information, it is indicated that companies have eighty per cent of their data in unstructured formats. In order to extract valuable information from it, Text Mining practices are required. Within the Text Analysis process, there is one item component under a step called “stemming”. Stemming is the grouping of words that revolve around the same concepts in the same classification by removing the suffixes of words.

The main difficulty with stemming is that there is more than one exception to the defined aggregate rule to establish a complete set of rules. Different languages also have different ways of spelling words, which create different suffixes. This causes another difficulty for stemming as a separate algorithm is required for each language. Based on the given points, one can conclude that Text Analysis and Stemming Algorithms have difficulty fulfilling the purposes that have been set out when it comes to different languages.

In order to consolidate a suitable time frame for research articles, Gartner's hype cycle was taken into consideration. Since Stemming Algorithms are a subset of Text Analysis, Text Analysis is taken as the benchmark for purposes of comparison. Text Analysis reaches its slope of enlightenment within two to six years after 2013. A suitable period would therefore be from 2013 to 2018. Based on the analysis of Gartner's hype cycle, the main focus of the study should be on the six years between 2013 and 2018. The next chapter discusses the Text Analysis and processing of the text.



Chapter Map 2: Text Analysis and Text Mining

CHAPTER 2:
TEXT ANALYSIS AND TEXT MINING

2.1 INTRODUCTION

This chapter discusses the typical processes of Text Analysis and Text Mining to understand where Stemming Algorithms can be applied within Text Analysis and Text Mining. This chapter begins by discussing Text Processing and how Text Analysis and Text Mining relate to Text Processing. This is followed by a discussion of the Text Mining process that makes use of Stemming Algorithms. The purpose of this discussion is to contextualise Stemming Algorithms before presenting a more in-depth discussion of Stemming Algorithms in the next chapter.

2.2 AIM OF THE CHAPTER

This chapter aims to provide an understanding of the context around Stemming Algorithms. Chapter 1 established the research questions around Stemming Algorithm and the applications thereof. With the establishment of an understanding of the processes within Text Analysis and Text Mining, it is possible to conclude the research in the later chapters, since it encapsulates Stemming Algorithms and the applications of Stemming Algorithms is applicable to the rest of the Text Mining process. The next chapter will discuss Stemming Algorithms in detail. The following section explains how the scope of this chapter will achieve this aim.

2.3 SCOPE OF THE CHAPTER

To achieve an understanding of Stemming Algorithms, this chapter will begin by discussing Text Processing, followed by an explanation of the different types of Text Analysis. The characteristics of Text Analysis are then defined. To contextualise Stemming Algorithms within Text Mining, the typical steps followed in Text Analysis are discussed.

The steps in the Text Mining process are discussed, starting with the cleaning of the data as a sub-process. The different methods to accomplish each sub-process are also discussed, since Stemming Algorithms can be applied to other components in the Text Mining process. This is followed by a discussion of the cleaning sub-process, where Stemming Algorithms exist. This chapter focuses on the application of Stemming Algorithms within the context of Text Mining.

2.4 TEXT PROCESSING

As set out in Chapter 1, humans understand textual data by analysing it on different levels based on the characteristic of “text”. Computers work with databases, which are data that is in a structured format. Textual data can also come in unstructured formats, for example emails, documents and web pages (Vijayarani et al., 2015). In order for computers to process text documents, the process of natural language processing, among others, needs to take place (Cai et al., 2016).

Natural language processing aims to provide a set of computational rules to understand textual data the way humans do. The purpose of this is for computers to extract knowledgeable constructs from the processed textual data (Cai et al., 2016). The set of rules that are provided to the computer allows it to analyse and create a structure with the unstructured textual data. Natural language processing is embedded into different steps of the Text Mining process. Any of the steps within the process that require the computer to analyse or create a structure of some sort to process unstructured textual data is inclusive in the study of natural language processing (Cai et al., 2016). The structures that are created from the Text Analysis process should assist with knowledge discovery (Uramoto et al., 2004).

It is possible to dig deeper into “natural language processing”, however, for the purpose of the research, not too much knowledge of this topic was required, since it is not a direct application of Stemming Algorithms. A brief understanding would therefore be sufficient.

Stemming Algorithms, however, play a role in the Text Mining process. Text Mining is a concept that is used to describe the gathering of trends and information from textual data, whereas Text Analysis is the analysis and understanding of textual data that is also carried out with trends and information. The two concepts began in two different sectors of interest: Text Mining as set out in the Data Mining sector, and Text Analysis as set out in the linguistics sector. They have become a merged concept as they both perform the same tasks, and are essentially the same thing (Vijayarani et al., 2015). Therefore, in this research, Text Analysis and Text Mining will be used interchangeably.

Before discussing the characteristics of Text Analysis, it is necessary to discuss the types of Text Analysis.

2.5 TYPES OF TEXT ANALYSIS

Text Mining originated from the task of processing large amounts of text to gather relevant data. Text Analysis is the act of acquiring knowledge from the mining processes. In the modern age, Text Analysis is incorporated into Text Mining and vice versa. Text Mining consists of the steps mentioned above up to the analysis stage. Text Analysis is the extraction of valuable information in the form of syntax, semantics, pragmatics and concepts (Cambria & White, 2014). Other Text Analysis options include phonetics, meaning and intention analysis (Bakhtin, 1977; Brown & Yule,

1983). However, for the purposes of this research, only conceptual, pragmatic, semantic and syntactic analyses are discussed in detail.

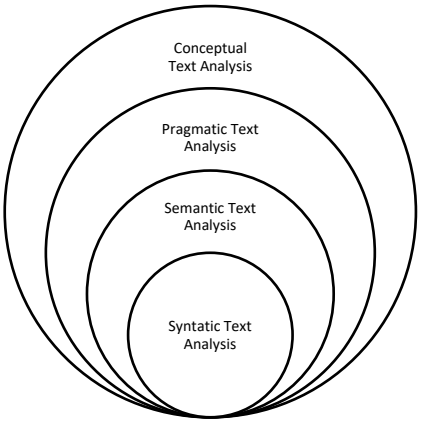


Figure 5: Types of Text Analysis (Cambria & White, 2014)

Figure 5 shows the classification of Text Analysis. Syntactic Text Analysis forms part of all the other types of Text Analysis. Semantic Text Analysis consists of syntactic Text Analysis and forms part of pragmatics, as well as conceptual Text Analysis. Pragmatic Text Analysis includes syntactic and semantic Text Analysis and forms part of conceptual Text Analysis. Conceptual Text Analysis embodies all the other forms of Text Analysis.

Language is defined as a method of human communication, either written or spoken. Both methods use words in a structured and conversational way (Cambria & White, 2014). These different classifications of Text Analysis assist in understanding each other. The four types of Text Analysis carry out communication from different levels of understanding (Schellens & Maes, 2000).

Syntactic analysis is the detailed examination of all the word and phrase arrangements that contribute to well-formed sentences in the body of text to discover the structure of language (Bessmertny, Platonov, Poleschuk & Pengyu, 2016; Naumov & Vykhovanets, 2016). Syntactic Text Analysis is done before the completion of pragmatic analysis (Kharlamov, Yermolenko & Zhonin, 2013). In the communication of languages, parties in communication need to understand the structure of the words that have been used. To accomplish this, the first step is to apply syntactic analysis. Once the parties in communication with each other are able to understand the structure of the language, they can then proceed to semantic analysis (Schellens & Maes, 2000).

Semantics is a branch of linguistics and logic that revolves around the meaning of expressed words. Semantic analysis has the aim of finding out the expressed view and opinion from a body of text, whether collectively or individually. There are two main categories of semantics: logical and lexical (Cambria & White, 2014).

Logical semantics deals with the physical senses of the human, the implications of the words and the references of the presupposition of the word. Lexical semantics deals with the underlying meaning behind the word and the relationship it has with other words. Sentiment analysis in practice is used to detect the feelings or opinions people have towards a particular topic (Cambria & White, 2014). In the communication of language, the parties involved need to have the skill of semantic analysis to communicate physical sense and presupposition with each other. However, human communication does not just stop there. This brings us to pragmatic analysis.

Pragmatic analysis is a detailed examination of the elements of a realistic conversation from a body of text in a way that is based on practical rather than theoretical considerations (Kharlamov et al., 2013). This method involves the analysis of the meaning that is transferred across the body of text based upon the author's intentions. This meaning is somewhat

“invisible” unless given the underlying knowledge of the author. The purpose of performing a pragmatic Text Analysis is to understand the underlying motives of the author at a deeper level (Kharlamov et al., 2013). For example, if we say: “he is going green”, it could mean that he is starting to work with recycled products or it could mean that his face is going green and he is about to become ill. The option is related to the context and intentions of the author. In the communication of language, the parties are required to apply pragmatic analysis to understand the context of the speaker or author. Up to now, the types of Text Analysis have been around sentences. Fortunately, humans can communicate with more than one sentence at a time, which leads us to conceptual analysis.

Conceptual Text Analysis is the detailed examination of the mental intellectual constructs presented within a body of text. This can be either the number of times a specific concept appears in the text or the pure existence of the concept. This is completed by understanding the grouping of words prevalent in the body of text. The purpose of doing so can establish connections between different concepts within a body of text (Aguiar, Cury & Zouaq, 2016; Štajner & Hulpus, 2018).

Conceptual Text Analysis also consists of the reader’s interpretation of the text, be it intended or unintended. Examples are irony, innuendo, euphemisms and sarcasm. These build up the reader’s conceptual understanding of the situation, even if it is not physically stated (Aguiar et al., 2016; Štajner & Hulpus, 2018).

Irony exists when two contradicting concepts are placed in one image, sentence or phrase, for example:

> “I feel lonely when I am surrounded by so many people.”

This sentence creates a contradicting idea since the definition of lonely means that you are by yourself. This sentence gives the reader the implication that it is the author's personal feeling towards being surrounded by people (Eldridge, 2017).[2]

Similarly, sarcasm also plays on the contradiction of situations. However, sarcasm contains a bit of wittiness to bring across a humorous insult, although sarcasm can also be considered a fine art of friendship where the use of sarcasm represents the strength of their friendship (Rajadesingan, Zafarani & Liu, 2015). For example, if we know that Chris has terrible looking shoes on today and we say:

"What a world-class choice for shoes today, Chris"

This sentence implies that Chris is wearing terrible shoes. This sarcasm consists of irony in the sense that it contradicts his dress code and is meant to say such a thing. However, depending on the situation, it can also mean that their friendship is close.

Innuendo is something different. It is something that appears to be diplomatic on the surface, but is a subtle way to indicate an insult, humour statement or criticism (Beale, 2018).[3] For example, if we know that Mark gets dragged to prison where the princess currently is and he says:

"I've found a way to get 'extra help' to save the princess, if you know what I mean?"

The ending phrase "if you know what I mean" implies an innuendo. It implies that Mark is not a usual prisoner, but rather getting himself caught in order to carry out some "activity" related to his "extra help".

[2] The arguments from Eldridge were taken into consideration when defining irony.

[3] Arguments in relation to innuendo from Beale were taken into consideration when defining innuendo.

Euphemism is slightly different in the sense that it does not contradict any situation; it is just a polite or soft way of saying things by replacing words or phrases with something with a similar understanding (Rittenburg, Gladney & Stephenson, 2016). For example:

> "After an intense struggle, he passed away."

This sentence indicates that he died, but to soften the impact, "passed away" is used.

All four of the above examples create a particular concept in the reader's mind, yet it is not directly written for the reader to interpret. In communications, it is necessary to understand conceptual analysis in order to understand the true intentions and meanings that the author would have liked to communicate without physically saying it.

The characteristics of Text Mining set out the purposes of accomplishing the aforementioned types of Text Analysis. The next sub-section will discuss the characteristics of Text Mining.

2.6 CHARACTERISTIC OF TEXT MINING

The first general purpose Text Mining program was written by Stone and Hunt (1963) to understand textual data from people's speech. They argued that they could reach psychological conclusions based on the results. One can therefore conclude that two of the characteristics of Text Mining are that the data should originate as textual data and that the output should allow the discovery of knowledge.

Stone and Hunt (1963) also discuss how Text Mining is carried out by a computer program, which provides the capability of a systematic method to accomplish the task. This means that the progress of accomplishment is repeatable in a step-by-step fashion. Thus, Text Analysis can also be defined as being systematic.

Stone and Hunt (1963) furthermore discuss how Text Mining needs to be able to establish trends or patterns from the analysis. This is for the purpose of establishing knowledge by understanding trends. Therefore, another characteristic of Text Analysis is its ability to establish trends and/or patterns.

In summary, Text Mining is systematic, it originates as textual data, and it is able to both establish trends and/or patterns and to discover knowledge. The next sub-section discusses the process of Text Mining so as to understand the role played by Stemming Algorithms and the application of Stemming Algorithms in the Text Mining process.

2.7 THE TEXT MINING PROCESS

The process of Text Mining can be broken down into smaller steps. These steps fulfil the characteristics of Text Mining. Figure 6 represents the simplified Text Mining process.

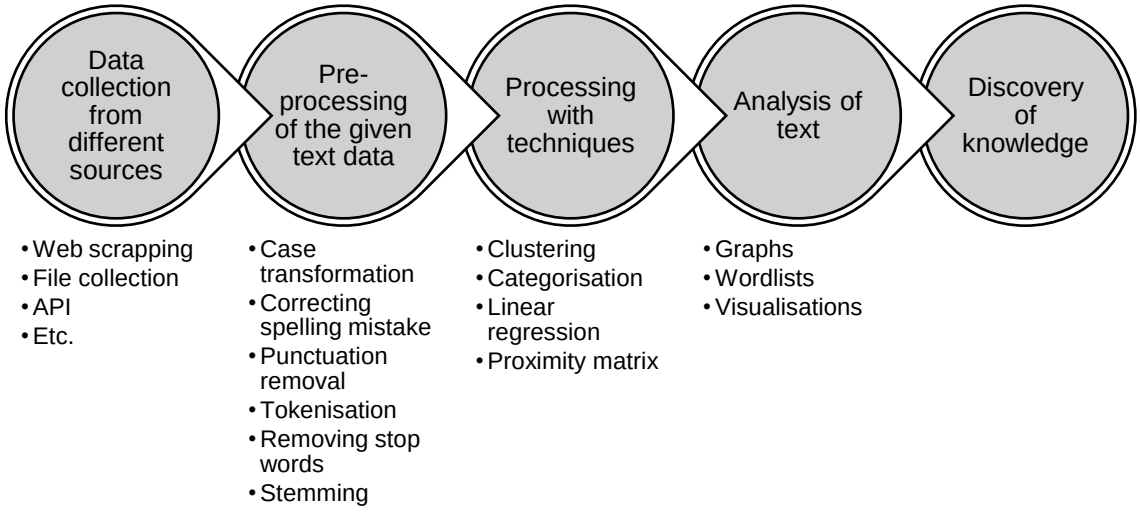


Figure 6: The Text Mining process (Gaikwad et al., 2014)

Figure 6 depicts the process of Text Mining. Having established steps fulfils the “systematic” characteristic of Text Analysis since it is then considered repeatable (Stone & Hunt, 1963). In Figure 6, there are four steps before the discovery of knowledge, starting with data collection and ending with the analysis of the text. The first step is to feed the system with documents that would be the subject of the mining process. The second step would be the pre-processing or cleaning of the given text data. The third step is to apply different Text Mining techniques to the data. The fourth step is to analyse the text after the Text Mining techniques have been applied. The last step is the contribution to knowledge.

The first step requires the loading of raw unstructured data into an application. This step is often called “data collection from different sources”. Unstructured text data originates from any source that provides text in an unstructured format. This is defined as a “document” (Gaikwad et al., 2014). This step contributes to Text Analysis by obtaining the relevant textual data for analysis (Neuendorf, 2016). It fulfils the first characteristic of Text Analysis in that the data originates in a textual format.

Pre-processing ensures that the Text Mining process can group words and their concepts. It acts as a “cleaning” process for documents so that time is not wasted on unsensitised data (Feldman & Sanger, 2007). The steps in the pre-processing stage are as follows: case transformation, correcting spelling mistakes, removing punctuation, tokenisation, removing stop words and stemming (Mathiak & Eckstein, 2004). This phase contributes to natural language processing (Feldman & Sanger, 2007). It furthermore contributes to Text Analysis by preparing the textual data, making sure that the data is in a format that can provide relevant data before using analysis techniques (Neuendorf, 2016).

After completing the pre-processing stage, the third stage can take place. This is the processing stage where the application of techniques such as clustering and categorisation takes place (Gaikwad et al., 2014). Processing is often considered the core of the Text Analysis process where the data is organised in such a way that it is ready for analysis. This stage contributes to Text Analysis by analysing the original textual data (Neuendorf, 2016).

The second last stage in the process is the analysis of the text by representing the Text Mining results from the previous stage graphically (Gaikwad et al., 2014). This stage contributes to Text Analysis by representing the data in such a way that it is understandable to the viewer. This should be done in a way that makes knowledge discovery possible (Neuendorf, 2016). This contributes to the Text Analysis characteristic that is defined as “analysis of trends and/or patterns”.

The last stage in the process is discovering knowledge. After the data has been visualised, knowledge can be extracted from the results. The viewer reads the results from the analysis and draws conclusions from it. The conclusions that are drawn become the knowledge gained from the Text Mining process (Usai, Pironti, Mital & Aouina Mejri, 2018). This step fulfils the Text Analysis characteristic that is defined as “discovery of knowledge”.

The following sections discuss the stages of Text Mining in more detail. They will break down all the sub-components of these stages and describe the functionalities of each stage illustrated in Figure 6.

2.7.1 Data collection in Text Analysis

There are a wide variety of methods for data collection. This study covers examples of web scrapping, file collection and application program interface (API). In the following section, these components of data collection will be described briefly.

Web scrapping is the method of extracting data through standard hypertext transfer protocol (HTML) on the world wide web (www). The www is a network of computers all over the world that can be accessed by personal computers (Marres & Weltevrede, 2013). It uses the internet as a medium for transferring data. HTML is a standard language structure or a set of rules that is used to define the layout of websites. The first step of the process is to search for documents on the www. The next step is to retrieve the web pages by downloading them from the www. The internal links are explored on each website to collect all possible websites up to a predefined level. Once the webpages have been downloaded, the HTML structure-specific data is first deleted, and the textual data that contains the topic-related information is saved (Marres & Weltevrede, 2013).

To search for websites on the internet, search engines are created. Search engines are computer programs or a set of computer instructions created to find websites. An example of a method used by search engines is to correlate a collected list of keywords on all the websites on the internet. The results gathered are ranked based upon relevance to show the most appropriate results to the searcher's first request. Search engines make use of information retrieval methods to structure the results (Croft, Metzler & Strohman, 2010). Information retrieval is discussed later in this chapter.

Another example of data collection is "file collection". File collection or document collection is a way to collect various types of files containing textual data. A file containing data is called a "document". The format in which a document is saved does not matter, as long as the document contains corpora or bodies of text. An extensive collection of bodies of text is considered a corpus, which can then be used for Text Analysis. An example of a document is a simple text file. Methods of file collection are adapted based upon the requirements of analysis and situations of data sources (Janetzko, 2016).

File collection is suitable for Text Analysis as it follows a non-reactive research methodology, which removes the influence of the researcher as a possible parameter in the research. File collection is considered to be non-reactive research, as the researcher is unable to influence the authors or creators of the collected files (Janetzko, 2016).

The last example of data collection is the API. Different technology platforms provide their own APIs, so the different APIs are platform specific. An API is a channel of collecting data through, for example, a web service. Web services are provided from the hosting technology companies. The data that can be gathered from them is limited to the data that is provided by the company itself (Arsanjani, Hailpern, Martin & Tarr 2003). Some examples of these are Google Plus API, Facebook API and Twitter API (Munzert, Rubba, Meißner, & Nyhuis, 2014). These APIs are all examples of social media platforms. The following paragraph will provide more detail pertaining to the why social media provides such a good example of APIs.

Facebook, Twitter and Google Plus are examples of social media platforms. On these platforms, people can create and share interpersonal content to interact socially with one another (Norouzizadeh Dezfouli, Dehghantanha, Eterovic-Soric & Choo, 2016). This creates a social network over the internet. A social network is different groupings of people that interlink with each other to provide each other with companionship. Social networks grow into large societies (Lin, 2017). Within social media and social networking platforms, societies all over the world produce textual data.

Social network analysis is performed to understand the social interaction that is taking place between entities on the social media platform. Since there is a large amount of textual data produced by these platforms, Text Analysis is used as a method of analysis. Text Analysis is embedded in social network analysis to provide results and insight (Scott, 2017). Therefore, social

media and social networking platforms are considered an excellent choice for data collection, which presents itself as a great platform for the application of Stemming Algorithms in Text Analysis. However, data for this research was not collected from social media since it is not relevant to this study's data collection.

2.7.2 Pre-processing in Text Analysis

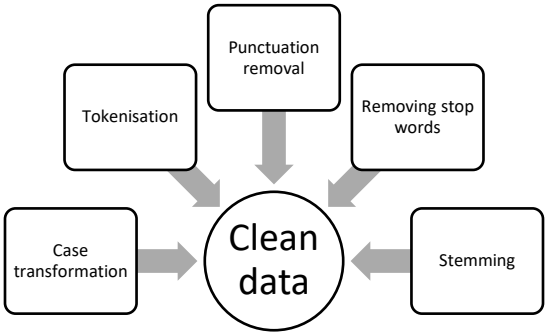


Figure 1: Clean data

Within the pre-processing stage of the Text Mining processes, there is case transformation, spelling correcting, tokenisation, punctuation removal, stop word removal and stemming. These processes will be briefly described in the following section. Figure 7 synthesises the pre-processing stage in Text Mining. This figure shows all the items that need to be completed for clean data to be processed. These items correlate with the pre-processing sub-items in the Text Mining process.

To have clean data, it is essential to have all the words in the same case. All the words that have no meaning need to be removed, and all the words with the same meaning need to be grouped together. Combining words with the same meaning is an essential step in the mining process as the results would be different if this step was not applied.

Case transformation is the process of establishing a standard case throughout the document. The two options are either all uppercase or all lowercase. Uppercase is capital letters, and lowercase would be the opposite (Elragal & Haddara, 2014). The primary purpose of case transformation is to make it possible to group the same word that appears at the beginning of a sentence with the same word that appears elsewhere in the sentence (Wijaya, Erwin, Galinium & Muliady, 2013). For example, if the casing is not transformed to be aligned within the text, the system application process identifies “Him”, “him” and “HIM” as different words where, in fact, they are the same word. These words are found in different scenarios in a text document. For example, “Him” would be found at the beginning of a sentence and “HIM” in a heading or in loudly expressed narrations.

Once case transformation has taken place, the next pre-processing step is fixing spelling mistakes. If the words are not spelt in the same way, they are considered to be completely different words in the process of mining the document. This causes the Text Mining process to yield incorrect results (Akhondi et al., 2014). For example, if one document spelt “insurable products” as “ensurable products”, the Text Mining results would consider the two phrases as two completely different words and the results would be skewed. Therefore, correcting spelling mistakes is considered a vitally important procedure for Text Mining.

Tokenisation is the process of assigning each word in the sentence or document with a relating “token” to represent its part of speech. This process is also known as parts of speech tagging (Elragal & Haddara, 2014).

A part of speech is the syntactic grouping of the functionality each word has within a sentence in the text (Fisicaro & Gauvin, 2018).

Table 1 provides a list of parts of speech and the functionalities of each.

Table 1: Parts of speech functionality

Part of speech	Function	Reference
Noun	The word used in a sentence to identify different entities, which can range from people, animals, ideas, places, objects or events.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Pronoun	The word used in a sentence to remove the repetition of nouns for readability. These words can be I, it, he, she, we or they.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Verb	The word that provides the mental or physical activity; or the state that the entity mentioned in the noun or pronoun is taking.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Adverb	The words used to describe adjectives, verbs and other adverbs. These are grouped into four types: manner, time, place and degree. Manner describes the way in which something is done. Time describes when the action is carried out. Place describes where the action is carried out. Degree describes the intensity of the action.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Adjective	The word that extends the reader's understanding of the entity provided by the noun or pronoun. These words extend the quality and quantity of the noun or pronoun.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Preposition	The words that are used to provide the relation of an entity to the location or time.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Conjunction	The words that are used for humans to understand the flow of the text by providing a joining ability.	(Fisicaro & Gauvin, 2018); (Brown, 1957)
Interjection	The words that convey the expression of emotions.	(Fisicaro & Gauvin, 2018); (Brown, 1957)

In Table 1, there are three columns. The first column is the name of the part of speech; the second column indicates the functions of the respective part of speech and the third column indicates the

references that justify the respective functionality and part of speech. This table shows that each part of speech has different functions associated with them. Since they provide different knowledge outcomes, there is a requirement to tag them to extract the knowledge behind them (Fisicaro & Gauvin, 2018). For example, nouns and pronouns provide knowledge of the object and the possible relationship between objects in a sentence. Verbs provide knowledge of the actions that are taken within the sentence. Adjectives provide the knowledge of more detail on the objects defined by nouns and pronouns. Verbs also play a role in determining emotions if the verbs used are related to emotions, such as “I hate meat”.

Once the tagging of the parts of speech has been completed, punctuations can be removed. The punctuation removal process defines the procedure of removing punctuation that does not contribute to the conceptual meaning of the document. Examples of punctuation to be removed are ? ! # / \ | ; , . (Dredze, Paul, Bergsma & Tran, 2013). The removal of punctuation allows the system to focus on the text that contributes to the final analysis. If the punctuation within the text is not removed, the system application process identifies “him.”, “him?” and “him!” as different words where, in fact, they are the same word. The removal of punctuation needs to take place after the parts of speech have been tagged. This is due to the fact that the full stop represents the end of a sentence and the system would require those punctuations to identify where to place those tags (Johnson, Bretonnel Cohen & Hunter, 2007).

Once the punctuation marks have been removed, the next step would be to remove stop words, otherwise known as filtering stop words. Removing stop words is the process of removing words that do not add conceptual understanding to the message of the document, but rather exist for readability. Examples of such words are “a”, “the”, “or” and “an” (Elragal & Haddara, 2014). These words are derived from prepositions, conjunctions and interjections as parts of speech that emphasise the importance of tagging parts of speech. These only enable us to read the document

and do not contribute to the underlying knowledge of the meaning in the text (Pimpalshende & Mahajan, 2017; Silva & Ribeiro, 2003).

The final task in the pre-processing stage of Text Analysis is stemming. As discussed in Chapter 1, the process of stemming defines the procedures that are undertaken to group the conceptual understanding of words into the same word (Elragal & Haddara, 2014). The result of stemming the words from a body of text is entirely different from the results of not stemming them. An example is if we find the words “booking”, “booked”, “rebook” and “unbookable” in a document. These are completely different words that contribute to the same conceptual understanding of the document. Therefore, these four words need to be grouped together. To group these concepts together, the common trend of these words needs to be established. The commonality of these words is “book”. Within all those words, “book” can be found. This form is known as the “stem” or the “root” of the word, where the rest of the word is considered the suffix and prefix. The stem or root word would be “book” since it is the shortest word within all of them (Moral et al., 2014). This can be a complex task as different languages have different ways of representing their prefixes and suffixes, as well as syntactic rules (Ismailov, Jalil, Abdullah & Rahim, 2016). Chapter 3 provides a further examination of the different methods that have been established to perform stemming for different languages. This is the method on which this research study mainly focuses.

2.7.3 Processing with techniques of Text Analysis

This stage of the Text Mining process is where the main processing happens. There are different techniques to perform Text Analysis. However, for the purposes of this research, only information extraction, clustering, categorisation, linear regression and proximity matrix are discussed.

Information extraction is the process of establishing a structure within unstructured or semi-structured text or concepts of text. Information extraction involves natural language processing to

extract knowledge. The results of information extraction allow a system to build a structure around the content provided within the document to create a suitable structure for information retrieval (Müller, Kenny & Sternberg, 2004). This structure can range from databases, formatted text or diagrams. Typical steps within information extraction can include named entity recognition, co-reference resolution, relationship extraction, and language and vocabulary analysis (Müller et al., 2004; Rindflesch, Tanabe, Weinstein & Hunter, 1999).

Named entity recognition is the establishment of real-life objects from unstructured text such as humans, places and currencies. After these have been identified, they are categorised into groups to which these entities belong (Van Rijsbergen, 1977).

Co-reference resolution is the ability of the computer to understand and associate the appropriate pronouns, proper nouns and nouns (Lee, Surdeanu & Jurafsky, 2017). The human brain would be able to process a sentence such as the following:

> "The coconut is pink, and it has been discarded for being pink."

As humans, we can understand that "it" refers to "the coconut". However, the computer cannot make a direct link. The co-reference resolution sets out to resolve this problem.

Relationship extraction works more closely with pronouns where the establishment of the connection between entities is created (Lee et al., 2017). For instance, in the sentence "Tom just climbed Mount Everest", the entities "Tom" (human) and "Mount Everest" (place) are linked together.

Information retrieval is like a search engine where relevant information is extracted from a system based on the keywords entered. A keyword is a word that has significant value within a piece of

text and can typically be found in the title of the document. This combination of keywords is called a query. This system can be created based upon an indexing method or a database (Hariharan, Hore, Li & Mehrotra, 2007). Usually, the indexing database is created through the information extraction techniques mentioned above (Kandogan et al., 2006). The query is compared to words within the documents relevant to their retrieval. Based on a set of rules and criteria, the different sets of documents are returned. Not all results returned are entirely in line with what the searcher was looking for or according to the query. Therefore, a ranking system must be put in place to establish the level of relevance (Croft et al., 2010). There is considerable debate on the association of information retrieval with Text Analysis, since it is practised as more of a document retrieval tool than a Text Analysis tool. However, information retrieval is still a method of retrieving information from different documents, and creates an overlap with Text Analysis.

Text Analysis and information retrieval can be seen as two different fields of study. However, the two fields of study overlap. There is some knowledge of information retrieval that is concurrent with Text Analysis and vice versa (Croft et al., 2010). The following paragraphs will discuss examples where Text Analysis practices and information retrieval practices overlap.

Information retrieval often uses Text Analysis practices to know how to save documents for retrieval later. Since we know that Text Analysis can create relationships between documents, it is actually possible to sort the documents and allow a proper classification or categorisation of the documents for later retrieval. On the contrary, information retrieval can be seen as e a standard method of data collection for further Text Analysis practices (Croft et al., 2010). Due to the fact that Text Analysis and information retrieval overlap in practice, some of the techniques are shared across the two fields of study. For example, vectors and matrices are used in both fields of study (Bhatia, 2013).

A vector is a measure of direction and magnitude, where one object is in a relative space compared to another. In the case of Text Analysis, the vectors would work on objects such as words, phrases, sentences and documents (Bhatia, 2013).

A matrix is a statistical method of calculating values placed in a rectangular array structure to find the pattern between objects. In this case, the objects could represent words, phrases, entities or documents (Bhatia, 2013). In Text Analysis, the technique “proximity matrices” uses the underlying “matrix” technique. Proximity matrices will be further discussed in Chapter 2.7.4: Analysis of Text Analysis.

Since the focus of this research is not on information retrieval, the discussion will move away from information retrieval. Clustering is used to find groups or outcomes within a document that are not defined before the Text Mining process begins. The results of the clustering typically end with a number of clusters that each contain a portion of the original document. The data within each portion should be similar, which is why it is considered to be within one cluster. If the data compared within each cluster is very similar, and the data compared across clusters is not similar, then the result of the clustering technique is considered good. The difference between clustering and categorisation is that clustering does not make use of predefined outcomes (Gaikwad et al., 2014).

Some of the more common methods or algorithms of performing clustering are K-means clustering and K-modes clustering (Gaikwad et al., 2014).

The term “K” within K-means and K-modes indicates the number of clusters that requires a result. Both these techniques refer to centroids. Centroids indicate the centre point of the cluster. Both these methods iterate the centroids closer and closer to their respective cluster centres. With K-means clustering, the distance between each data point and the centroids is calculated. The data

points that are closest to the centroid are then assigned to that centroid. The centroid then moves closer to all the data points that were assigned to it by taking the average distance of each data point within that cluster. This then iterates a few times (as required) until it is stable. K-modes follow the same outlined steps. The only difference is that when K-modes recalculate the centre point for the centroids, it does not use the averages, but rather the modes or most common data points (Huang, 1998).

Another technique of Text Mining is categorisation. Categorisation is the process of assigning one or more categories or outcomes to an unstructured text document. Categorisation techniques are powered by the input and output of processes. Categorisation groupings are defined before the process of Text Mining begins. This is done to see whether the outcome of the document falls into any of the defined categories. Therefore, the goal of categorisation is to train the process to fit the document into the predefined groupings or to automatically classify the document into a non-categorised group (Gaikwad et al., 2014).

Some of the more well-known methods or algorithms of performing categorisation are the Naïve Bayesian classifier, the K-nearest neighbour classifier (KNN) and the decision tree (Gaikwad et al., 2014). The following paragraphs will discuss these examples in further detail.

The Naïve Bayesian classifier is a classifier that is based on probability. Again, the categories are predefined before the process begins. The probability of the words on hand is then calculated against every category's class. The class indicates an existing grouping of characteristics for its respective category. The category with the highest probability is then selected. The probability is calculated with a particular formula (Karthika & Sairam, 2015). Although it is possible to go deeper into the formula for the probability, it is only mentioned briefly in order to stay within the scope of the research.

KNN is a technique that can be used for either classification or regression. To begin the KNN technique, the dataset should have two attributes that allow it to be placed on a scatter graph. This method calculates the distance between the data point and the next closest data point. As a result, with the calculation of each data point, there is a division of space on the graph. The division between the points can indicate which point should be categorised when the data points are placed on a graph at random. Once the categories have been established, a boundary line can be drawn between the two sets of categories, which is a non-linear line. However, to decrease the number of errors produced, it can be done with more than one neighbouring data point (Nowak, Nowicki, Woźniak, & Napoli, 2015). Although it is possible to go deeper into the calculations, it is only mentioned briefly in order to stay within the scope of the research.

The last example of categorisation is that of the decision tree. A decision tree uses a tree structure to define categories based on a sequence of attribute values. We know that the categories need to be defined beforehand. The groups are built based upon a divide and concur (approval) method. The division of the attributes is derived from the range of established attributes. This means that if three possible variations of values can be found within an attribute, it is divided into three subgroups of data relative to the original attribute. If the subgroups cannot provide the final categorical answer, then the attribute continues to be divided for the subgroup until the values come to a halt on the categorical answers. Following the sequence of the grouped attributes, one would be able to derive a sequence of attributes that lead to a certain category of attributes (Song & Ying, 2015). Although it is possible to go deeper into the calculations, it is only mentioned briefly in order to stay within the scope of the research.

The result of both clustering and categorisation returns groups of words. Thus, from these groups of words, it is possible to run both clustering and classification techniques on the documents

themselves to provide groups of documents (Priandini, Zaman & Purwanti, 2017; Violos et al., 2014).

Linear regression, as another example of a processing technique for Text Analysis, is a linear approach to modelling the relationship between random responses of the classification of text over two dimensions. The two dimensions are the dependent variable and the independent variable. The independent variable is the number of documents in which different words occur or the time series of analysis. The dependent variable is the number of times the word occurs in total. This shows the trend of the words, as well as the correlation between each word and the trend (Darlington & Hayes, 2016; Westgate, Barton, Pierson & Lindenmayer, 2015).

Lastly, proximity matrix uses the distance between two words to determine the relationship between the words. This provides a specific strength in the relationship between the two words. The different words are placed in a table that compares each word with every other word in the given body of text. The stronger the relationship, the higher the chance that the two words have some sort of meaning when they are used together. These distances are calculated through a standard formula based on the total number of times the word appears in the document (Graepel, Herbrich, Bollmann-Sdorra & Obermayer, 1999).

This study covered classification, categorisation, linear regression and proximity matrix as examples of Text Analysis techniques. However, there are other techniques, such as state-of-the-art and rule-based sentiment analysis, which are beyond the scope of the study. With the above-mentioned techniques of Text Analysis, examples of where Stemming Algorithms are indirectly applied have been discussed. The next section discusses the analysis of Text Analysis once these techniques have been processed.

Once the different techniques have been applied, the data needs to be presented in a format that the reader can analyse and interpret. This leads to the next sub-section, analysis of Text Analysis, in which the different methods used to present and analyse the results are discussed.

2.7.4 Analysis of Text Analysis

In Figure 6, three examples of the analysis of Text Analysis are provided: graphs, word lists and the proximity matrix. Each has its own way of presenting the data to be analysed.

Graphs are represented in a diagram format, which usually compares the relationship between variables. The variables are usually on a different axis. However, this may differ based on the type of graph. Graphs have been found to be suitable for quantitative data since the primary function of a graph is to compare variables (Harris, 2000).

Another method of analysis is the word list. A word list is a table containing three main items. The word or term itself, the total occurrence of the word and the total occurrence in documents. The total word or term occurrence displays the number of times a particular word or term appears throughout all documents. Total document occurrence shows in how many articles this word or term appears (Ertek, Tapucu & Arin, 2013). The higher the occurrence of a word or term in a document, the more prevalent or frequent the concept is within the document (Nielsen, 2011). Methods have been developed to further analyse this word list. An example would be the proximity matrix.

Word lists can be returned in different formats with a different number of variables of comparison, for example, word count or the number of documents in which each word appears. If the word list that is returned has more than one variable to compare it with, then different statistical methods can be used to calculate the results, such as linear regression. However, if there is only one

variable, such as only a word count, then a proximity matrix will be required to analyse it (Van Deventer, 2014).

A proximity matrix is the result of the proximity matrix technique discussed in Chapter 2.7.4. Figure 8 represents an example proximity matrix with random data. From the results, one can see that the weighting of the values is illustrated in different colours. Dark red represents a robust negative distance, while dark blue represents a strong positive correlation. The pattern of the colours represents the trend that is set from the results of the analysis.

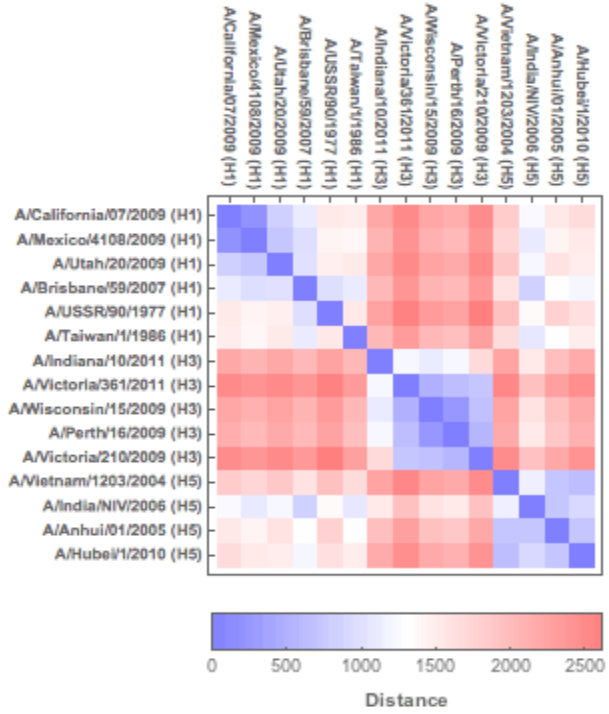


Figure 8: Example proximity matrix

Network diagrams show a pool of connections between different words that establish the relationship or network these words have with each other (Kucher & Kerren, 2015). An example is given in the background of Chapter 1 with the word “create”.

2.8 CHALLENGES OF CURRENT TEXT MINING METHODS

Even though the establishment of Text Analysis has been accomplished over many years, there are still problems faced with Text Analysis. Some of the significant problems that will be discussed later come from the conversion of human-readable text into something that a computer can process. Examples are problems that are established through the transformation from the linguistics sphere into the Text Analysis sphere, which results in a loss of context (Dohare, Karnick & Gupta, 2017).

When Text Mining is applied to different languages, it becomes a challenge to compute the languages' different structures. However, to focus on the research, only one language will be discussed. To fully discuss the problems associated with Text Analysis, the different types of Text Analysis will be mentioned, and the problems associated with each will be discussed.

From a syntactical analysis perspective, the challenges that are established are related to the spelling structures and the sentence structures.

The spelling structures of different languages are different. This poses a problem in Text Analysis as it creates difficulties in allowing the same software to understand the different texts (Akasereh, 2015). This can be seen in the difference between Afrikaans and English. The plural suffix in Afrikaans is, for example, "-e", "-ers" or "-etjies", and the plural suffix in English is, for example, "s" or "es" (Stell, 2015). This is only a small example compared to the large number of languages that exist today. This means that, for every language, a unique way to handle their differences based on their spelling structure is required. Even within English itself there are exceptions to the spelling structure based on the origin of the word.

Some languages have the verb at the beginning of the sentence, and others have it at the end of the sentence. This difference in the placement of verbs results in different meanings. Before one can engage in tagging the parts of speech, one would need to understand the sentence structure of a particular language. This again affects the implementation and processes of Text Analysis (Akasereh, 2015).

Therefore, to apply Text Analysis to different languages, one would need different algorithms or rules for each language. Exceptions in a language require elaborate algorithms. This requires a lot of design and creation to establish methods to accomplish the same set of Text Mining tasks for different languages (Akasereh, 2015).

From a semantic perspective, as mentioned in the discussion above, sentiment analysis is the analysis of meaning in words. The sentiment of a word can be changed by changing the structure of a sentence (Thelwall, 2017). This also ties in with conceptual analysis. Consider the following sentence:

> "I love shoes."

The common understanding of the sentence would indicate that "love" is a positive sentiment or has a positive meaning. However, with the introduction of sarcasm, the word "love" becomes a negative sentiment or will have a negative meaning.

From a pragmatic perspective, the meaning of words can change based on the context of the current situation within the text. Additionally, the meaning of the words can change based on the knowledge of the reader (Widdowson, 2008).

From the perspective of context, if the document is about diseases, and one sees the sentence: “He is going green”, there is a greater likelihood that the sentence means: “He is going to be sick” rather than: “He is going to start recycling”. This presents a challenge for computational Text Analysis, since the computer, somehow, needs to know which words provide the pragmatic connotations and which ones do not (Widdowson, 2008).

When considering the reader’s knowledge in relation to the previous example, the following can be said: If the reader has no mental association between the word “green” and becoming “sick”, the sentence will be misinterpreted. If the reader has the idea that “going purple” is, more appropriately, the way to express becoming sick, then there is a conflict in communication (Widdowson, 2008).

From a conceptual perspective, the difficulty lies in cases of irony, sarcasm, innuendo and euphemism (Widdowson, 2008).

In the case of irony, the sentence requires context. It is possible for a computer to pick up irony in phrases and sentences since the context is given. However, if irony is established through real-life scenarios outside the context of the document, a computer will have difficulty finding that association. An example of such a case is the following:

> “The heaven of drugs.”

With this sentence, the irony comes in when drugs are associated with the opposite of heaven. However, from the given context, there is no way of distinguishing the irony.

Sarcasm is also not always clear, making it difficult for computers to pick up. Sarcasm requires the context of the author. Given the following example, if the author thinks that my pink shoes are terrible, he sarcastically says:

> "I just love your pink shoes so much!"

Without understanding the author's perspective, it becomes difficult to pick up the sarcasm in that sentence. It is already difficult for humans to pick up that the author has the opposite intention; how would it be possible for a computer to pick up the author's intentions? Currently, the only way that a computer can know that something is sarcastic is if the Text Analysis application is provided with the context beforehand to allow it to understand the author's personal perspective (Widdowson, 2008).

In the case of innuendo, with the example given in this chapter, it is difficult for computers to identify that "if you know what I mean?" indicates that the author of that statement is about to do something untoward, since this is a conceptual communication of ideas and no direct text is provided to state the actions the author intends to take. In fact, not even humans can claim knowledge of any actions that the author intends to take since the author did not express the action himself in a direct manner. It becomes difficult for computers to even pick up on these innuendos, let alone establish their meaning.

In the case of a euphemism, with the example given in this chapter, it is difficult to establish the fact that words are written in a manner that is easier to accept, unless enough context is given and the direct actions are mentioned. If the text states "passed away", the computer will process it as it is and will not convert it to "died". This becomes a difficulty as the correlation between concepts in a textual document requires real-life associations.

Many attempts have been made to develop technology to learn, interpret and analyse sarcasm, irony, innuendo and euphemism. However, Minhas (2016) argues that not even computer algorithms with learning ability can solve the problem. Since the text is disassociated from the real-life conditions of the reader and the author, the technology has the limited capability to include those in its parameters (Minhas, 2016).

2.9 SUMMARY AND CONCLUSION

In conclusion, this chapter gives a basic introduction to the Text Mining process and the different steps that are taken to clean the data in the pre-processing phase. It covered the aim and scope of the research, Text Processing, types of Text Analysis, the characteristics of Text Analysis, the Text Analysis and Text Mining process, and the challenges experienced. It further discussed the need for Stemming Algorithms.

From the discussion, the four types of Text Analysis that were identified are syntactic, semantic, pragmatic and conceptual Text Analysis. Each of these, together with the difficulty of Text Analysis to process them, is discussed in the challenges section.

The four characteristics of Text Analysis are that it is systematic, that it originates from textual data, that it can be used to establish trends and/or patterns, and that it is useful in the discovery of knowledge.

There are five main steps in the process of Text Mining: obtaining the text document from different sources, pre-processing the given text, applying Text Mining techniques, analysing the text and discovering knowledge.

In obtaining the text document from different sources, it was discussed that any raw document can be used for processing.

In the pre-processing of a given text, it was discussed that a number of tasks need to be completed to clean the data. These comprise case transformation, correcting spelling mistakes, tokenisation, punctuation removal and stemming. Case transformation relates to changing all cases to either uppercase or lowercase. Correcting spelling mistakes means ensuring that no word is misspelt, unless this is intentional. Tokenisation relates to assigning parts of speech tags to each word in the document. Punctuation removal means removing all the punctuation marks that do not contribute to the conceptual understanding of the document. Stemming is the process of removing the suffix and prefix of a word to get to the root word so that all the words with the same concept can be group together.

Two Text Mining techniques can be applied: clustering or categorisation. Categorisation has predefined goals or groups into which the document needs to fit. Clustering groups the document into clusters on the fly. Commonly known methods for categorisation are the Naïve Bayesian classifier, the nearest neighbour classifier, the decision tree, and support vector machines. Commonly known methods for clustering are K-means clustering and K-mode clustering.

The analysis of the text is the process of analysing the results generated from the text mine. This is generally done by giving a graphical representation of the results.

Discovery of knowledge relates to the results that are gathered from the analysis of the text.

From the individual ideas given above, one can see that the main focus of the study is the pre-processing procedures of Text Mining. This chapter reviewed different methods of mining the text and the standard methods that are used for each method.

The following chapter discusses the characteristics of Stemming Algorithms and the existing Stemming Algorithms.

Chapter Map 3: Stemming Algorithms

CHAPTER 3:
STEMMING ALGORITHMS

3.1 INTRODUCTION

This chapter discusses the different Stemming Algorithms that exist. For the purposes of the research, only the Stemming Algorithms that are found in research articles accessible to the University of Pretoria are used. The reason for this chapter is to define the algorithms, discuss how they work and compare them. Understanding Stemming Algorithms makes it is possible to analyse them.

3.2 AIM AND SCOPE OF THE CHAPTER

The aim of this chapter is to discuss the various Stemming Algorithms that have been established and to gain background knowledge on them. It is important to understand this in order to know how to deal with the change in language structure in the case of a Stemming Algorithm.

3.3 CHARACTERISTICS OF STEMMING ALGORITHMS

As mentioned in Chapter 1, Stemming Algorithms form part of Text Analysis and Text Mining. Stemming Algorithms group words of the same or similar basic meaning together. Examples of words that are similar can be “definable”, “definability” and “definition”. Another purpose of Stemming Algorithms is to improve the Text Mining process by indexing words as one word instead of different words. Indexing words from collected sources will speed up the time required to finish the Text Mining process. It also reduces the variety of words required to retrieve results from Text Mining as the words have been organised into the same groups (Moral et al., 2014).

“Stemming Algorithms” can be separated into two components: “stemming” and “algorithm”. Therefore, characteristics of Stemming Algorithms can be derived by determining the characteristics of stemming and the characteristics of algorithms.

Moral et al. (2014) argue that the characteristics of stemming are determining the stem of a word and removing the prefix and/or suffix.

Parekh, Saksena, Ringshia and Chaudhari (2017) argue that the characteristics of algorithms are precision, uniqueness, finiteness, input, output and generality. Precision is the clarity of the defined steps in the algorithm. Uniqueness refers to the unique actions carried out within each step of the algorithm where only the input will change the resulting outcome of the step. Finiteness refers to the algorithm coming to an end after a certain number of steps. Input and output refer to the algorithms taking in input and output respectively. Generality refers to the capability of the algorithm to perform on various sets of input.

Based on the arguments of Moral et al. (2014) and Parekh et al. (2017), one can assume that the characteristics of Stemming Algorithms are precision, uniqueness, finiteness, input, output, generality, determining the stem of a word and removing the suffix. For the purpose of this study, we will focus on the aforementioned. This does not imply an exhaustive list in analysis. Now that we have identified the characterstics of Stemming Algorithms, let us consider their classification.

3.4 STEMMING ALGORITHMS

Vijayarani et al. (2015) argue that there are three types of Stemming Algorithms: truncating, statistical and mixed Stemming Algorithms. From a global perspective, truncating takes on a rule-based approach of stemming where a corresponding suffix rule set to the usage instructions is provided. Statistical approaches do not provide any suffix or rules. Instead, they

base their approach on using statistical formulae over the whole body of text to provide a conjunction set of letters, from word to word, to determine the underlying word's stem.[4] The mixed method is a combination of the first two methods (Vijayarani et al., 2015). Figure 9 depicts the classification of Stemming Algorithms into the three types of Stemming Algorithms mentioned above.

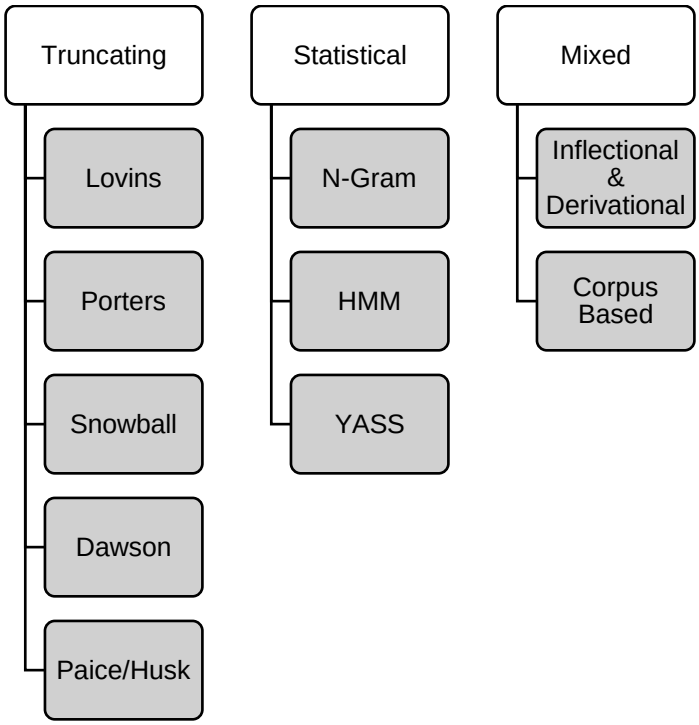


Figure 9: Classification of Stemming Algorithms (Vijayarani et al., 2015)

In Figure 9, there are three main categories of Stemming Algorithms. These are truncating, statistical and mixed Stemming Algorithms. The truncating methods have five Stemming Algorithms within them: Lovins, Porter's, Snowball, Dawson and Paice/Husk. The statistical category of Stemming Algorithms have three types of Stemming Algorithms: n-grams, the

[4] Porter (1980) describes the stem of a word as the combination of letters in a word that remains consistent if the suffix of the word were to be removed.

Hidden Markov Model (HMM) and Yet Another Suffix Striper (YASS). Under the mixed category of Stemming Algorithms, there are inflectional and derivational, as well as corpus-based Stemming Algorithms. The next sections discuss Stemming Algorithms in more detail.

3.4.1 Porter's stemmer

Porter's Stemming Algorithm is one of the oldest Stemming Algorithms in existence. It was developed based on rules and predefined prefixes and suffixes (Moral et al., 2014). It has 62 rules and 51 suffixes (Ismailov et al., 2016). The language it supports is English (Moral et al., 2014).

Porter's algorithm defines five steps. The first step deals with inflectional suffixes. The next three steps deal with derivational suffixes, and the last step deals with putting the word together (Moral et al., 2014). The following paragraphs will discuss the abovementioned algorithm in greater detail.

Each of the five steps in Porter's algorithm contain a set combination of letters as an indication of the data to be stemmed.

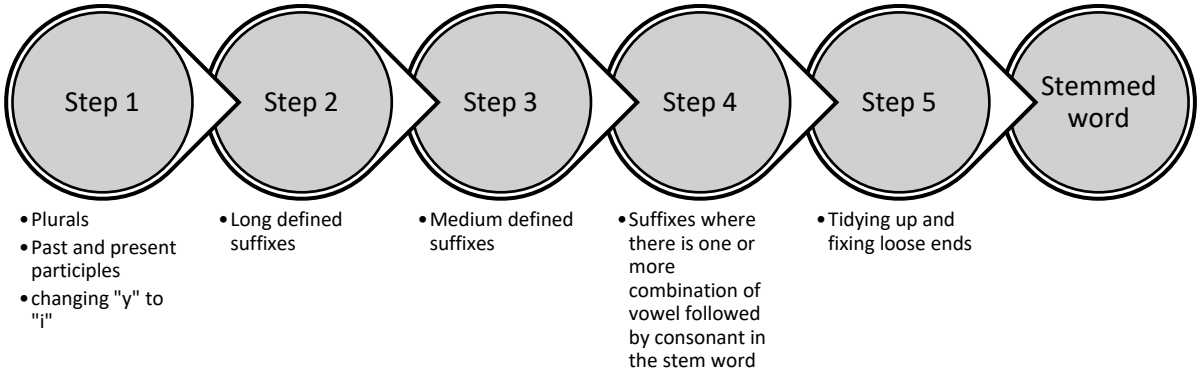


Figure 10: Steps in Porter's stemmer[5]

According to Figure 10, five steps are taken before one reaches the stemmed word. The first step comprises three parts. The first is to clean out plural suffixes such as "s" and "ies". The second is to remove present and past participles such as "ing" and "ed". The third is to change the suffix "y" to "I". The second step deals with larger suffixes. Although it is possible to dig deeper into each suffix, only examples as given for the purposes of this study as the overview interest is prioritised. Examples of such rules are "ational", which needs to transform to "ate" and "isation", which needs to change to "ise". The third step deals with medium-sized suffixes such as "ative", which needs to be removed, and "ical", which needs to change to "ic". The fourth step deals with suffixes that are attached to stem words with one or more combinations of a vowel, followed by a consonant. The fifth step cleans up the rest of the words and ties up loose ends such as words still ending with "ll", such as the double "l" in "controlling" and the remaining suffix "e" that should be removed (Porter, 1980).

[5] Diagram adapted from the original study on suffix stripping of Porter (1980).

Before discussing the advantages and/or disadvantages of Porter's algorithm, there are mainly two error margins associated with stemming accuracy. These are over-stemming and under-stemming. It has already been discussed that Porter (1980) considers stemming to be the action of removing the suffix of a word to retrieve the root of the word. Gurusamy and Nandhini (2017) discuss how over-stemming and under-stemming occur. Over-stemming is when the algorithm returns a stem that is too short. Under-stemming is when the algorithm returns a stem that is too long and still contains quite a few letters of the suffix. The advantages and disadvantages that follow in the rest of this chapter make use of these two terms when they are indeed a disadvantage.

The advantage of this Stemming Algorithm is that it produces quite clean results compared to other rule-based stemmers. Therefore, it also has a lower error margin. It is considered to be a lightweight stemmer as it has less rule sets to store compared to the algorithms of Lovins and Dawson (Ismailov et al., 2016).

The disadvantage of this Stemming Algorithm is that there are a lot of non-realistic stemmed words, due to over-stemming. Another disadvantage is that the process can take longer as it has many steps compared to other stemmers. It also fails to handle some exceptions in English (Jivani, 2011). It is limited to the implemented language, which is English (Porter, 1980). Based on the steps gathered from Porter (1980), the algorithm can only work on suffixes and not on prefixes.

3.4.2 Snowball stemmer

The snowball Stemming Algorithm is an extension of Porter's Stemming Algorithm. The snowball algorithm is considered to be a framework of Porter's algorithm. This framework

allowed other programmers to develop their own framework based on this algorithm for their own languages (Jivani, 2011).

The steps in this Stemming Algorithm are the same five steps as in Porter's algorithm. The differences are that the code behind it is optimised and altered to accommodate different languages. Porter created a framework to allow an external contribution to the Stemming Algorithm rules (Vijayarani et al., 2015).

The advantage of this Stemming Algorithm is that it caters for different languages by allowing the community and the public to contribute to its development (Jivani, 2011). The snowball algorithm supports Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Portuguese, Romanian, Russian, Spanish and Swedish (Moral et al., 2014).

The disadvantage of this Stemming Algorithm is that over-stemming and under-stemming still occur since it is a rule-based Stemming Algorithm, and the algorithm follows the same steps as in the algorithm of Porter (1980) (Jivani, 2011). Additionally, even though it caters for different languages, every time a new language needs to be catered for, the rules have to be manually defined. Lastly, similar to the algorithm of Porter (1980), it only works on suffixes.

3.4.3 Lovins's stemmer

The Stemming Algorithm of Lovins (1968) was the first Stemming Algorithm to be established and to be used by the public. Moral et al. (2014) discusses how the stemmer of Lovins (1968) has 294 word endings, 29 word conditions and 35 transformation rules. All these word endings, word conditions and transformation rules were defined by Lovins (1968).

This algorithm comprises two steps. The first step is to remove the longest possible endings from the words. Once the endings have been removed, the next step is to transform the

endings based on the 35 rules that have been defined. This is also a variant depending on how much of an ending is removed (Moral et al., 2014).

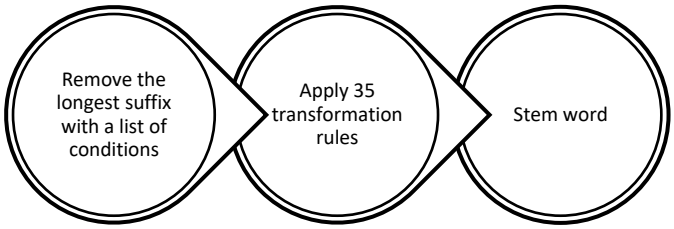


Figure 11: Steps in Lovins's Stemming Algorithm[6]

According to Figure 11, this algorithm has two main steps. This figure has been simplified to bring across the main ideas that are required. Lovins uses a table of rules to correlate both steps. The first step looks for the longest possible suffix that it can correlate. That is removed from the word. The rules set out in the first step are all context-specific rules. The second step involves the 35 rules for the transformation of the ending. This step is actioned regardless of the outcome of the first step (Lovins, 1968).

The advantage of this Stemming Algorithm is that there are only two steps in the process, so the processing time required for each word is very short. It also handles a lot of exceptions to the language structure that Porter's algorithm was unable to do, such as the double "t" in "getting" (Ismailov et al., 2016).

[6] The diagram is adapted from the explanation in the study of Lovins (1968).

The disadvantage of this Stemming Algorithm is that it is intensely data consuming as it has to store all the different rules in tables (Ismailov et al., 2016). Ismailov et al. (2016) discuss another disadvantage of Lovins's stemmer, which is that it leads to over-stemming. This can be seen from the results of using Lovins's stemmer. Furthermore, it is a rule-based stemmer where the suffixes are stored in a table. If the table is no longer maintained, the adaption of new words into the language are not stemmed (Ismailov et al., 2016). It is also limited to the implemented language, which is English. Further extensions of the different languages would require a complete alteration of the rules (Lovins, 1968). Additionally, Lovins (1968) mentions how the algorithm only works on word endings, which means that it does not cater for prefixes.

3.4.4 Dawson's stemmer

Dawson's Stemming Algorithm is an add-on to Lovins's Stemming Algorithm. This add-on, however, is not open to new languages, but is rather a collection of more suffixes. There are 1 200 suffixes in Dawson's Stemming Algorithm (Moral et al., 2014). The developers also improved on this algorithm to allow it to be accessed a lot quicker than Lovins's algorithm. The downside to this algorithm is the complexity of its design and the lack of reconfiguration (Moral et al., 2014).

This Stemming Algorithm has the same steps as Lovins's algorithm. The difference between the two is that the data is stored in a branched character tree method that assists with processing time (Dawson, 1974; Ismailov et al., 2016). The steps are therefore not presented again.

The advantage of this Stemming Algorithm is that it covers more suffixes compared to Lovins's algorithm. It is also a lot quicker to process compared to Lovins's algorithm, since the data is stored in a tree format that speeds up the search rate of rules (Ismailov et al., 2016).

The disadvantage of this Stemming Algorithm is its complexity, which eliminates the ability for it to be reused in other scenarios or languages (Ismailov et al., 2016). The rules provided by Dawson (1974) were suffixes of the English language. Further extensions of the different languages would require a complete alteration of the rules. Additionally, since the steps in the algorithm are adopted from Lovins (1968), Dawson's stemmer would also only cater for suffixes and not prefixes.

3.4.5 The Paice/Husk stemmer

The Paice or Husk Stemming Algorithm, or the Lancaster stemmer, is an iterative Stemming Algorithm with 120 rules. These rules are defined in a table and based on the last letter of the suffix. It then determines if it is a replace or a remove motive, which iterates until it terminates (Moral et al., 2014).

Moral et al. (2014) identified a few conditions that can cause the algorithm to terminate:

- The first letter of the word is a vowel.
- The suffix in the word has no correlating rule in the table.
- There are two letters left in the word.
- There are three letters left in the word and the word begins with a consonant.

Moral et al. (2014) argue that the Paice/Husk stemmer is a very powerful algorithm. However, Moral et al. (2014) identified a high chance of this algorithm over-stemming words.

Figure 12 provides a flow chart that displays the steps that are taken in the Paice/Husk algorithm. There are primarily four decisions and two processes. The algorithm begins by checking whether the last letter of the word is a vowel. If it is, the algorithm returns the

existing/remaining word as the stem. If the last letter of the word is not a vowel, it checks to see if there is a rule in the rule tables for this word. If there is no existing rule for this word, the existing/remaining word is returned as the stem. If there is a rule for the word, it checks two more criteria before applying the rule. The first criterion is that if the word only has two letters left, it returns that as the stem. The second criterion is that if the word has three letters left and the last letter is a consonant, that word is returned as the stem. Once both of those conditions have been checked, the rules found in the tables are applied to the word. After the application of these rules, the remaining word is iterated through the process until the stem is found.

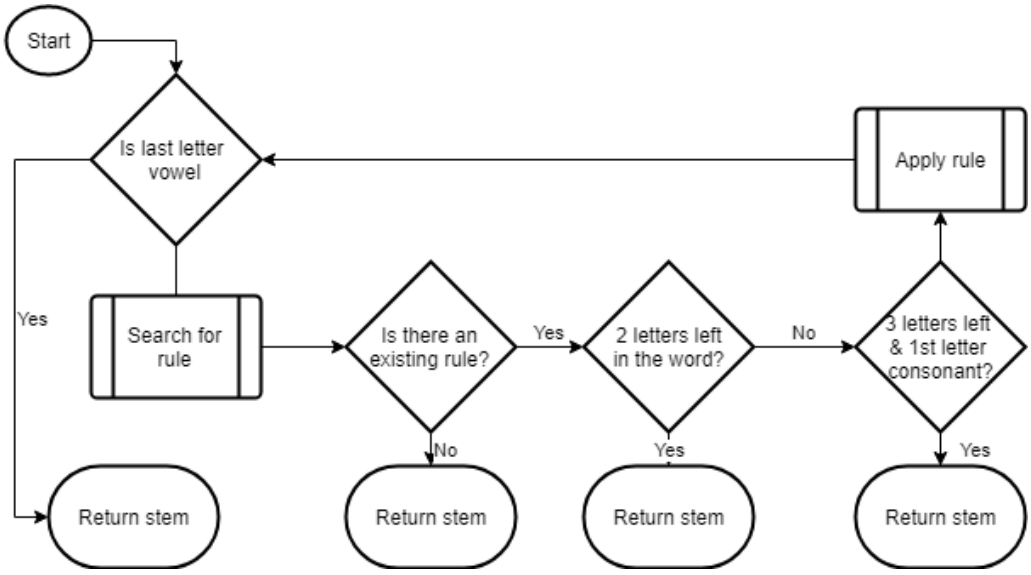


Figure 12: The Paice/Husk algorithm flow chart[7]

The advantage of this Stemming Algorithm is the simplicity of its logical steps, where each step involves removing and replacing suffixes (Ismailov et al., 2016). It is also easier to set

[7] The diagram is adopted and adapted from Moral et al. (2014) by converting the descriptions into a diagram.

up a new rule in the algorithm for different languages when compared to Porter's algorithm (Moral et al., 2014).

The disadvantage of this Stemming Algorithm is that the word can easily be over-stemmed and under-stemmed (Moral et al., 2014). Even though, in the previous paragraph, it was discussed that alteration to the rule would be easier than in the algorithm of Porter (1980), it still means that an alteration is required, which is regarded as a disadvantage because, in order to adapt the algorithm to another language, it would need moderation and experimentation to make sure that the alterations are accurate. The moderation and experimentation will take time. Another disadvantage would be that the algorithm itself is heavy, and takes lots of processing power to run (Sirsat, Chavan & Mahalle, 2013). Additionally, it can only work with prefixes (Ismailov et al., 2016).

3.4.6 The n-gram stemmer

The n-gram is a combination or collection of letters or words that have meaning. The term "n" represents the number of letters or words that are required to provide such a possible meaning. The combination of these letters or words generally means that there is an associated relationship between the "n" number of words. For example, this research paper constantly talks about "Text Analysis". With the help of n-grams, it would be possible to identify the collection term and not treat each word separately (Cavnar & Trenkle, 1994).

N-gram stemmers work on bi-grams or tri-grams (Rani, Ramesh, Anusha & Sathiaseelan, 2015). A bi-gram is the grouping of consecutive letters that come in pairs, and a tri-gram is the grouping of consecutive letters that come in groups of three letters (Goldwater, 2016). N-gram stemmers work on the probability that groups of letters belong alongside each other.

From such probability, it is possible to determine the actual root word separately from the suffix. The suffix is then stripped from the root word (Rani et al., 2015).

Figure 13 displays the steps that are taken in the n-gram stemmer to return the root word.

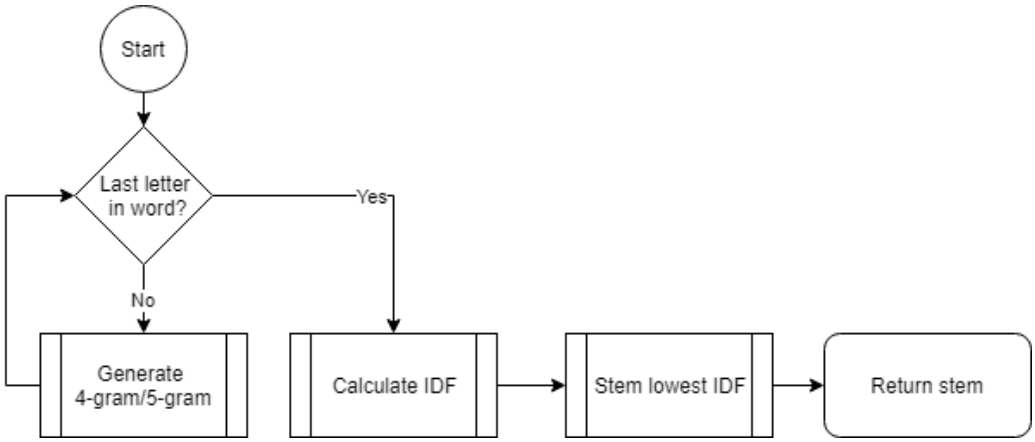


Figure 13: N-gram stemmer steps[8]

According to Figure 13, the stemmer starts off by checking if it has reached the last letter of the word. If the stemmer has not reached the last letter of the word, it continues generating 4-grams and 5-grams relative to the word itself. The 4-gram/5-gram is generated using probability calculations on the sequence of letters. Once all possible combinations of 4-grams/5-grams have been generated, an inverse document frequency (IDF) calculation is carried out (Mayfield & McNamee, 2003). The formula is as follows:

$$IDF(gram) = log(\frac{Total\ number\ of\ terms\ in\ document}{Number\ of\ terms\ in\ document\ with\ gram\ in\ it})$$

[8] The diagram is adopted and adapted from Mayfield and McNamee (2003) by converting the descriptions into a diagram.

The IDF with the lowest value is selected as the suffix. The last step in Figure 13 is to stem the word and return the stem (Mayfield & McNamee, 2003).

The advantage of this Stemming Algorithm is that it is a lot more precise as it has been developed on the combination of letters with the n-gram technique. Based on the way the algorithm is built, the stemmer is language-independent since it is context-specific and uses statistical formulae to determine the root words (Jivani, 2011). This algorithm is mainly dependent on the corpus of text provided. This means that the larger the corpus of text, the more accurate it becomes (Rani et al., 2015). Additionally, based on the description given by Mayfield and McNamee (2003), this algorithm can work with prefixes and suffixes.

The disadvantage of this Stemming Algorithm is that the algorithm makes provision for under-stemming or over-stemming due to the fact that the n-gram architecture is based on the combination of letters (Cavnar & Trenkle, 1994). The longest possible root word found in the document is one that contains a suffix, which would be the one taken as the root word. Running the algorithm is very time consuming since n-grams take a long time to build. Once those n-grams have been built, the computer requires a lot of space to store all the possible results, in addition to an indexing method to correlate the n-grams (Jivani, 2011). Mayfield and McNamee (2003) discuss how the algorithm is built based upon 4-grams or 5-grams. With such a limitation, words with less than four letter stems become redundant, which may cause inaccuracies in stemming.

3.4.7 The Hidden Markov Model (HMM) stemmer

The Hidden Markov Model (HMM) stemmer uses the Hidden Markov Model as its basis. It considers the probability of changes through different states of the word. In addition, this method does not require linguistic knowledge of the dataset, since it uses machine learning.

It uses automata graphs[9] to determine the probability for each variation change as the word moves from state to state. It then selects the highest probability to get to the actual word from the stem (Rani et al., 2015).

Figure 14 illustrates the steps that are taken in the HMM stemmer to return to the root word.

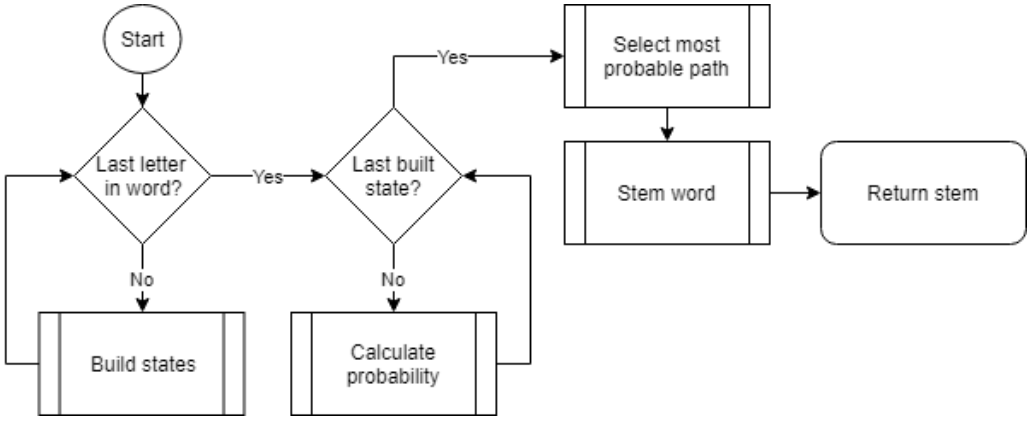


Figure 14: HMM stemmer steps[10]

Figure 14 starts by going through every letter in the input word and builds two states. A state in this scenario can be considered to be a combination of letters. The two states comprise the possible root word and the possible suffix. The states are built based on the index on which the loop is currently located. Once all the states have been created, the algorithm goes through each set of states and calculates the probability of the root word state being given the suffix state. Each calculated probability is considered a path. Once the algorithm has calculated all the probabilities, it selects the most probable path using automata graphs. The

[9] A pre-programmed graph that is designed with a predetermined set of coded instructions to perform certain tasks. In this scenario, the predefined tasks are to plot calculated probabilities on a graph and make decisions from it (Verwer, Eyraud & De la Higuera, 2014).

[10] The diagram is adopted and adapted from Saharia, Konwar, Sharma and Kalita (2013) by converting the descriptions into a diagram

next step would be to stem the word on the probable path's root word state and finally return the stemmed word (Saharia, Konwar, Sharma & Kalita, 2013).

The advantage of this method is that it can be used in a variety of languages. Based on the structure of the algorithm, it is adaptable to different languages (Jivani, 2011).

The disadvantage of this algorithm is that it cannot always be 100% accurate with such a large amount of complexity built into it (Rani et al., 2015). Another disadvantage extends from the structure of the algorithm, which has a higher possibility of over-stemming, since it calculates the shortest distance between the root of the word to the suffix (Jivani, 2011). Additionally, based on the descriptions of Saharia et al. (2013), this algorithm only works with suffixes.

3.4.8 Yet Another Suffix Stripper (YASS) stemmer

YASS stands for Yet Another Suffix Stripper. This method is a statistically based method of using lexicon logic and distance measures in the word to cluster and differentiate letters in the word to determine the suffix from the root. It also determines the variations of the root words that exist (Rani et al., 2015). This stemmer can be classified as both statistically and corpus-based (Ismailov et al., 2016).

Figure 15 illustrates the steps that are taken in the YASS stemmer to return to the root word.

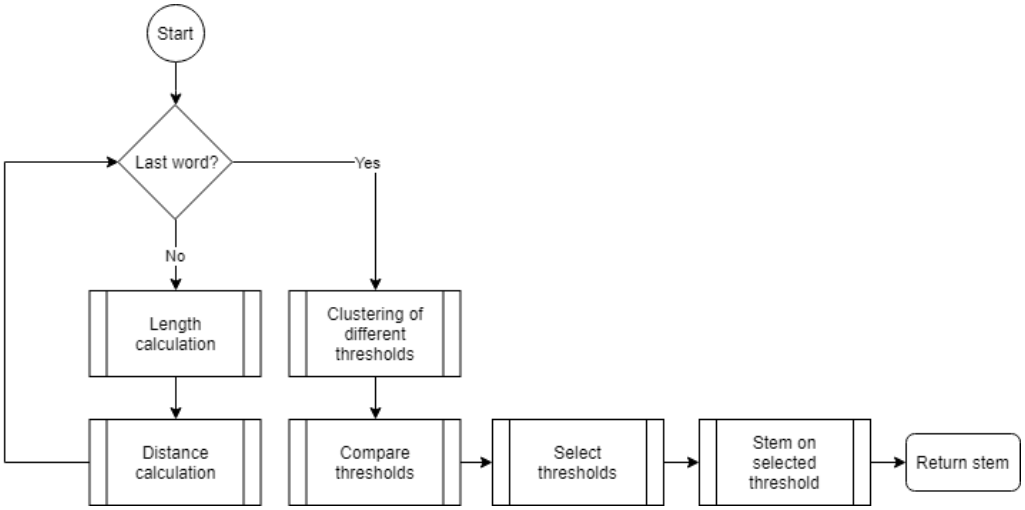


Figure 15: YASS stemmer flow chart[11]

In to Figure 15, one can see that the algorithm starts by checking to see if it has looped through all the words. If the algorithm has not gone through all the words in the corpus, it moves on to the next word and does a length calculation by providing "null" values at the end of the shorter words to match the longer words. Once the words have the same length, it performs a penalty-based calculation on every word in the body of text. The penalty-based calculation compares words and assigns a smaller penalty for the longer root word. Once penalty values have been established for all the words in the body of text, the words are clustered into homogenous groups. To determine the number of groups in which to cluster the words, all the options for the number of clusters is generated, and the thresholds of each cluster are plotted onto a line graph in comparison to the number of clusters. On the line graph, there will be a "step-like" region where the values of the threshold do not change. The number of clusters indicated at the beginning of the "step-like" region will represent the number of clusters that are required. The last step in the

[11] The diagram is adopted and adapted from Majumder et al. (2007) by converting the descriptions into a diagram.

process is to stem every word in each cluster to the “central” word since that word will contain the word with the common root word for the cluster (Majumder et al., 2007).

The advantage of this Stemming Algorithm is that it can be used across different languages without needing the language structure as base knowledge (Rani et al., 2015).

The disadvantage of this Stemming Algorithm is that it is difficult to decide the clustering centroid. It also requires a large amount of processing power (Rani et al., 2015). Since Majumder et al. (2007) mentions that the algorithm will pick the longest possible stem based on the penalty points and the given corpus, this can cause under-stemming to occur. Under-stemming causes inaccuracies to increase. Additionally, based on the descriptions of Majumder et al. (2007), this algorithm only caters for suffixes.

3.4.9 The inflectional and derivational stemmer

Inflectional and derivational stemming is a combination of both inflectional changes and derivational changes to words and how they change based on situations (Rani et al., 2015). This method also involves a corpus that can analyse the changes. A large corpus of text is used for these Stemming Algorithms. Therefore, it is classified as a mixed stemmer (Rani et al., 2015). Inflectional changes refer to the alteration to the word to express different grammatical parts of a sentence such as tense, case, voice and gender. Derivational changes refer to the addition of a prefix or suffix to alter the meaning of the word. Derivational changes can be associated with the part of speech of the word (Vijayarani et al., 2015). Some examples of inflectional and derivational stemmers are those of Kroverts and Xerox (Vijayarani et al., 2015).

The three steps of the Kroverts stemmer can be seen in Figure 16.

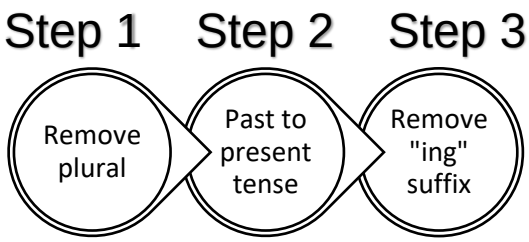


Figure 16: Steps in the Kroverts stemmer

The three steps in the Kroverts stemmer demonstrated in Figure 16 consist of removing the suffix of plurals, converting past tense to present tense, and removing the “ing” suffix from words. With all three these steps, a dictionary approach is taken where the word of interest is compared to a dictionary of words. If the word is found, the rule is applied. However, if the word is not found in the dictionary, it is no longer stemmed. This stemmer is very accurate when it comes to the words that are in the dictionary as it correlates directly with the grammatically correct words. However, because it has to look at a large dictionary of words, the processing time will increase. This algorithm will not be efficient in terms of time where large data samples are concerned. This algorithm is also dependent on the dictionary that is provided. This means that, based on the provided dictionary, the algorithm would not be able to consistently produce correct results (Vijayarani et al., 2015). Therefore, one can say that the algorithm depends on the dictionary that is fed into it. Additionally, based on the descriptions of Vijayarani et al. (2015), this algorithm only caters for suffixes.

3.4.10 The Xerox stemmer

The Xerox stemmer, or the Xerox inflectional and derivational analyser, was created by employees of Xerox. They built this stemmer by putting together a database that correlates words to their possible stem in the dictionary (Vijayarani et al., 2015).

Noun -> (Singluar)	Verbs -> (Infinitive)	Adjectives -> (Positive Form)	Pronoun -> (Nominative)
•Babies -> Baby •Geese -> Goose	•Failure -> Fail •Sleeps -> Sleep	•Better -> Good •Worse -> Bad	•His -> he •They're -> They

Figure 17: Category of look-up for the Xerox stemmer[12]

Figure 17 illustrates the category of words that undergo a transformation. In the figure, there are four areas of focus: nouns, verbs, adjectives and pronouns. Within all four these areas, the goal of the stemmer is to convert all variations of a word within these classifications into one form. Nouns are all converted to their singular form. Verbs are all converted into the infinitive form. Adjectives are all converted into one variation of the adjective. Pronouns are all converted into the normative form. These transformations are all mapped from the database. Similar to the Kroverts stemmer, this stemmer can be very accurate, based on the given database. However, the results are not necessarily always consistent, based on the database provided (Vijayarani et al., 2015).

A singular noun names one “thing”, “place”, “person” or “idea”, while a plural noun names more than one of the aforementioned. Plural nouns are denoted with the originating singular noun with an additional suffix at the end of the word. By transforming the word into its singular form, one removes the plural suffix from the word (Huddleston & Pullum, 2005).

The infinitive verb is the verb in its most simplistic format, which means that this form would be the chosen one for dictionaries. This format has no appending suffix (Huddleston & Pullum,

[12] The diagram is adopted and adapted from Vijayarani et al. (2015) by converting the descriptions into a diagram.

2005). With the example in Figure 17, the word “failure” is converted to the word “fail” since the word “fail” is the infinitive form of the verb.

There are three levels of adjectives: positive, comparative and superlative. Positive adjectives are the most basic form of the adjective (Huddleston & Pullum, 2005). An example provided in Figure 17 with all three levels is “good”, “better” and “best”. The word “good” is in the positive form of the adjective, whereas “better” is in the comparative form and “best” is in the superlative form of the adjective.

Nominative pronouns are usually the subject and the action makers of the sentence. Nominative pronouns are “I”, “you”, “he”, “she”, “it”, “they” and “we” (Huddleston & Pullum, 2005). In the example given in Figure 17, the pronoun “his” is transformed back to “he”.

The advantage of the Xerox stemmer is the speed it provides for a corpus-type stemmer, compared to the Kroverts stemmer. It is always very accurate since it encapsulates all the words in the dictionary (Vijayarani et al., 2015).

The disadvantage of the Xerox stemmer is that it is difficult to implement in other languages since the language structures are different across different languages. As mentioned above, another disadvantage would be the fact that the results of the stemmer depend on the database provided, which can cause inconsistent results (Vijayarani et al., 2015). Additionally, based on the descriptions of Huddleston and Pullum (2005), this algorithm only caters for suffixes.

3.4.11 Corpus-based stemmer

Corpus-based stemming involves a body of text. It determines the combination of the suffix size based on statistical calculations done by comparing a specific body of text

(Vijayarani et al., 2015). The underlying hypothesis is that the words that need to be grouped together would appear in the same document, even if two words have the same stem. However, if they do not both appear in the same body of text, they should not be grouped together based on the root stem of the words (Jivani, 2011).

This method helps to fix the grouping of different words in the same stem. An example of such a grouping is that the words “policy” and “police” result in the same stem in Porter’s algorithm, but they do not have the same conceptual meaning (Vijayarani et al., 2015).

Figure 18 shows the steps in the corpus-based Stemming Algorithm.

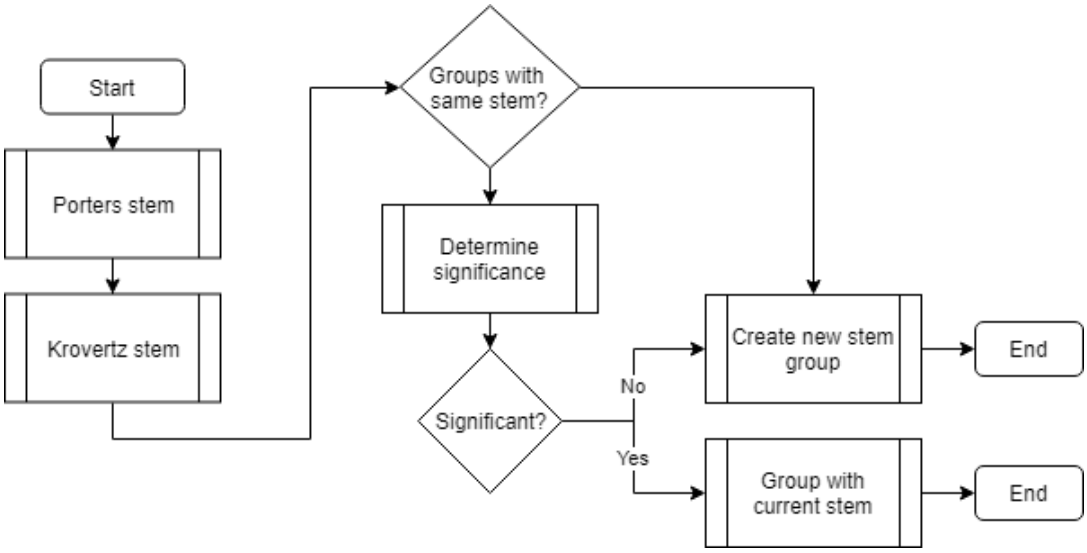


Figure 18: Steps in the corpus-based Stemming Algorithm[13]

According to Figure 18, each word goes through four processes. The first two processes involve performing Porter’s stemmer and the Krovertz stemmer on the word. This means that this algorithm encapsulates both these stemmers respectively. Once these two processes

[13] An algorithm adopted based on descriptions given by Jivani (2011).

have been applied, the stem of the word will be determined. The next few steps involve determining if the words should be grouped together with other similar stems. The algorithm first checks to see if there are existing groups of the same stem. If these do not exist, it creates a new grouping. If there is a group with the same stems, it will use a statistical formula to determine the significance of the word in relation to all existing groups. The highest significance above the threshold is selected for this grouping. If the stem does not meet the threshold for any groupings, a new group is created.

The statistical formula that is used to determine the significance is as follows:

$$Em(a,b) = \frac{nab}{(na + nb)}$$

In this formula, “a” is the new word and “b” is the word from the existing group; “na” and “nb” are the number of times the word “a” and the word “b” occur in the same body of text; “nab” is the number of times both the word “a” and the word “b” occur in a given text range. This text range needs to be determined by the user of the algorithm.

The advantage of this method is that it works based on the given body of text that allows each stemming practice to be context specific, thus increasing accuracy. This means that the stemming performed for different bodies of text will result in different groupings of words. Therefore, this algorithm has a high probability of not grouping words together that are not meant to be grouped together, which solves a lot of issues in language related to heteronyms (Vijayarani et al., 2015).

The disadvantage of this Stemming Algorithm is that one needs to develop a statistical measure for each performance of the algorithm. This is due to the fact that the algorithm is

built on a statistical formula that is specific for each document (Jivani, 2011). Another disadvantage is that this algorithm takes a lot longer to process compared to the truncating methods, as it requires the first two steps of Porter's algorithm and the Krovertz algorithm to be performed before the statistical formula is applied (Jivani, 2011).

3.5 COMPARISON

After reviewing every group of Stemming Algorithms and each Stemming Algorithm itself, the advantages and disadvantages of each Stemming Algorithm are presented. Table 2 presents the advantages and disadvantages of each Stemming Algorithm individually. Table 3 presents the advantages and disadvantages of each Stemming Algorithm as a group.

Table 2: The advantages and disadvantages of each Stemming Algorithm

Stemming Algorithm	Advantage	Disadvantage
Porter's	• Low storage space • Smaller error margin • Lightweight stemmer – fewer computer resources required	• Weak exception handling • Over-stemming • Language dependent • Suffix only
Snowball	• Caters for a variety of languages • Low storage space • Smaller error margin • Lightweight stemmer	• Exception handling is not that great • Over-stemming • Can have an incorrect list of rules • Semi-language dependent • Suffix only
Lovins	• Quick • Exception handling is a lot better than Porter's	• Exception handling is a limitation • Over-stemming occurs • Lage data consumption • Language dependent • Suffix only
Dawson	• More accurate than Lovins • Quicker than Lovins	• Heavily language dependent • Exception handling is a limitation • Over-stemming occurs • Large data consumption • Suffix only
Paice/Husk	• Easy to set up for new languages	• Over-stemming and under-stemming occurs • Heavy on resources. • Suffix only
N-gram	• Cross-language capability	• Resource hungry

Stemming Algorithm	Advantage	Disadvantage
	• More accurate than truncating methods • Caters for both prefixes and suffixes	• Time-consuming • Under-stemming and over-stemming may occur • Limited to words with at least four letters in the stem
HMM	• Cross-language capability	• Over-stemming may occur • Accuracy will never reach 100% • Dependant on given corpus – inconsistent • Suffix only
YASS	• Language independent	• Resource hungry • Under-stemming may occur • Suffix only
Inflectional and derivational	• Extremely accurate	• Heavily language dependent • Database-dependent, which can cause inconsistent results • Suffix only
Corpus-based	• Solves heteronym issues	• Impractical • Time-consuming • Language dependent • Suffix only

There are three columns in Table 2. The first column lists the Stemming Algorithms. The second column gives the advantages of each Stemming Algorithm. The third column gives the disadvantage of each Stemming Algorithm.

In Table 2, there are a few groupings of disadvantages: resource consumption, accuracy, performance duration, prefix inclusion and language dependency. The Stemming Algorithms that have accuracy issues, including over-stemming and under-stemming, are Porter's, Lovins, Dawson, Paice/Husk, HMM and YASS. The Stemming Algorithms that have resource consumptions as a disadvantage are Lovins, Paice/Husk, N-gram and YASS. The next grouping of disadvantage is language dependency, which is encountered with Porter's, snowball (semi-dependent), Lovins, Dawson, Paice/Husk (semi-dependent), the inflectional and derivational, and the corpus-based algorithms. The last grouping of disadvantage relates to performance. As the most accurate algorithm, the corpus-based algorithm comes with performance issues, which are

encountered a lot of the time. Of all the algorithms, only the n-gram stemmer caters for prefix stemming.

Seventy per cent of the Stemming Algorithms evaluated are incapable of adapting to other languages. The algorithms that can cater for different Stemming Algorithms have accuracy issues.

Table 3: The advantages and disadvantages of each type of Stemming Algorithm

Type of Stemming Algorithm	Advantage	Disadvantage
Truncating	• Quick • Low processing power required • Low computer memory required	• Exceptions are not handled dynamically • Language dependant based on given rules • Cannot stem to a specific context
Statistical	• Not dependent on language. • Exception handling is better than the truncating algorithm	• Slow • Large processing power required • Large computer memory space required
Mixed	• Corpus-based is quick • Exception handling is better than the truncating algorithm (based on the given corpus) • Much more accurate compared to the other two methods	• Dependent on language (based on the given corpus)

The advantages and disadvantages of each type of Stemming Algorithm are compared in Table 3. The table has three columns in the table: type of Stemming Algorithm, advantage and disadvantage. The type of Stemming Algorithm indicates the group or type of Stemming Algorithm. The advantage column indicates the advantages of each type of type of Stemming Algorithm, and the disadvantage column indicates the disadvantages of each type of Stemming Algorithm. Three types of Stemming Algorithms are given in the table: truncating, statistical and mixed.

Table 3 shows that while truncating methods have the advantage of speed, they do not have the advantage of cross-language capabilities or accuracy. While statistical methods allow for some

form of cross-language capability, they use a lot of processing power. Mixed methods are more accurate, while corpus-based methods sort out a lot of issues related to heteronyms. However, they become extremely language dependent. Therefore, none of the Stemming Algorithms presented have the combined advantage that each algorithm can provide.

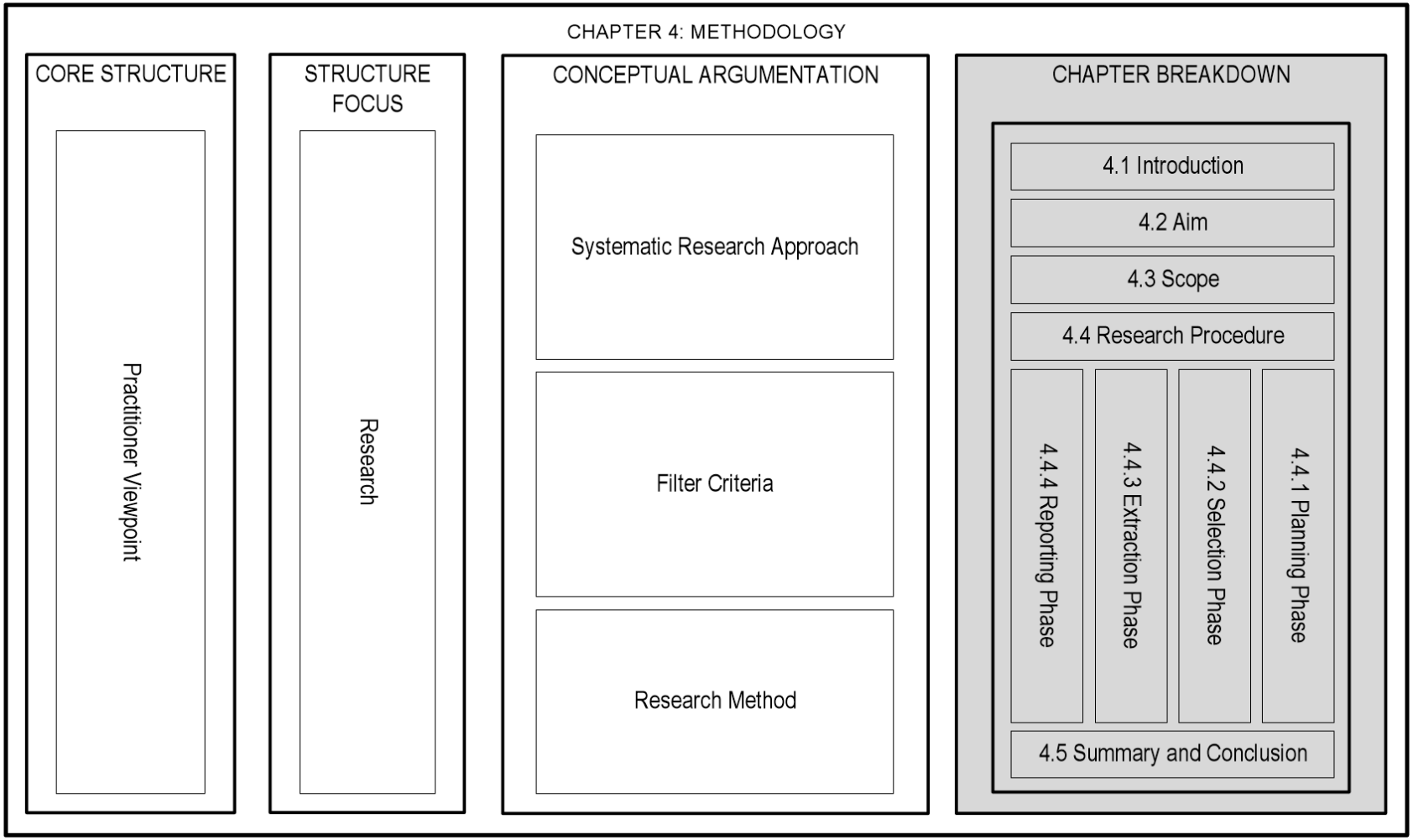
3.6 SUMMARY AND CONCLUSION

In this chapter, the different algorithms that could be used to stem words in unstructured text were discussed. The different groupings of Stemming Algorithms were discussed and grouped together. Each algorithm was discussed in terms of their descriptions, steps in the stemming process, advantages and disadvantages. A table was then presented in which each different grouping or stemming type was compared. A second table was presented to compare the advantages and disadvantages of each algorithm.

The Stemming Algorithms were categorised into three groups: truncating, statistical and mixed algorithms. Truncating methods include the Lovins, Porter's, snowball, Dawson and Paice/Husk methods, which remove suffixes from the words by rules and conditions with defined word endings. Statistical methods include the n-gram, YASS and HMM methods, which are based on the calculations done through probability or other statistical formulae to determine the suffix of the word and extract the stem. Mixed methods include the inflectional and derivational, and the corpus-based methods, which are either a combination of the truncating and statistical methods, or the indexed dictionary method.

From the comparison of the different groupings of Stemming Algorithms, it can be concluded that none of the Stemming Algorithms presented in this chapter has the combined advantage that every algorithm can provide.

Seventy per cent of the Stemming Algorithms that were evaluated are incapable of adapting to other languages. The algorithms that can cater for different stemming conditions have accuracy issues. All the algorithms will have at least one of the following disadvantages: resource consumption, accuracy, performance duration, being incapable of prefix stripping and being language dependent. The methodology is discussed in the following section.



Chapter Map 4: Methodology

CHAPTER 4:
METHODOLOGY

4.1 INTRODUCTION

This chapter presents the research design, which includes the philosophy, research strategy, methodological choice and data collection method, and provides the selected approach for the research. It includes the articles that have been accessed, the selection process, method of analysis and interpretation of the articles to ultimately answer the research question: *What are the advances of Stemming Algorithms in Text Analysis over the past six years (2013 to 2018)?*

4.2 AIM OF THE CHAPTER

This chapter provides a detailed overview of the research that was conducted. This allows the research to be duplicated in future to validate the reliability of the research study. The research procedure is also discussed so that the process can be structured until the results of the research are presented to gain a deeper understanding thereof.

4.3 SCOPE OF THE CHAPTER

To achieve the aim of the research, this chapter will cover the execution of the research by discussing the following phases: the planning phase, the selection phase, the extraction phase and the reporting phase.

4.4 RESEARCH PROCEDURE

A systematic review was performed of academic articles published between 2013 and 2018. Research stages were adopted from Okoli and Schabram (2010) and adopted by including

additional components from Attride-Stirling (2001). The data preparation and presentation were adopted from Van Deventer (2013). The Text Analysis process was adapted from Aryal et al. (2015), which is an addition to the extraction phase of Okoli and Schabram (2010). The Text Analysis process will be further discussed in the extraction stage. Calculations and interpretations are done in the reporting stage, as described by Kerlinger and Lee (2000). This will be further discussed in the reporting stage. Figure 19 depicts the steps of the methodology, which was divided into four different stages.

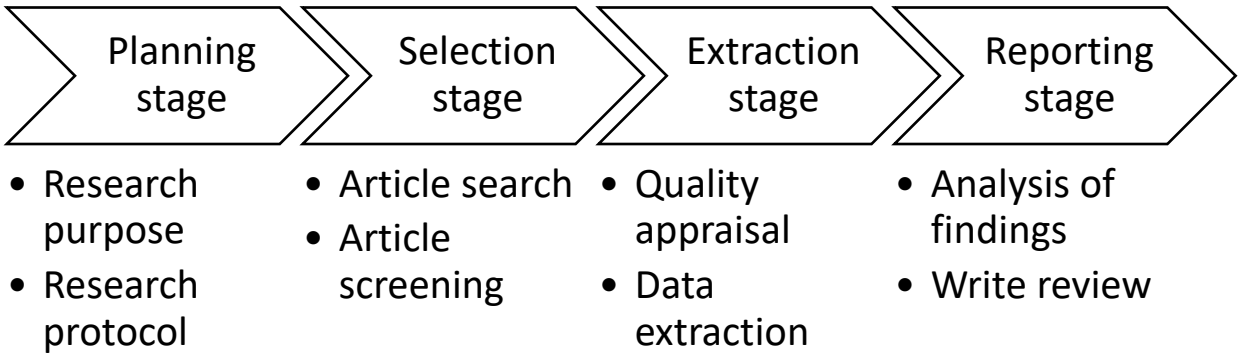


Figure 19: **High-level research stages (Okoli & Schabram, 2010)**

As illustrated in Figure 19, the research was carried out in four main stages. Each stage contained two steps. The four stages were the planning, selection, extraction and reporting stages. These stages are discussed in detail in the sections that follow. The results of the data extraction step (within the extraction stage) are discussed in more detail in Chapter 5, while the results of the analysis of findings step (in the reporting stage) are discussed in more detail in Chapter 6. The last step, write review, completes the research document.

The following sections will discuss the steps outlined in Figure 19 in more detail, beginning with the planning stage.

4.4.1 Planning stage

The first stage of the process outlined in Figure 19 is the planning stage. This stage comprises two steps: research purpose and research protocol (Okoli & Schabram, 2010).

When focusing on the research purpose, the purpose of the research that has been conducted is confirmed. The aim of the research, stipulated in Chapter 1 , was to establish the existing research conducted on Stemming Algorithms between 2013 and 2018. This, additionally, fulfils the following objectives:

- Identify the characteristics of Stemming Algorithms
- Identify the problems associated with Stemming Algorithms in text and Text Analysis
- Consider the application of common Stemming Algorithms in Text Analysis
- Identify how Stemming Algorithms have been applied and changed in Text Mining over the past six years (2013 to 2018)

After defining and verifying the purpose of the research, the researcher starts to work with what is known as the research protocol. The research protocol provides a set of rules and procedures within a systematic review to enforce consistency and reliability. The reason for the protocol is to ensure repeatability in the future that will result in the same conclusion (Okoli & Schabram, 2010). The research protocol discusses the rules required for the completion of the research. This is further discussed in the stages to follow: selection, extraction and reporting.

After clearly verifying the purpose of the study to ensure focus, and setting the research protocol to ensure consistency, the selection stage is launched.

4.4.2 Selection stage

The second stage in the research process is the selection stage. This stage comprises two steps: article search and article screening. For the purpose of this study, the process of data collection is also discussed in this stage.

The article searching step entails searching through two databases: Open WorldCat and ProQuest. These databases were the only ones to which the researcher had access at the point of this study. The instructions to set the databases under Google Scholar are illustrated in Figure 20.

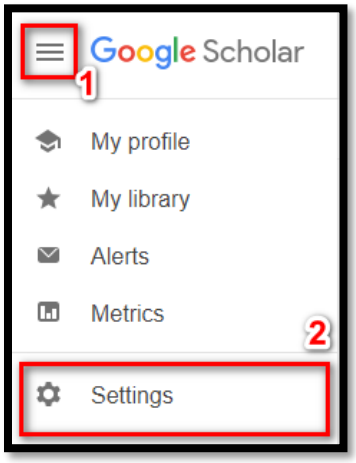


Figure 20: Google Scholar settings

Figure 20 indicates the steps required for the user to find the settings tab in Google Scholar. In this figure, two numbers are indicated as “1” and “2”. These numbers indicate the steps taken in sequence. Step 1 was to click on the menu of three lines in the top left-hand corner. This step opens a menu strip that provides the researcher with options. The second step was to click on the settings button. To specify any configurations required for searching anything other than the keywords, the researcher needed to use the settings tab. This diagram was used for the

configuration of other settings as well. Therefore, it will be referred to again later in this chapter. The instructions that are required to set up the required databases and access controls to select the datasets and perform the search are illustrated in Figure 21.

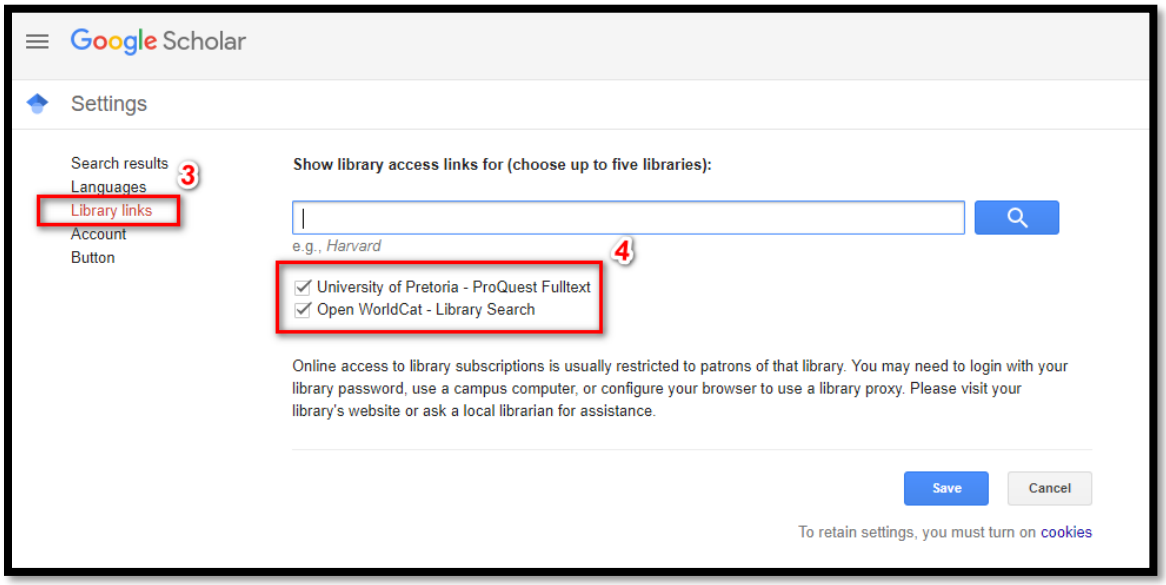


Figure 21: Setting databases on Google Scholar

Figure 21 directs the user to the databases to be used. The steps in this figure follow from the previous figure. In Figure 21, there are two steps, indicated by “3” and “4”. Step 3 allows the researcher to indicate which database or library the search engine should include, while Step 4 indicates all the accessible databases for the research. The researcher selected both the ProQuest and Open WorldCat research categories. Once these four steps had been completed, the researcher was able to search the accessible databases.

Once the databases have been selected, the researcher’s next task is to set the criteria to be used to search the articles. Table 4 shows the criteria that were used and provides a brief description of the underlying reasoning when searching for articles.

Table 4: Search criteria for article search

Number	Criterion	Logical understanding
1	"Stemming Algorithm" or "stemmer" contained in the title, abstract, keywords or text.	The research topic is "Stemming Algorithm". Other search areas were included to make sure of a comprehensive inclusion of articles from the search. This included the application of Stemming Algorithms.
2	"Text Mining" or "Text Analytics" contained in the title, abstract, keywords or text.	Stemming Algorithms is a process within Text Mining and Text Analytics (Vijayarani et al., 2015). Stemming Algorithms might be established through articles that contain the search terms "Text Mining" and "Text Analytics" since it falls under both fields. This also ensures that the articles returned are within the relevant field of study.
3	"Language" contained in the title, abstract, keywords or text.	The word "stemming" could be used in other practices other than the field of language, for example, stemming a plant. To focus the data collection on the field of language, the keyword "language" is used.
4	An "empirical study" (removing searches containing the words "review", "overview", "case study" and "survey").	The article must be of an empirical type to show an establishment in knowledge and not just limited to reviews or overviews (Okoli & Schabram, 2010).
5	The file format must be *.pdf.	As per specification in demarcation.
6	The document must be a journal article.	To ensure the reliability of the source.
7	The date of publication must be between 1 January 2013 and 31 December 2018.	This is to derive the latest work on Stemming Algorithms. As mentioned in the problem statement, 2019 was excluded.
8	The article must be written in English, even if the algorithm was developed for the content of another language.	The limitations of the researcher's capabilities need to be taken into consideration.

Table 4 has three columns. These include the number of the criterion, the description of the criterion and the logical understanding or reasoning behind the criterion. The methodology followed by the research (from the collected articles) should be a valid empirical study to ensure that it enables an observation or experiment to be carried out. Additionally, the resource should be a journal article, to ensure that it is a valid publication. It should also be in the format of a pdf file, as per the demarcation. The date of publication of the article should range between 2013 and 2018. Finally, the article should be written in English so that the author can read the article.

To fulfil the search criterion laid out in Table 4, the following search string was used:

> *"("Stemming Algorithm" OR "stemmer") and ("Text Mining" OR "Text Analysis") language -overview -review -"case study" -survey filetype:pdf"*

The abovementioned search keywords cover the first four points in Table 4. Table 5 shows how each section of the search keywords fulfils the requirements set out in Table 4:

Table 5: Search keyword breakdown

Keyword	**Criterion**
("Stemming Algorithm" OR "stemmer")	1 - *"Stemming Algorithm"* or *"stemmer"* contained in the title, abstract, keywords or text.
("Text Mining" OR "Text Analysis")	2 - *"Text Mining"* or *"Text Analytics"* contained in the title, abstract, keywords or text.
language	3 - *"language"* contained in the title, abstract, keywords or text.
-overview -review -"case study" -survey	4 - Empirical study (removing the searches containing the words "review", "overview", "case study" and "survey").
filetype:pdf	5 – The file format must be *.pdf.

In Table 5, one can see the execution of the keyword and the criterion. In the keyword column, the search criterion is broken up into sections. This is so that each section can be explained and compared to the respective criterion. The criterion is taken from Table 4, combining the criterion number with the criterion column. In the first row of the table, the keywords that are used instruct the search engine to search specifically for the combination of keywords “Stemming Algorithm” or just the exact match of “stemmer”, followed by either of the two exact matches: “Text Mining” or “Text Analysis”. The keyword “language” is placed in Table 5 to make sure that the article is relevant to language. Next, is a list of keywords that we do not want to include for the purposes of this study, since they involve non-empirical research, which is a requirement as stated in the fourth row of Table 5. Lastly, the indication “filetype:pdf” ensures that the search engine only returns results that are in a pdf format.

The next set of criteria applied in this study, searching for journal articles only, could not be placed in the keywords of the search engine, but rather in the settings that needed to be configured.

To fulfil the sixth criterion in Table 5, where the results are all journal articles, the steps in Figure 20 are followed, together with the steps in Figure 22, which are applied to limit the search to journal articles only.

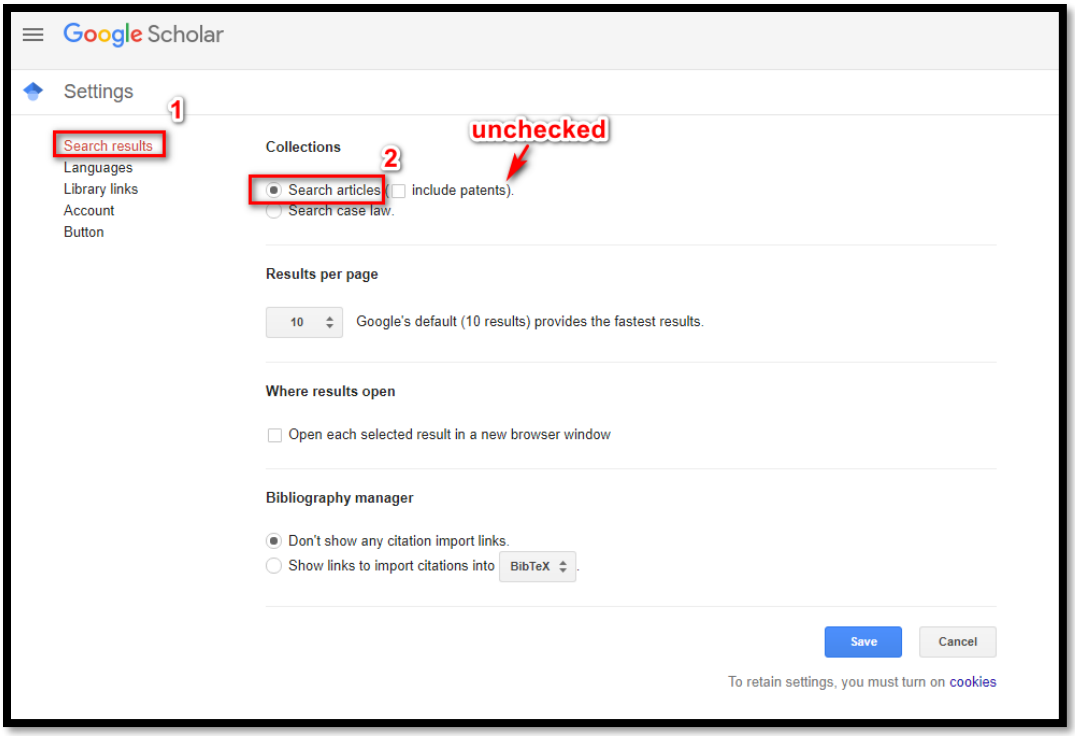


Figure 22: Google Scholar limiting journal articles

In Figure 22, we refer to Step 1 and Step 2. The boxes show what the researcher clicked on, and the number shows the sequence of selections. The first step was to click on the “search result” menu on the left to make sure that the “journal option” was selected. Additionally, the researcher had to make sure that the “include citation” option was unticked. This ensured that all the results returned were only assigned as journal articles. However, incorrect associations of journal articles are still probable. As such, the results were filtered manually later to identify the dataset again.

The next criterion in Table 4 related to ensuring that the results were within the date range 2013 to 2018. The following diagram depicts the steps that were followed to limit the results:

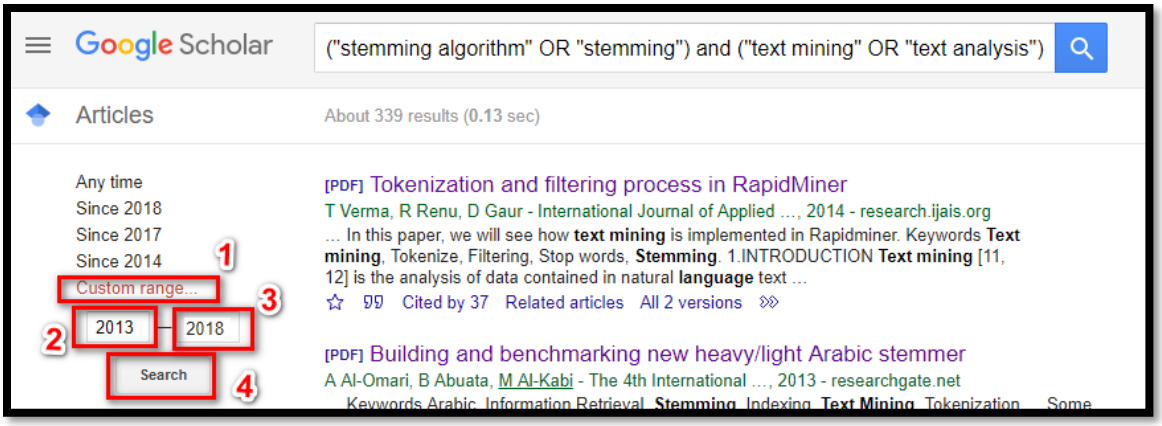


Figure 23: Google Scholar setting date range

In Figure 23, the boxes show where the researcher clicked on the screen, and the numbers show the sequence of steps to be followed. These steps were completed separately from the steps indicated in Figure 20. Step 1 indicated that the researcher clicked on the custom range link. The controls on steps 2, 3 and 4 were then displayed. Then the researcher placed 2013 and 2018 in point number 2 and 3 respectively. Once both these date ranges had been entered, the researcher completed that setting by clicking on the button indicated in point number 4.

To make sure that all articles returned from WordCat and Proquest, as searched by Google Scholar, were in English, the steps in Figure 20 were followed, and the results refined as indicated in Figure 24.

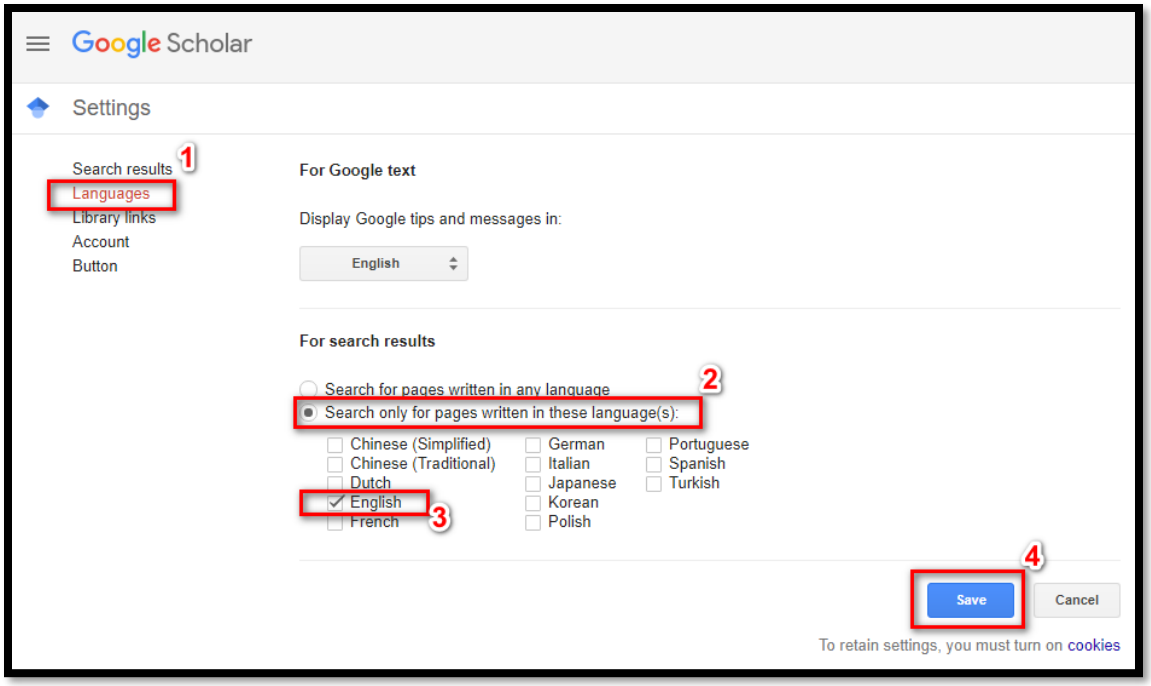


Figure 24: Google Scholar language settings

In Figure 24, the boxes show where the researcher clicked, and the numbers show the sequence that was followed. The first step was to click on the language option on the left. The second step was to select the option “search only for pages written in these language(s)”. The third step was to select the option “English” and make sure that all the other options were unticked. The final step was to click on the save button to make sure that the results reflected accordingly.

The final search resulted in 354 articles. To download all 354 articles, an automation process was used through a program called WebHarvy. WebHarvy is an automated data collection tool that allows the user to indicate sequences of steps taken on websites to collect specific data (Haddaway, 2015). Once WebHarvy was open, the researcher could navigate to Google Scholar and enter all the criteria mentioned above. Once the Google Scholar results were as expected, the following approach was followed in WebHarvy:

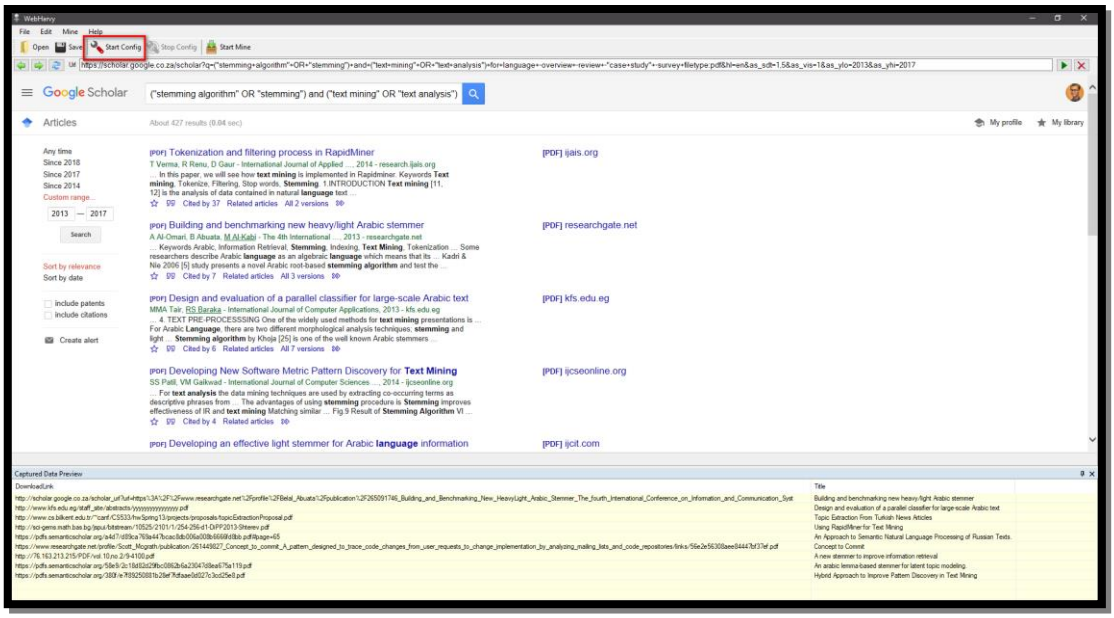


Figure 25: WebHarvy – start configuration

Figure 25 displays the WebHarvy screen, with the indication of how to begin the configuration. In this figure, one can see Google Scholar opened with all the configurations mentioned earlier. One can also see that there is only one red square that is around the button “start config”. Once the configuration begins, this button is disabled, and the “stop config” button is enabled. The configuration allows WebHarvy to understand the sequence of steps that it needs to take to complete the task (Haddaway, 2015).

Figure 26 shows four steps. Again, the boxes show where the researcher clicked, and the numbers show the sequence of the steps that were taken. The first step was to click on the link in the result set of Google Scholar indicated in Figure 26. The second step was to click on “capture target URL”. The next step was to enter the title of the column for all the research articles on the page. The last step was to click on OK to complete the configuration for the URL. In the given example, the title of this column is “DownloadLink”.

This allowed the collection of a list of URLs from a Google Scholar search. A temporary example of the results is displayed in a table at the bottom of the window.

Once a list of the URLs was extracted, a list of corresponding titles of each article was required. This made it possible to save the downloaded file with the corresponding article name.

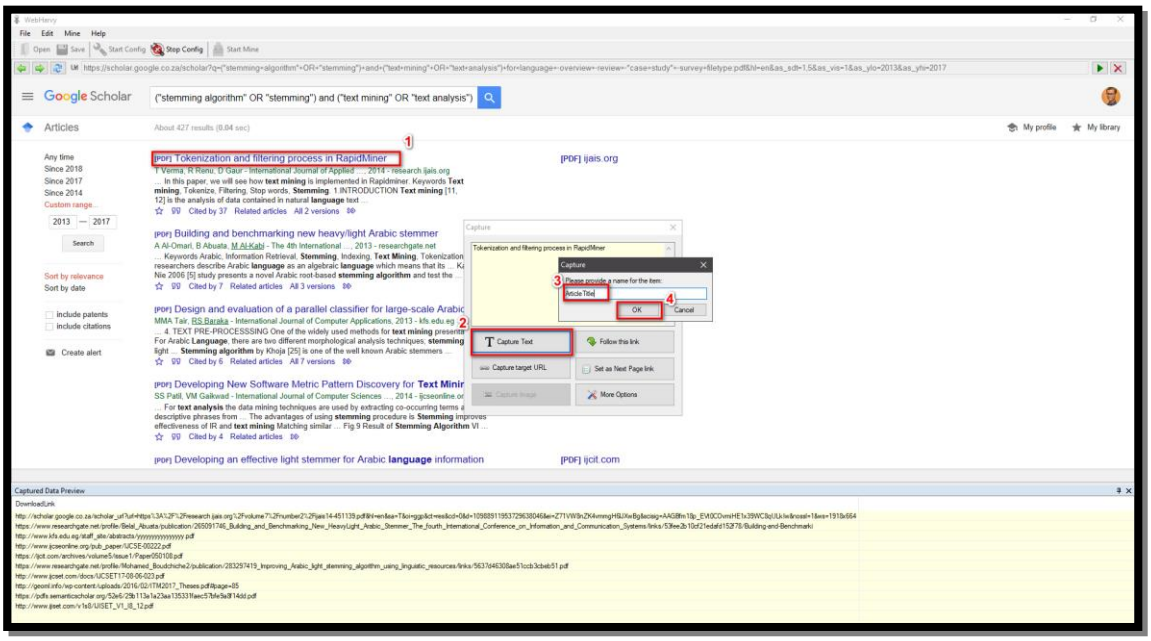


Figure 26: WebHarvy – get article name

Figure 26 shows the steps that are required to get the article names. Again, the boxes show where the researcher clicked, and the numbers show the sequence of steps that were followed. The first step was to click on the “title” link, followed by selecting the “capture text” option and then filling in the text “article title”. This collected a list of titles and correlated them with the target URL. These example results were displayed in the table at the bottom of the screen. Each title was on the right of the corresponding URL in tabular format.

The last two things to complete was to configure WebHarvy to go to the next page and start collecting articles on consecutive pages.

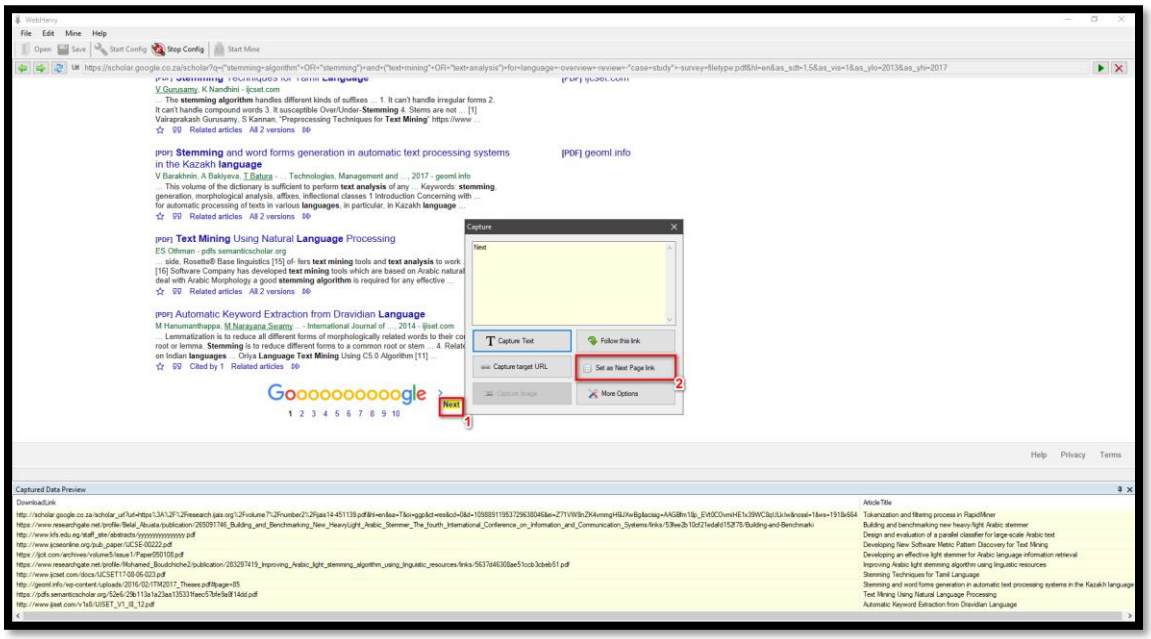


Figure 2: WebHarvy – next page link

Figure 28 shows the steps that are required to set the next page button for WebHarvy. Once WebHarvy has been configured to follow the page links, it moves on to the next page until it reaches the end of the collection of pages and extracts all the relevant information. The first step was to click on the word “next” in the pager of Google Scholar results. The researcher then selected the option “set as next page link” on the WebHarvy pop-up dialogue.

Figure 29 indicates the stop configuration process.

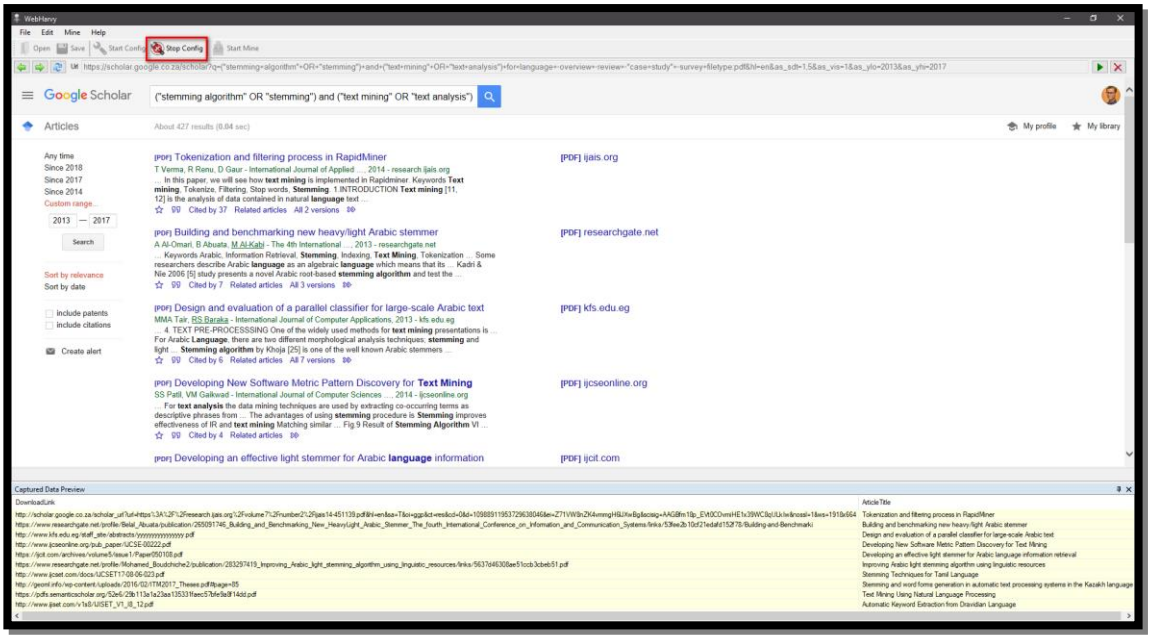


Figure 3: WebHarvy – stop configuration

Figure 29 shows the steps that are required to stop the configuration process in WebHarvy. Once WebHarvy has been configured with all the steps, it needs to be configured when to stop. In the figure, the boxes show where the researcher clicked, and the numbers show the sequence of steps. The only step was to click on the instruction “stop config” in the menu strip.

The last step is to give WebHarvy the command to begin the harvesting process, as illustrated in Figure 29.

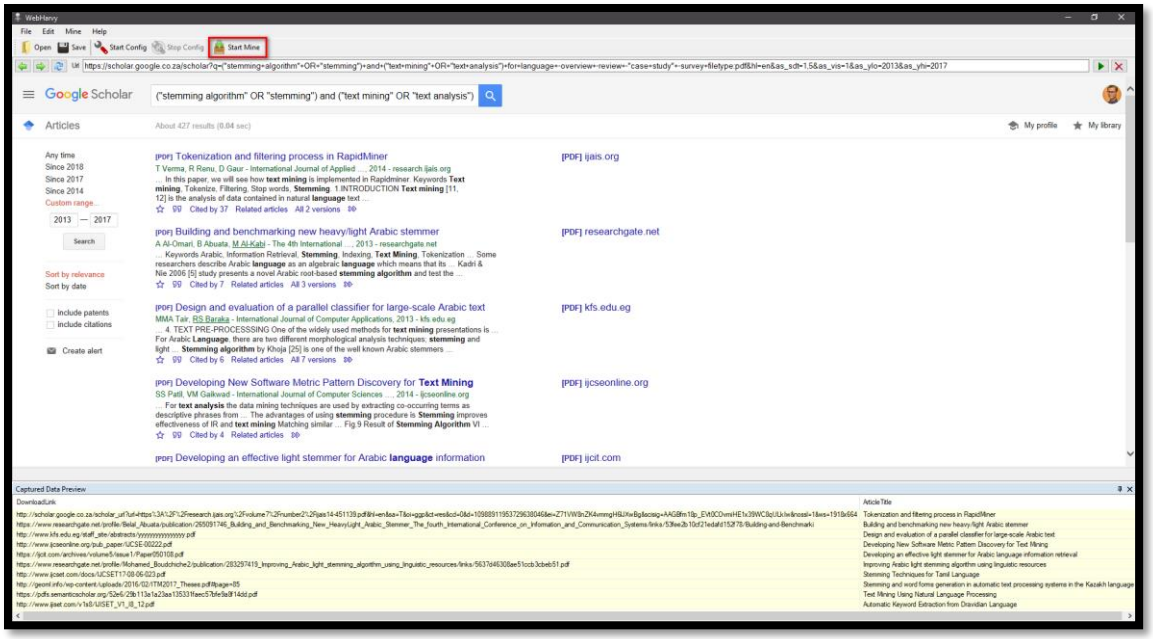


Figure 29: WebHarvy – begin mine

In Figure 29, there is only one box, so the number is removed. The only step is to click on the button “start mine” to start the mining process. A window popup confirms whether the user is ready to start mining. The only thing left to do is to select “mine until the end” and confirm the start of the mining process. WebHarvy now follows all the mouse clicks and links specified by the research and collects all the data as mentioned.

The results gathered from WebHarvy, are in a tabular format and is attached in APPENDIX 4-A. The results are in a tabular format of two columns: “downloadLink” and “title”. The “downloadLink” column displays the URL from which the file can be downloaded. The “title” column displays the title of each article that corresponds to the respective URLs. Providing an example of a dataset from the table, the first row has the following URL:

“https://research.ijais.org/volume7/number2/ijais14-451139.pdf.”

The title of the article that can be downloaded from that URL is as follows:

"Tokenisation and filtering process in RapidMiner."

This paper will appear in the given example as it covers all the functions that RapidMiner covers in the pre-processing step of the Text Analysis process, which also discusses the program's ability to handle Stemming Algorithms. This example shows how Stemming Algorithms have been applied to other software applications. This was expected from the steps followed above. This was then saved and exported to any file that can save the details. The preferred option was CSV.

Once these links were collected, a small program was written in C# to download all the required files and save them with their respective title names. The snippet of this code is attached in APPENDIX 4-B. The dependencies required for the program to run are in the first line of the code. Any version of C# compiler will work for the code snippet. The development environment used by the author was Visual Studio 2015 Enterprise edition. In the code, both the URL and the corresponding title were provided. A list of all the URLs collected and the corresponding list of titles from WebHarvy were fed into the program. The program looped through each URL and downloaded the corresponding pdf file from the given URL and then saved it with the corresponding title name.

Once the download was complete, another program was built to double-check that all the articles had been downloaded completely and successfully. The code snippet is attached in APPENDIX 4-C. The same development environment requirements will apply to the code in APPENDIX 4-B. This code collected all the file names from a folder and cross-checked them with the original list of file names that was collected from WebHarvy. The titles that did not have a corresponding file were added to the missing list. The anomalies and failures were handled manually by downloading them from a browser and adding them to the folder.

The final literature was processed in the “article screening” step. This step is performed to scale down the number of articles that were accessible by the University of Pretoria. There were three main criteria for the screening. The first criterion was the permanency of the journal in which the article was published. The second criterion was the fact that the article had to focus on the topic of Stemming Algorithms or their application. The third criterion was that the author had to have access to the article through the access rights of the University of Pretoria. Thanks to Google Scholar, these criteria had already been accounted for in the search engine.

For the purposes of this study, the first screening criterion required the author to evaluate the title of the article. The title needed to have a clear indication of the establishment of knowledge on Stemming Algorithms or their evaluation or application. If the article did not fit the criterion, it was immediately disregarded as it was considered irrelevant.

The second screening criterion required the author to read the abstract of each article. Based on the relevance to the topic of the Stemming Algorithm within the practice of Text Analysis or Text Mining and under the context of language(s), the article was either selected or removed. The methodology of the article was also reviewed to make sure that the article was an empirical study. If the abstract and the methodology were not sufficient to make a decision, the rest of the text was read to finalise the decision. The mere mention of Stemming Algorithms for languages was not sufficient to make a decision. The article needed to provide some sort of knowledge contribution to the algorithms.

The third screening criterion consisted of the author eliminating articles that he could not access or download. The author attempted different methods of gaining access to journal databases that contained journal articles, which resulted in time consumption with negative feedback.

The first two criteria filtered the original results down to 354 articles. The last criterion resulted in the removal of four articles, as they were either inaccessible or could not be found. These four articles were as follows:

Table 6: Articles not found

Article title	Year of publication
“A good read”	2013
“Similarity for classical Arabic poetry ranking”	2013
“Short paper_”	2015
“JaTeCS, a Java library focused on automatic text categorisation”	2016

Table 6 shows a list of articles that were not found. Each file name is provided in the column on the left. The articles’ corresponding year of publication is indicated in the column on the right for referencing purposes. As can be seen, two articles from 2013, one from 2015 and one from 2016 were not accessible. The articles were either downloaded with a file size of 0 kb or not downloaded at all. In an attempt to manually download these articles, the websites returned with a message “404 not found”, indicating that the article was no longer publicly accessible on the internet.

The following articles were removed because they were not journal articles:

Table 7: Incorrect type filter

Article title	Year of publication	Type
“Textual data clustering and cluster naming”	2013	Master’s thesis
“Semantic intelligence interfaces for ambient assisted living”	2013	PowerPoint presentation
“Topic extraction from Turkish News articles”	2013	Project plan report

Article title	Year of publication	Type
“Empirical study on term selection for patent classification”	2014	Poster
“Textual information extraction”	2014	PowerPoint presentation
“Textflo”	2014	User guide
“CSE – Mohammed Mussafer Hussain 03”	2014	Book
“Improving event extraction by discourse-guided and multi-faceted event recognition”	2014	PhD thesis
“Large-scale structured learning”	2014	PhD thesis
“Topic detection within public social networks”	2014	PhD thesis
“Unsupervised classification for main features extraction in natural disaster text sources”	2014	PhD thesis
“Improved search, evaluation and web search”	2015	PowerPoint presentation
“Automatic text summarisation using importance of sentences for email corpus”	2015	Master’s thesis
“MA Translating Popular Culture Mark sheet for dissertations – Option B (print) Translation with reflective”	2015	Dissertation
“T & F proof”	2015	Book
“صوصنلل يئاقلت صيخلت ةيبرعلا automatic Arabic text summarisation”	2016	Book
“Hindi to English machine translation”	2016	Master’s thesis
“Textual analysis of expert reports to increase knowledge of technological risks”	2017	PowerPoint presentation
“Topic modelling and clustering for analysis of road traffic accidents”	2017	Master’s thesis
“Tram – An approach for assigning bug reports using their metadata”	2017	Master’s thesis

Article title	Year of publication	Type
“TFf user guide”	2017	User guide
“Preface (RuleML+ RR 2017)”	2017	Preface document
“Ontology-driven computational processing for unstructured text”	2017	Symposium introduction
“Cette thèse a été dirigée par”	2017	Master’s thesis
“Automatically identifying key sentences in biomedical abstracts using semi-supervised learning”	2017	Bachelor’s research report
“A stylistic analysis of dystopia hopelessness and disturbance in George Orwell s 1984”	2017	Bachelor’s senior project

Table 7 has three columns, indicating the title of the article, the year of publication and the document type. The year of publication is provided as a point of reference. The “type” column displays what type of document was presented. Even though it was specified that the search engine should only return journal articles, there were still some resources online that indicated the incorrect type of document. Therefore, those documents were also collected and needed to be filtered. Even though some of these resources could possibly have contributed to the body of knowledge in terms of the topic, such as a PhD thesis, they were removed. Other examples were project plan reports and PowerPoint presentations that made no contribution to knowledge or were not journal articles.

The following articles were removed as they did not present the topic of Stemming Algorithms in the context of Text Analysis or Text Mining:

Table 8: Non-topic-related filter

Article title	Year of publication
“Quality assurance in EFL proficiency assessment in a tertiary educational context”	2013
“A study on socio-ecological cultural complex in urban milieu”	2013
“Petar Hr. Ilievski”	2014
“First version _v1_ of the integrated platform nand documentation”	2014
“Web 2.0 online communities or Bla-Bla Land”	2017

Table 8 has two columns: article title and year of publication. The year of publication is provided for reference purposes. These five articles were removed as they did not contribute to the research focus on Stemming Algorithms. The first paper focused on the topic of quality assurance in education. The second paper focused on socio-ecology and culture. The third paper was about an actual person called “Petar”. The fourth paper was on the architecture of railways. The last paper was on the topic of “Bla-Bla Land”. None of these contributed to Text Analysis or Stemming Algorithms. These papers came out in the search results as they used words similar to that of the current paper’s topic. For example, Drozdova, Utochkin and Kleppe (2017) used words such as “in order to avoid a bias stemming from potential differences” in their paper “Web 2.0: Online communities or Bla-Bla Land?”, which is used as a synonym for “originating from”. Even though Drozdova et al. (2017) applied a Text Analysis approach, there was no application or extension to Stemming Algorithms. The rest of the paper did not address the main topic of Stemming Algorithms.

The final collection of articles after the filtering criterion contained 317 articles.

Figure 30 gives a summary of the screening stage of the research.

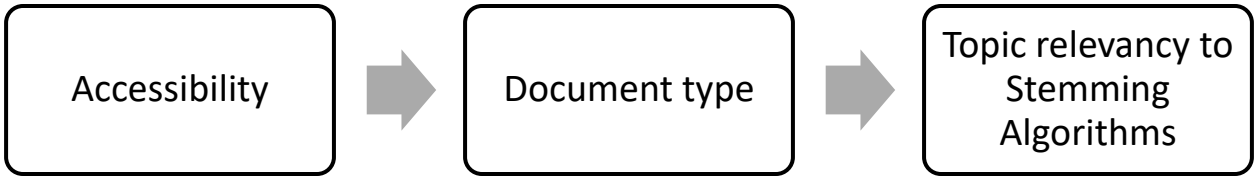


Figure 30: Summary of the screening stage

The screening stage concludes the selection stage of the research process. The last section discusses the extraction stage.

4.4.3 Extraction stage

The third stage of the process is the extraction stage. This stage comprises two steps: quality appraisal and data extraction.

The quality appraisal step determines which articles gathered in the previous step have the required quality. The quality was set by a defined list of requirements and not the personal opinion of the researcher. The requirements were adopted from Okoli and Schabram (2010), where applicable. The requirements were as follows:

- The study must be of an empirical nature, and the research methodology must be clearly defined. This step was assured during the selection stage with the help of Google Scholar. The purpose of this requirement was to ensure the establishment in knowledge and not a review or overview (Okoli & Schabram, 2010).
- The results of the study must be clearly scripted to ensure the establishment in knowledge.
- The data collection method should be indicated for future reproduction.
- The paper should be written in proper English, which the author must be able to follow. This has already been covered in the search.

- The conclusion from the collected article should reflect the results produced from the research and not from any other source.
- The study should provide evidence for any possible claims made to show that there is a proper contribution to knowledge and that it is not just a collection of different authors' references.

The data extraction step extracted information that was relevant to the current study. There are two possible methods of extracting data from the collected articles: qualitative and quantitative methods (Okoli & Schabram, 2010).

Qualitative analysis is the in-depth analysis of underlying meaning and motivation of individual components of the whole body of text. This method treats each individual component separately in the belief that these individual components have different underlying meanings and motivations (McCusker & Gunaydin, 2015).

Quantitative methods generalise a sample population of data by aggregating the findings into possible trends and patterns that can be gathered from all the data within the results set. This method treats each individual component of the whole as the same in the belief that these individuals have the same underlying meanings and motivations (McCusker & Gunaydin, 2015).

This research will follow a quantitative extraction method by taking a Text Analysis approach. The extraction of themes from the Text Analysis approach is not subjective in terms of any individual as it has been automated through the application of RapidMiner. The Text Analysis process was adapted from Aryal et al. (2015), whose research identified themes from health care information systems research. The author's research is similarly adapted according to themes from Text Analysis articles. The process went through data collection and pre-processing, and produced a word list as a result. Data collection has already been mentioned in WebHarvy so the data collection method was not followed according to Aryal et al. (2015). The outcome of their research

is different from the research conducted in this paper. The steps of aggregating concepts and reducing them to common research terms was adopted from Aryal et al. (2015). Aryal et al. (2015) focused on clusters of words, while this research uses linear regression to analyse the word list.

In the extraction step, the articles were saved in separate folders based on their year of publication. They were then imported into a Text Mining tool to gather an aggregative collection of themes. The chosen Text Mining tool was RapidMiner version 9.1.000 with Student Licence, which provided full functionality up to a year.

RapidMiner is an open source software available in January 2019, which is categorised as one of the top 20 most commonly used tools for Data Mining (Ristoski, Bizer & Paulheim, 2015). RapidMiner makes use of a drag and drop interface with different operators that perform a specific task. Operators can have sub-processes (Ertek et al., 2013). As mentioned by Gupta and Malhotra (2015), RapidMiner is a world-leading software package that comprises many functionalities and also allows third-party extensions if the default is not enough. Gupta and Malhotra (2015) also mention that RapidMiner plug-ins can perform all the required tasks for Text Mining at an academic research level, which makes it very suitable for this study. There are a lot of plug-ins associated with RapidMiner, which allow the user to download operators from third parties. This allows the flexibility of Text Mining with dynamic requirements to be implemented. For Text Mining, the plug-in for Text Mining needs to be installed (Ertek et al., 2013). The operators that are specific to the research are discussed below. For each year of publication, RapidMiner followed the Text Mining process outlined by the steps discussed in the literature chapter.

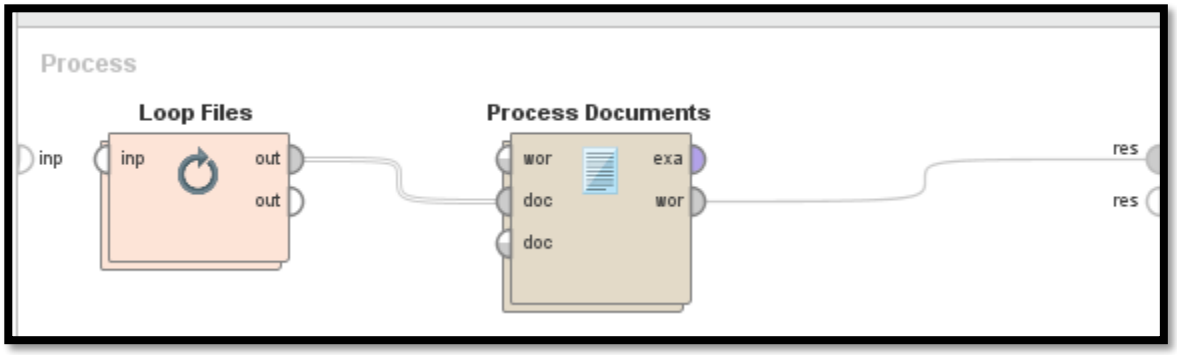


Figure 31: RapidMiner – main process

"Loop files" parameters	**"Process documents" parameters**
• File directory: Desktop\articles (this indicates where the article files could be located) • Filter type: glob (this indicates that there should not be a bias in terms of file type) • Filter by glob: * (this indicates that it can be of any file extension type) • Recursive: True (this indicates that the program should continue onto the next file once it is done with the previous one) • Enable macro: False (this indicates there are no extra macro requirements to complete this task) • Reuse results: False (this indicates that there is no need to reuse the results to complete this task)	• Vector creation: TF-IDF (this stands for term frequency-inverse document frequency, which indicates that the system should use the statistical method that represents how important words are to the documents themselves) • Add meta information: True (this collects additional information about the processing; this is true on default and has no influence on the study) • Keep text: False (this indicates that there is no requirement to keep the files in the program memory after the process has been completed) • Prune method: Absolute (the pruning method is a way to remove results; the absolute method is a way of removing records based on the actual value given) • Prune below: 2 (this clears all records where the outcome is less than two; this means that if the themes only appear in the articles once, they are removed) • Prune above: 1 000 (this clears all articles if they appear more than 1 000 times; this is an unreal value as there are no occurrences above 1 000; it has been set there for requirement purposes)

Figure 31 shows the operators that have been placed there to complete the task. There are two operators: loop files and process documents. The “loop files” operator runs through a directory of files and performs a task with specified instructions. The “process document” operator runs through the given output from reading each file and performs given instructions. These two operators both contain sub-processes that would indicate further instructions.

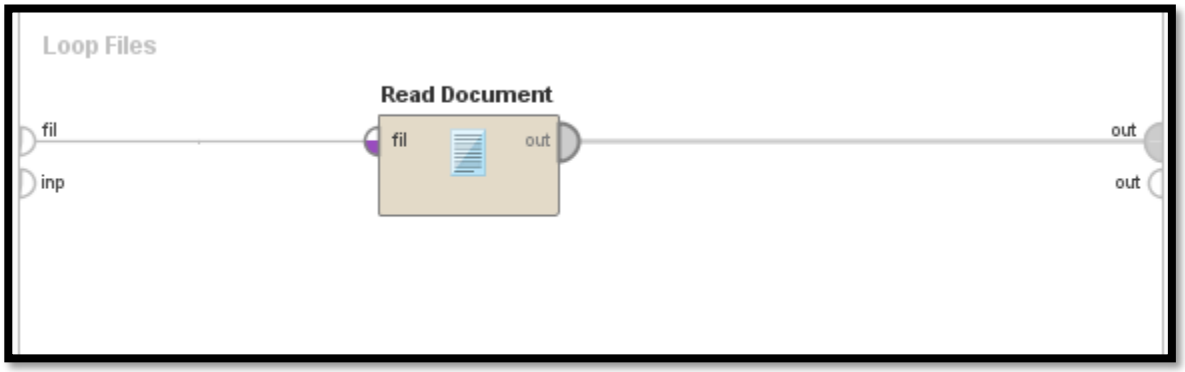


Figure 32: RapidMiner – loop files sub-processes

“Read document” parameter

- Content type: pdf (this indicates that the files that are being read are pdf files)

Figure 32 depicts the operators that have been placed within the “loop files” operator. “Read document” imports the data in the document into RapidMiner for processing purposes.

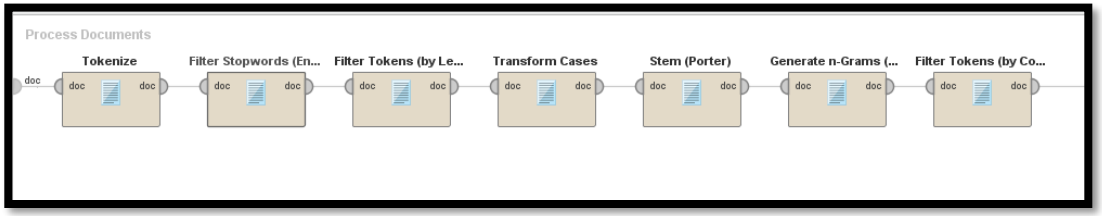


Figure 33: RapidMiner – process document sub-processes

“Tokenise” parameters	**“Filter stop words” (English)**	**“Filter token (by length)” parameters**	**“Transform cases” parameters**	**“Stem” (Porter)**	**“Generate n-grams (terms)” parameters**	**“Filter tokens (by content)” parameters**
• Mode: non-letters (this makes sure that the tokenisation is accomplished against and between no letters)	• None	• Minimum character: 4 (this makes sure that themes shorter than four letters are not included) • Maximum character: 25 (this make sure that themes longer than 25 letters are not included)	• Transform to: lower case (this makes sure that all letters in the text documents are in lower case for grouping purposes)	• None	• Maximum length: 2 (this makes sure that two words are combined to create word themes with more than one word)	• Condition: contains string: “_” (this allows RapidMiner to clear all themes that only have one word in them since the researcher wants to focus on themes that have more than one word)

Figure 33 depicts the operators that have been placed in the “process document” operator. “Tokenise” operator provides tags to each word in all sentences. The “filter stop words” operator removes all words that do not contribute to the final conceptual output of the Text Mining process. The “filter by token length” operator removes words above and below a certain character length. The “transform cases” operator turns all the letters in the document into either upper case or lower case. The “generate n-gram” operator searches for all possibilities of n-gram combinations within the document. The “filter tokens by content” operator selects cases based on a given condition. In this case, the researcher used the “filter token by content” operator to remove all words that are not an n-gram. The parameters for each operator are stipulated below.

After the set-up of all the RapidMiner operators, the only step left is to click the run button at the top of the screen. RapidMiner produces a table containing all possible combinations of themes (generated n-grams), the number of documents in which this theme can be found, and the number of times this theme appeared in all the documents. The resulting list of themes is created based on statistical formulations without prior knowledge of the topic. Therefore, the combination of themes can contain irrelevant content. To interpret the results and prioritise the themes, the author narrowed down the records using the Pareto Principle. The Pareto Principle states that 80% of the effective product originates from 20% of the cause (Goeminne & Mens, 2011).

This principle originated in the late 19th century with the evaluation of population wealth distribution. The underlying mathematical principles were later integrated into studies of various professions and proved to be true (Sanders, 1987). This principle seems to cover a wide range of subjects, including Text Analysis, where 20% of the words within a document represents 80% of the concepts (Zhu et al., 2015).

The research focused on 20% of the words from the results that contribute 80% of the knowledge. The researcher focused on the top 20% of total occurrences of themes (that existed in all six years).

These values were placed in a table that contained the themes that appeared in all the years, as well as the document occurrence and total occurrence values for each year. This table was imported into Microsoft Excel, and the duplicate theme columns were deleted. Finally, the column headings were renamed appropriately.

The results displayed a total word occurrence and a total document occurrence for each theme. Total word occurrence refers to the number of times a certain word appears throughout all the documents. Total document occurrence refers to the number of articles in which that word appears. For this research, the total word occurrence will be the same as the total theme occurrence.

4.4.4 Reporting stage

The fourth stage of the research process is the reporting stage. This stage comprises two steps: analysis of findings and writing the review.

The analysis of findings step analyses the results of the findings quantitatively. In traditional research methodology, there are two methods of analysis: qualitative and quantitative (Oates, 2005). The chances are high for other methods of analysis. However, since this research can be categorised into one of the two methods (quantitative vs qualitative), only these two methods will be discussed.

This research takes a quantitative approach as the data is not subjective in respect of any individual, and statistical formulae are used to interpret the research results. It does not treat each

individual component separately. Instead, it aggregates all the data according to common terminologies. Calculations were followed according to Kerlinger and Lee (2000), who discuss linear regression as a prediction calculation. Kerlinger and Lee (2000) argue that linear regression occurs when two variables are compared to each other, and a trend line is determined as the "mean" or "prediction" line. The "y" values are the predicted values. The following sections will explain how the author carried out the calculations described by Kerlinger and Lee (2000).

Three extra columns were added to the RapidMiner results. These were "total", "average" and "accumulated". The "total" column summed the total occurrence for each theme over the six-year period. The "average" column divided the calculated total for each theme by the grand total of the occurrence of all the themes. The "accumulated" column contains a formula that starts at the top and adds the average themes' values to the previous accumulated average value. The themes where the "accumulated" value reaches 20 will determine the cut-off point for the top 20% of themes according to the Pareto Principle.

Before any results were gathered, the data was first grouped together for readability purposes. Each year's results were placed in an Excel spreadsheet with the headings shown in. This was done for both values of "total occurrence" and "document occurrence".

Table 9: Results heading

Theme	2013	2014	2015	2016	2017	2018	Total	Standard deviation

indicates the eight headings provided in the Excel spreadsheet. These are the theme itself, the resulting value of the document occurrence and/or total occurrence value over the six years, the total value and the standard deviation. The total value is a sum of the results gathered for that theme over the six years. The standard deviation was calculated for each of the themes' values over the six years. This standard deviation was only used to sort the values for display purposes. The standard deviation is calculated with the STD.DEV formula in Excel. The values in the table are then sorted according to the standard deviation values from largest to smallest.

The first set of results begins by analysing the number of documents that contain a certain theme and how it changed over the six-year period. The number of documents that contain a theme indicates the number of research papers in which this theme appeared. The number of research articles that discuss a particular theme indicates the interest in that theme. An interest in a theme can either be the application of the theme or further development of the theme.

A few steps were followed to analyse how the themes changed over the years. The first step was to calculate the difference between each year and the previous year. Since the collected data did not contain articles in 2012, the calculation started with the difference between 2014 and 2013. If the value is positive, there is an increase, and if it is negative, there is a decrease. The "increase" and "decrease" indicators were accomplished for each year for each theme. This indicator provides a trend from one year to the next. The next step is to determine the turning points. The turning points indicate a change in direction between two indicators. For example, if all the indicators for a theme show "increase", it means that this theme did not have any turning points in the year. If a theme shows an increase twice followed by three decreases, it means that there was an increase from 2013 to 2015 and a decrease from 2015 to 2018. The turning point would be in 2015. If all the indicators alternated from the first to the last, this would represent an equilibrium. Table 10 shows all the possible combinations of trends that could be found.

Table 10: All possible combinations of trend outcomes

Analysis outcome	Description
Increasing	There was a constant increase in interest in this theme, which indicates a possible breakthrough in the topic.
Decreasing	There was a constant decrease in interest in this theme, which indicates a possible barrier to a breakthrough in this topic or the topic has been exhausted.
Equilibrium	There was an alternation between breakthrough and decline in research, which indicates that this theme is continuously being researched.
Low peak 2014	There was a decline in interest in the theme until 2014. From 2014, there was a breakthrough.
Low peak 2015	There was a decline in interest in the theme until 2015. From 2015, there was a breakthrough.
Low peak 2016	There was a decline in interest in the theme until 2016. From 2017, there was a breakthrough.
Low peak 2017	There was a decline in interest in the theme until 2017. From 2017, there was a breakthrough.
High peak 2014	There was an increase in interest in the theme until 2014. From 2014, there was a decline to research, and the interest in the theme declined.
High peak 2015	There was an increase in interest in the theme until 2015. From 2015, there was a decline to research, and the interest in the theme declined.
High peak 2016	There was an increase in interest in the theme until 2016. From 2016, there was a decline to research, and the interest in the theme declined.
High peak 2017	There was an increase in interest in the theme until 2017. From 2017, there was a decline to research, and the interest in the theme declined.
High peak 2014 Low peak 2015 High peak 2016	There was an increase in interest in the theme until 2014. There was a decline in research until 2015 and a breakthrough in research until 2016. Finally, there was a blockage in research on the theme.

Analysis outcome	Description
High peak 2014 Low peak 2015 High peak 2017	There was an increase in interest in the theme until 2014. There was a decline in research until 2015 and a breakthrough in research until 2017. Finally, there was a blockage in research on the theme.
High peak 2014 Low peak 2016	There was an increase in interest in the theme until 2014. There was a decline in research until 2016 and another breakthrough in research on the theme.
High peak 2014 Low peak 2016 High peak 2017	There was an increase in interest in the theme until 2014. There was a decline in research until 2016 and a breakthrough in research until 2017. Finally, there was a blockage in research on the theme.
High peak 2014 Low peak 2017	There was an increase in interest in the theme until 2014. There was a decline in research until 2017 and another breakthrough in research on the theme.
High peak 2015 Low peak 2016 High peak 2017	There was an increase in interest in the theme until 2015. There was a decline in research until 2016 and a breakthrough in research until 2017. Finally, there was a blockage in research on the theme.
High peak 2015 Low peak 2016	There was an increase in interest on the theme until 2015. There was a decline in research until 2016 and another breakthrough in research on the theme.
High peak 2015 Low peak 2017	There was an increase in interest in the theme until 2015. There was a decline in research until 2017 and another breakthrough in research on the theme.
High peak 2016 Low peak 2017	There was an increase in interest in the theme until 2016. There was a decline in research until 2017 and another breakthrough in research on the theme.
Low peak 2014 High peak 2015	There was a decline in research until 2014, followed by a breakthrough until 2015. Finally, there was a decline in interest from then onwards.
Low peak 2014 High peak 2015 Low peak 2016	There was a decline in research until 2014, followed by a breakthrough until 2015. There was another decline until a breakthrough was made in 2016.
Low peak 2014 High peak 2015 Low peak 2017	There was a decline in research until 2014, followed by a breakthrough until 2015. There was another decline until a breakthrough was made in 2017.

Analysis outcome	Description
Low peak 2014 High peak 2016	There was a decline in research until 2014, followed by a breakthrough until 2016. Finally, there was a decline in interest from then onwards.
Low peak 2014 High peak 2016 Low peak 2017	There was a decline in research until 2014, followed by a breakthrough until 2016. There was another decline until a breakthrough was made in 2017.
Low peak 2014 High peak 2017	There was a decline in research until 2014, followed by a breakthrough until 2017. Finally, there was a decline in interest from then onwards.
Low peak 2015 High peak 2016	There was a decline in research until 2015, followed by a breakthrough until 2016. Finally, there was a decline from then onwards.
Low peak 2015 High peak 2016 Low peak 2017	There was a decline in research until 2015, followed by a breakthrough until 2016. There was another decline until a breakthrough was made in 2017.
Low peak 2015 High peak 2017	There was a decline in research until 2015, followed by a breakthrough until 2017. Finally, there was a decline from then onwards.
Low peak 2016 High peak 2017	There was a decline in research until 2016, followed by a breakthrough until 2017. Finally, there was a decline from then onwards.

Table 10 has two columns. The first column shows all the possible trend outcomes from the analysis, while the second column shows the respective descriptions of each outcome. For example, the analysis outcome of “increasing” would mean that there is a constant increase in interest in the theme, which indicates a possible breakthrough in the topic. Similarly, the analysis outcome of a theme with “low peak 2016; high peak 2017” would mean that there was a decline in research until 2016, followed by a breakthrough until 2017. Finally, there was a decline in interest from then onwards.

In Table 10, one can see that “high peak 2013”, “high peak 2018”, “low peak 2013” and “low peak 2018” are not presented. The reason for these outcomes is because a high peak in 2013 would

be the same as constantly decreasing and a high peak in 2018 would be the same as constantly increasing. Respectively, a low peak in 2013 would mean constantly increasing, and a low peak in 2018 would mean constantly decreasing.

Even though this analysis already returns valuable results, the total occurrence values still need to be added to the interpretation. Total word occurrence for a theme in one document indicates the number of times a researcher mentions the theme. The more the researcher interacts with the theme, the more the research will mention it. To analyse how the total word occurrence over all documents influences the results, they have to be compared to word occurrence.

If the total word occurrence and the total document occurrence are both high, this means that it is a topic of great discussion throughout all the articles. If the total word occurrence is high, but the total document occurrence is low, that means that it is a hot topic for some of the articles. If the total word occurrence is low and the total document occurrence is high, it means that the topic is in all articles, but it is not a big focus anymore.

To analyse the results, the calculations and interpretations explained by Kerlinger and Lee (2000) were carried out where the correlation between two factors was compared to each other. This interpretation assisted with answering the questions set out in the problem statement.

To provide a numeric association in the analysis of how much a theme is in focus or out of focus, a linear regression calculation was carried out for each year. The x-axis is the total occurrence, and the y-axis is the document occurrence. The distance from the trend line was calculated. Figure 34 shows an example output of what a linear regression would look like.

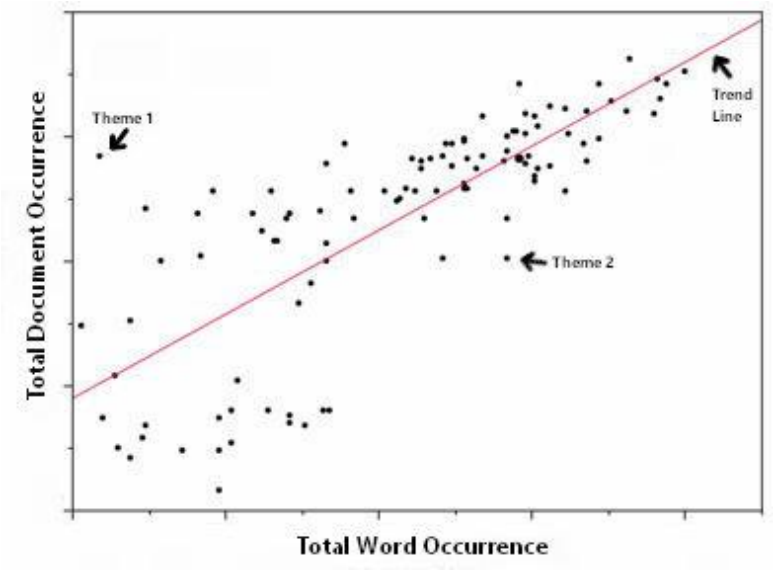


Figure 34: Example of a linear regression

Figure 34 presents a graph comparing “total word occurrence” with “total document occurrence”. “Total word occurrence” is placed on the x-axis and “total document occurrence” is placed in the y-axis. This graph is presented for the purpose of explaining the concept of linear regression. Therefore, the unit labels are not present in detail. Data points can be observed on the graph. These data points can be referenced as the output from RapidMiner. There are three labels on the graph: “trend line”, “Theme 1” and “Theme 2”. The trend line indicates the average or moving trend of all the data. It indicates the line of best fit for all the data. It also indicates a probable moving direction or the collective data points. Data points that lie on the trend line indicate that the outcome occurrence for that theme is expected. The label “Theme 1” references a data point above the trend line. Theme 2 references a data point below the trend line. Theme 1 represents a data point that has a low total occurrence, but a high document occurrence. Therefore, Theme 1 is a global concept that is discussed in many research papers, but is not the main focus of the research. Similar to Theme 1, all data points above the trend line would have the same conclusion. Theme 2 would result in the opposite conclusion, where the document occurrence is

low, but the total occurrence is high. Theme 2 indicates a theme that is a focus area. Similarly, the same conclusion is reached for data points below the trend line. The trend line value is calculated from the trend line equation, which is as follows:

$$Y = a + bX$$

where “Y” is the Y-axis point, “a” is the Y-intercept (where the first X data point would cross the Y axis line), “b” is the slope and “X” is the X-axis point. The “a” and “b” values are calculated using the following formula:

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Table 11 was created to establish the “a” and “b” values for the trend line formula. This table was established for each year. The table has two columns. The first column (column H) indicates the required values to accomplish the formula. The second column (column I) indicates the formula used in Excel to derive the required values. The first row shows the formula used to calculate the total value of all “x” values of the respective year. The second row shows the calculations done for the total of all “y” values. The third row shows the total addition calculation after the product of “x” and “y” has been calculated for each respective theme. The fourth row indicates the calculation for the total addition of all “x” values squared. The fifth column indicates the total values in the whole year. The sixth and seventh rows indicate the calculation for both “a” and “b”. The last row displays the calculation to show the trend line formula. The formula in the last row is then used to determine the distance from the trend line.

Each year's data is placed in an Excel spreadsheet. There are six columns ranging from A to F. Column A indicates the theme that was taken from the results. Column B indicates the total occurrence of the theme for the year. Column C indicates the document occurrence of the theme for the year. Column D has the heading “XY”, which indicates the multiplication of column B and column C. Column E has the heading “X^2”, which indicates the multiplication of column C by itself. Column F has the heading “distance”, which indicates the final calculation of the distance from the trend value for the respective theme. In the spreadsheet, there is also a column H and a column I, which have the breakdown values to calculate the trend line formula. Each cell represents a section to the formula which then build up to the final trend formula at the bottom. Column A to Column F follows the structure of APPENDIX 5-E.

Table 11: Calculations of the “a” and “b” values

Column H	**Column I**
SUM x	=SUM(B2:B95)
SUM Y	=SUM(C2:C95)
SUM XY	=SUM(D2:D95)
SUM X^2	=SUM(E2:E95)
n	=COUNT(B2:B95)
a	=(I3*I5-I2*I4)/(I6*I5-POWER(I2,2))
b	=(I6*I4-I2*I3)/(I6*I5-POWER(I2,2))
Trend formula	="Y = " & I8&" + " & I9&"X"

The trend line indicates the direction in which the data should be moving. Data points on the trend line indicate an expected trend and are not out of the ordinary. The data in which one is interested is the data that is out of trend: that above or below the trend line. The data points above the trend line indicate a large document count, but a small total occurrence. This means that the data point

indicates a global concept that is not in focus. A data point below the trend line indicates a small document count, but a large amount total occurrence. This means that that topic is in focus.

To find out to what extent the data points are either in focus or out of focus, a calculation needs to be done on each data point to see how far it is from the trend line. The calculation is done by taking the Y point of the data point and subtracting it from the Y point of the trend line at the same X point. The Y point of the trend line is calculated with the formula established in one of the previous steps. The distance from the trend line value is calculated by subtracting the trend line on the respective x value (total occurrence) from the actual y value (document occurrence) of the respective x value (total occurrence).

This would return either a positive or a negative value for each theme. The positive value means that the value is in focus where a small number of articles are engaging in much discussion about the theme. A negative value indicates that many articles are discussing the theme, but are not discussing much about the theme.

There are two factors of analysis related to document occurrence. The first is either the increase in document occurrence over the years or the decrease in document occurrence over the years. This indicates either a growing interest in the theme or a decreasing interest in the theme over the years. The second factor of analysis is the actual number of document occurrences. If there was a high document occurrence for the theme, then the analysis shows that the theme is widely studied; otherwise, it would be the opposite. Document occurrence shows the interest of the authors. Total occurrence shows the intensity of the concept. If all documents have a theme, but the theme only appears once in each article, it means that the theme is an old concept or a small section of the study. Thus, the theme has a low intensity of focus in all those articles. It can also

be interpreted as the theme being an old concept that existed before and is only being used in the article as some sort of application.

The analysis of total occurrence is slightly different to that of document occurrence. The number of times a theme appears in all the articles denotes the intensity of the theme relative to the research. The more the theme is mentioned in all the collected articles of that year, the more that theme is involved in research. Therefore, if the total occurrence is high, it is interpreted that the theme is highly researched. The opposite is true for low total occurrence. An increase in total occurrence over time denotes an increase in the intensity of research over time. A decrease in total occurrence over time denotes a decrease in the intensity of research over time.

One needs to interpret the mixture of the two. If there is a high document count, but a low total occurrence count, interprets that many papers discuss this concept, but it is not a research focus. If there are a few papers that constantly discuss one theme, it means that there is a high research focus on that theme, but it is not a widely established field among all researchers.

The interpretation of the distance from the trend line values is made for each theme over the six-year period. If all the values are above zero, it is interpreted that the theme is in focus over the six-year period. If all the values are below zero, it is interpreted that the theme is a global concept or has just been borrowed.

The writing the review step puts all the above stages into the physical paper. This includes capturing the systematic review process and the results. These results are presented in the following chapters.

4.5 SUMMARY AND CONCLUSION

In this chapter, the process and method for the execution of the research have been discussed. This includes a discussion of the articles that have been accessed, the process of analysis and the method of interpretation of the articles to ultimately answer the research question: *"What are the advances of Stemming Algorithms in the application of Text Analytics between 2013 and 2018?"*. This ensures that the research is carried out in an orderly fashion, that the steps taken are transparent, and that the articles used allow the research to be repeated (Okoli and Schabram, 2010).

This chapter discusses how the study follows a systematic review. This was based on the methodology presented by Okoli and Schabram (2010), accompanied by the knowledge and understanding of Attride-Stirling (2001). Within the methodology proposed by Okoli and Schabram (2010), there are four main stages: planning, selection, extraction and reporting.

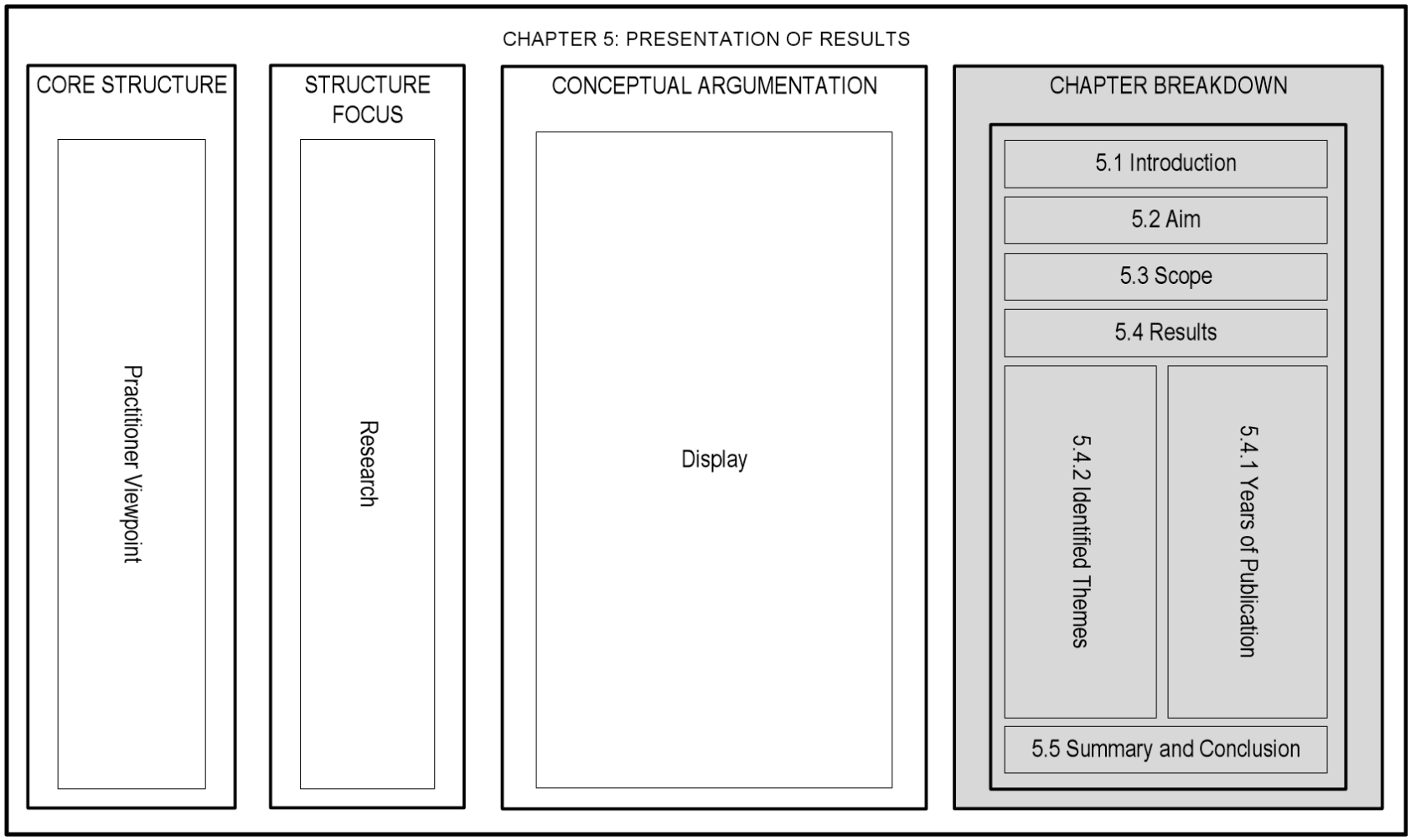
To fulfil the steps required by Okoli and Schabram (2010), Google Scholar was chosen as the platform to collect data. The following specifications were provided on the databases: specify only empirical research (journal articles); specify only pdf files; specify articles written in English; use specific terms relating to the theme of the topic, such as Stemming Algorithms, stemmer, Text Mining and Text Analysis.

WebHarvy was used to identify each Google Scholar result and download it. A Text Analysis process was performed on the downloaded articles to determine themes.

The Text Analysis process was adapted from Aryal et al. (2015), whose research identified themes from healthcare information systems research. The process included data collection, pre-processing and the production of a word list. This data collection differed from other research by

using WebHarvy for the presentation of the results. The data preparation and presentation were adopted from Van Deventer (2013). Finally, the calculations and interpretation of the results were described by Kerlinger and Lee (2000),

This chapter therefore discussed how to analyse the results. The calculations and interpretations explained by Kerlinger and Lee (2000) were carried out where the correlation between two factors was compared to each other. This interpretation then assisted in answering the questions set out in the problem statement. The following chapter will display the results gathered by following the aforementioned process, based on the detailed methods, tasks and activities that are presented.



Chapter Map 5: Presentation of results

CHAPTER 5:
PRESENTATION OF RESULTS

5.1 INTRODUCTION

The research that was conducted for this study assists in answering the main research question: *What are the advances of Stemming Algorithms in Text Analysis over the past six years (2013 to 2018)?* In this chapter, the results are presented, as obtained from the process as detailed in Chapter 4. The research that was conducted produced results that pertain to and answer the main question, as well as the sub-questions set in Chapter 1. To reach a conclusion based on the research, it is necessary to first present the results, as obtained in RapidMiner. The presentation of the results in this chapter forms part of the requirements of this research.

5.2 AIM OF THE CHAPTER

This chapter aims to present the results from the research process as stipulated in Chapter 4. The results are presented in tables and graphs. The product of the research is presented by following the steps of the stipulated research methodology. An interpretation of the results will be presented in Chapter 6.

5.3 SCOPE OF THE CHAPTER

To achieve the aim of the research, the data is first grouped according to the total number of articles gathered for each year. The themes identified for each year are then discussed, followed by a summary and the conclusions that can be drawn from these findings.

5.4 RESULTS

Based on the research question, the data is presented according to years of publication to determine the trends for each year. The total number of articles established in the research is generally presented in the form of bar graphs. This will be followed by the RapidMiner results presented in tabular format, and a discussion of the details. This is done to identify changes in research interest over the six-year period.

5.4.1 Years of publication

The total number of articles found for each year was roughly constant with an average of 57.5 articles per year. The most publications were found in 2014. Figure 35 provides a bar graph to indicate the number of articles gathered per year.

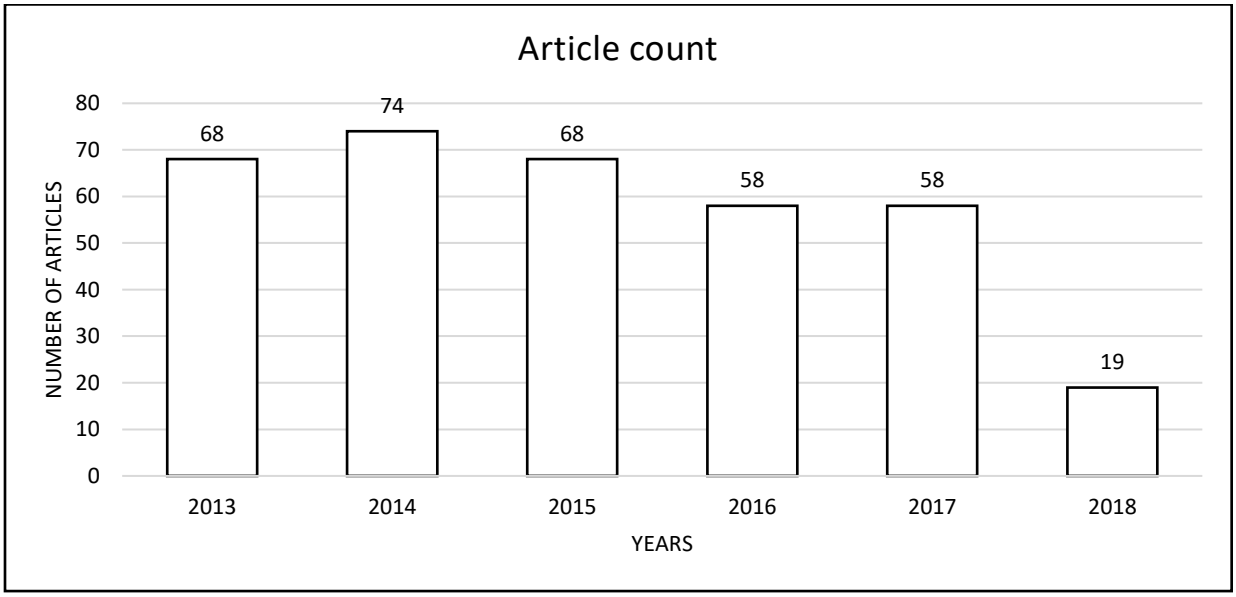


Figure 35: Number of articles per year

As can be seen in Figure 35, 68 articles were collected in 2013; 74 articles were collected in 2014; 68 articles were collected in 2015; 58 articles were collected in 2016; 58 articles were collected

in 2017; and 19 articles were collected in 2018. The last year therefore had the lowest number of articles. A clear trend can thus be seen in the chart of the collected articles. After the spike in articles in 2014, the collected articles started to decrease every year. The last year on the chart shows a drastic drop in number from 58 to 19 articles. Excluding 2018, the average of the collected articles would have been 65.2. However, taking 2018 into account, the average drops to 57.5.

5.4.2 Identified themes

The terms identified in RapidMiner were considered to be the identified themes. This sub-section displays the results gathered in RapidMiner and imported into an Excel spreadsheet. The resulting themes were obtained by combining two words (terms) with an underscore character. Before the results were displayed, some themes were removed as they did not contribute to knowledge on the topic. These themes were either academic journal references or ways of writing academic journals. Some examples would be “results obtained”, “conference proceedings”, “IEEE journal”, “shown below” and “information technology”. These themes did not contribute to the overall knowledge on the topic of Stemming Algorithms, and skewed the results. These themes were academic references found in the headers, footers and citations of articles. Upon closer examination, it was found that the terms “information technology” and “conference proceedings” in the articles were mostly used in citations. For example, the article “Design and evaluation of a parallel classifier for large-scale Arabic text” had 14 references that contained the theme “conference proceedings” to indicate the origin of the sources. The theme “conference proceedings” does not contribute to this study. Another example of a removed theme is the term “shown below”. This theme indicates a discussion around a provided example, figure, table or formula presented in an article. For example, the paper “Best treatment identification for disease using machine learning approach in relation to short text” stated that “a dictionary is constructed

as shown below", which shows that the theme "shown below" does not contribute to knowledge on this research topic. The examples given are academic terms used in research. The abovementioned example does not contribute to Stemming Algorithms in any way. It is merely a term used when writing up research. These themes are captured in Appendix 5-B.

The full set of results obtained after removing all the irrelevant themes is presented in Appendix 5-A, with three additional columns to determine the top 20% of themes. The table in Appendix 5-A is presented in the form as discussed in Chapter 4. There are 43 themes in the top 20% of total theme occurrences, with the lowest "total" theme occurrence over the six-year period being "Arab language", with only 274 "total" occurrences. The theme with the highest "total" occurrence is "machine learning" with 1 272 "total" occurrences.

The results are further divided into two separate tables. One table only contains the document occurrences, while and the other displays the total word occurrences. APPENDIX 5-B represents the results gathered from the research methodology process. The words that represent the themes are listed in the first column. The next five columns indicate the number of documents in which the theme appeared. Each of the five columns represents a year between 2013 and 2018. The second last column indicates the total number of document occurrences for that theme over the six-year period. The last column indicates the standard deviation of all the document occurrences over the six-year period. The results were then sorted based on the standard deviation, beginning with the largest standard deviation and ending with the smallest standard deviation. The sorting was done as a means of organising the data in a presentable manner.

Following an evaluation of the research of Altman and Saunders (1997) on credit risk management, it was determined that standard deviation is a statistical formula used to indicate how much each datum within the data set differs from the mean. A mean is a form of average.

With this indication, it is possible to see how far apart these data are from each other, which indicates a rate of change over the six-year period.

Within the themes, one can see that the words do not have suffixes. This is because of the stemming that took place in RapidMiner to complete the process. An example of this is "text_classif*",* which should read "text classification" or "text classifier". An example of all possible combinations of "text classif" can be found in APPENDIX 5-C. One can see that there are over 5 000 possible variations of the theme that can be the origination of the stemmed theme. The researcher will not explain each theme and all the possible suffixes that are associated with those themes in detail. However, APPENDIX 5-D contains a list of the most plausible originations of each resulting theme based on examples taken from the collected dataset. The table in APPENDIX 5-D has two columns: "stemmed theme" and "plausible original theme". Note should be taken of these changes throughout the results.

Table 12 presents a small snapshot of values from the table in APPENDIX 5-E for explanation purposes.

Table 12: Document occurrences of themes

Theme	2013	2014	2015	2016	2017	2018	Total	Std.dev
text_mine	53	50	52	45	40	12	240	5.43
natur_languag	46	49	51	40	35	10	221	6.61
inform_retriev	49	48	42	38	33	8	210	6.75
languag_process	37	43	46	41	33	8	200	5.1
machin_learn	40	40	40	39	37	15	196	1.3
data_mine	38	38	41	32	31	7	180	4.3
text_document	33	36	34	27	23	9	153	5.41
word_remov	28	28	28	31	20	10	135	4.12
naiv_bay	25	28	24	20	33	5	130	4.85
term_frequenc	21	33	28	25	22	8	129	4.87
word_stem	24	28	29	26	22	4	129	2.86
text_classif	28	34	18	16	29	4	125	7.68

Theme	2013	2014	2015	2016	2017	2018	Total	Std.dev
document_frequenc	24	31	25	21	21	7	122	4.1
artifici_intellig	23	25	23	21	19	5	111	2.28
precis_recal	26	25	19	22	16	7	108	4.16
support_vector	24	23	15	20	26	6	108	4.28
train_data	19	24	17	22	25	6	107	3.36
text_process	21	19	23	23	21	2	107	1.67
text_analysi	19	22	25	22	18	8	106	2.77

Table 12 has nine columns that reflect the same columns as the table in APPENDIX 5-E. The first column presents the theme, while the next six columns indicate the number of documents in which the theme appeared. The second-last column indicates the sum of the themes' value over the six-year period. The last column indicates the standard deviation for the theme over the six-year period. In this table, the theme with the highest standard deviation is "information retrieval", and the theme with the lowest standard deviation is "machine learning".

The table in APPENDIX 5-E contains the highest standard deviation of 15.48 for the theme "text mine". The lowest standard deviation is 0 (zero) for the theme "apache core". The theme with the lowest total document occurrence is "apache core" with five document occurrences. The theme with the highest total document occurrence is "text mine*"* with 240 document occurrences.

Table 13 displays the quartile values for all the document occurrences over the six-year period. It also includes the quartile values for the standard deviation. The minimum document occurrence value in 2013 is 1. The first quartile, which represents the 25th percentile value in 2013, is 2. The second quartile or 50th percentile value in 2013 is 3. The third quartile or 75th percentile value in 2013 is 6, and the maximum document occurrence in 2013 is 53. The same can be seen for the standard deviation over the six years. The minimum standard deviation value over the six years is 0.00. The first quartile, which represents the 25th percentile value, is 1.17. The second quartile or 50th percentile value is 1.72. The third quartile or 75th percentile value is 2.61, and the maximum value is 15.48.

Table 13: Quartile values for document occurrence

	2013	2014	2015	2016	2017	2018	STD.DEV
Minimum	1.00	1.00	1.00	1.00	1.00	1.00	0.00
First quartile	2.00	2.00	2.00	2.00	2.00	1.00	1.17
Mean	3.00	4.00	3.00	4.00	4.00	1.00	1.72
Third quartile	6.00	6.00	5.00	6.00	6.00	2.00	2.61
Maximum	53.00	50.00	52.00	45.00	40.00	15.00	15.48

The table in APPENDIX 5-F represents another section of results gathered from the research. In the first column are the words that represent the themes. The next five columns indicate the number of times that this theme appeared across all the documents combined. Each of the five columns represents a year between 2013 and 2018. The second-last column indicates the total, which is the sum of the total occurrences of that theme over the six-year period. The last column indicates the standard deviation of all the document occurrences over the six-year period. The results were then sorted based on the standard deviation, beginning with the largest standard deviation and ending with the smallest standard deviation.

Table 14 presents a small snapshot of values taken from the table in APPENDIX 5-E for explanation purposes. The format of Table 14 will follow that of Table 12. The only difference is that the values presented over the six years will now be the total occurrence values over all the gathered documents.

Table 14: Total occurrence of themes

Theme	2013	2014	2015	2016	2017	2018	Total	Std.dev
inform_retriev	287	214	174	238	80	31	993	97.66
machin_learn	200	239	165	306	304	58	1214	93.91
text_mine	222	242	306	174	160	32	1104	93.06
sentiment_analysi	41	138	80	77	257	15	593	86.85
neural_network	91	103	66	46	256	12	562	85.01
natur_languag	185	265	232	205	163	33	1050	80.59
naiv_bay	103	115	138	111	270	30	737	78.68

Theme	2013	2014	2015	2016	2017	2018	Total	Std.dev
data_mine	149	179	189	122	208	17	847	69.26
part_speech	52	109	73	70	204	2	508	67.95
featur_select	144	84	139	115	198	6	680	65.03
social_media	33	171	81	63	176	45	524	63.09
name_entiti	26	62	86	172	18	8	364	61.36
text_classif	175	181	128	111	148	22	743	58.21
stop_word	139	142	109	216	181	49	787	57.8
text_categor	152	115	96	49	39	11	451	52.94

In Table 14, the theme with the highest standard deviation is “information retrieval”, and the theme with the lowest standard deviation is “text categorisation”.

The table in APPENDIX 5-F has the highest standard deviation of 97.66 for the theme “information retrieval”. The lowest standard deviation is 0 for the theme “apache core”. The theme with the lowest total occurrence is “apache core” with 0 (zero) appearances. The theme with the highest total appearance across all documents is “machine learning” with 1 214 appearances.

Table 15 displays the quartile values for all the total occurrences over the six-year period. It also includes the quartile values for the standard deviation. The minimum value of the total occurrences in 2013 is 1. The first quartile, which represents the 25th percentile value in 2013, is 2. The second quartile or 50th percentile value in 2013 is 5. The third quartile or 75th percentile value in 2013 is 9, and the maximum total occurrences in 2013 is 287. The same can be seen for the standard deviation over the six-year period. The minimum value of the standard deviation over the six years is 0. The first quartile, which represents the 25th percentile value, is 1.87. The second quartile or 50th percentile value is 3.37. The third quartile or 75th percentile value is 6.35, and the maximum value is 97.66.

Table 15: Quartile values for total occurrences

	2013	2014	2015	2016	2017	2018	STD.DEV
Minimum	1.00	1.00	1.00	1.00	1.00	1.00	0.00

	2013	2014	2015	2016	2017	2018	STD.DEV
First quartile	2.00	2.00	2.00	3.00	2.00	1.00	1.87
Mean	5.00	5.00	4.00	5.00	5.00	2.00	3.37
Third quartile	9.00	11.00	8.00	11.00	11.00	3.00	6.35
Maximum	287.00	265.00	306.00	306.00	304.00	76.00	97.66

Figure 37 displays a graph that represents the top 20% of document occurrences of the themes sorted by the grand total of the themes' total occurrences over the six-year period. For the purposes of the research, only the document occurrence graphs are shown due to the fact that the total occurrence values are interpreted with the linear regression analysis, whereas the document occurrence values can show direct value in terms of interest from researchers. Figure 37 has 43 themes. The bars for each theme are presented chronologically, with the first bar representing 2013, and the last bar representing 2018. The x-axis displays the themes in descending order of standard deviation. The y-axis displays the total and the document occurrences respectively.

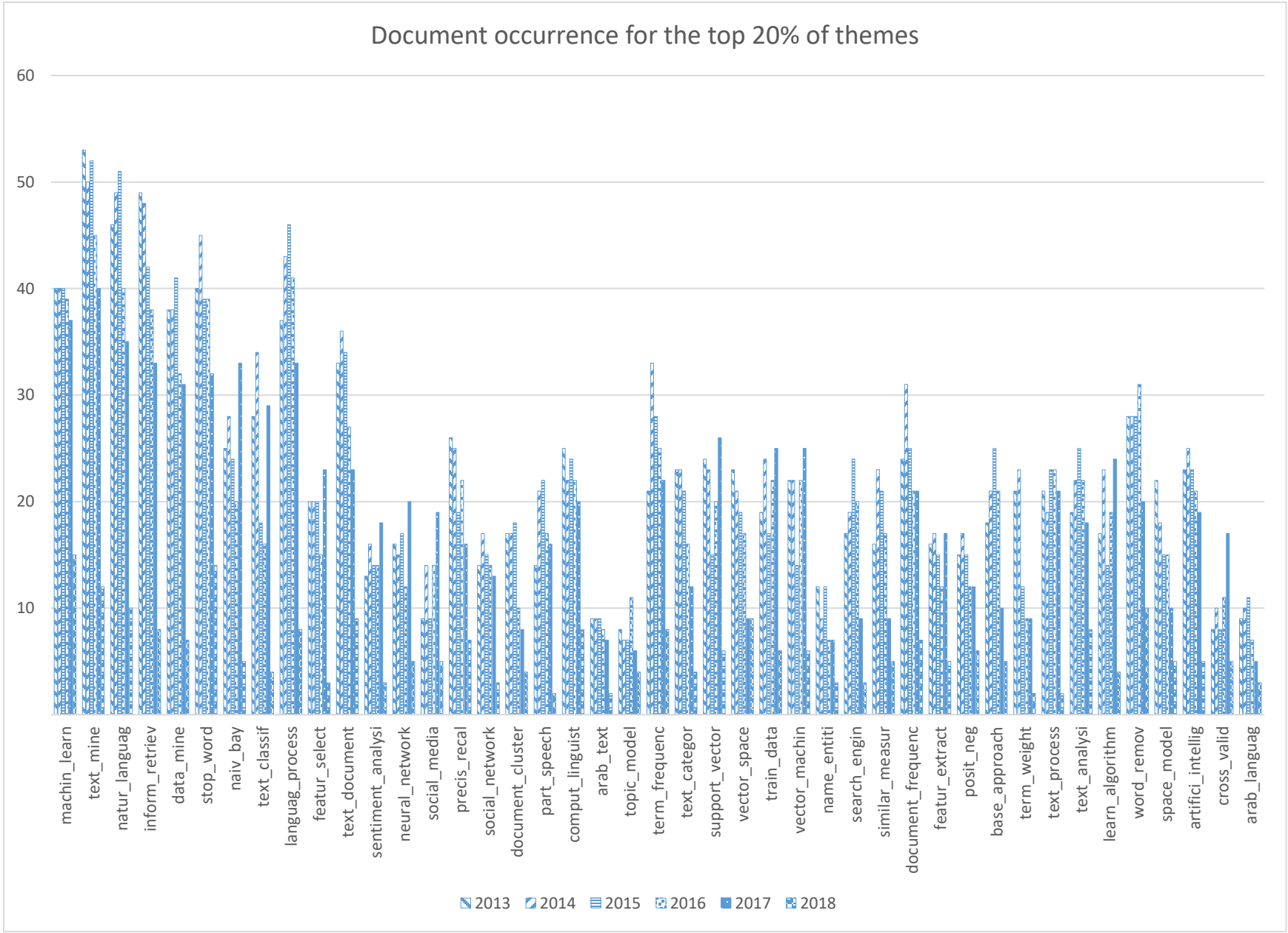


Figure 4: Document occurrence for top 20% of themes

In Figure 37, one can see that there is a trend for each theme over the six-year period. To provide an example, the theme “Arab language” increases until 2015 when it decreases again until 2018. One can, therefore, say that there is a high peak in 2015 for the theme “Arab language”. Table 16 illustrates the trends for each of the top 20% of themes.

Table 16: Document occurrence trend interpretation for the top 20% of themes

Theme	Interpretation
machin_learn	Decreasing
text_mine	Low peak 2014; high peak 2015
natur_languag	High peak 2015
inform_retriev	Decreasing
data_mine	Low peak 2014; high peak 2015
stop_word	High peak 2014
naiv_bay	High peak 2014; low peak 2016; high peak 2017
text_classif	High peak 2014; low peak 2016; high peak 2017
languag_process	High peak 2015
featur_select	Low peak 2016; high peak 2017
text_document	High peak 2014
sentiment_analysi	High peak 2014; low peek 2016; high peak 2017
neural_network	Equilibrium
social_media	High peak 2014; low peak 2015; high peak 2017
precis_recal	Low peak 2015; high peak 2016
social_network	High peak 2014
document_cluster	Low peak 2014; high peak 2015
part_speech	High peak 2015
comput_linguist	Low peak 2014; high peak 2015
arab_text	Decreasing
topic_model	Low peak 2015; high peak 2016
term_frequenc	High peak 2014
text_categor	Decreasing
support_vector	Low peak 2015; high peak 2017
vector_space	Decreasing
train_data	High peak 2014; low peak 2015; high peak 2017
vector_machin	Low peak 2015; high peak 2017
name_entiti	Low peak 2014; high peak 2015
search_engin	High peak 2015
similar_measur	High peak 2014

Theme	Interpretation
document_frequenc	High peak 2014
featur_extract	High peak 2014; low peak 2016; high peak 2017
posit_neg	High peak 2014
base_approach	High peak 2015
term_weight	High peak 2014
text_process	Low peak 2014; high peak 2015
text_analysi	High peak 2015
learn_algorithm	High peak 2014; low peak 2015; high peak 2017
word_remov	Low peak 2015; high peak 2016
space_model	Decreasing
artifici_intellig	High peak 2014
cross_valid	High peak 2014; low peak 2015; high peak 2017
arab_languag	High peak 2015

Table 16 has two columns. The first column is the theme and the second column is the interpretation of the theme. The interpretation column is only used to discuss the trend that can be found for the theme in Figure 37. A full description of each possible interpretation can be found in Chapter 4. Other than the interpretations found in Table 16, the rest of the interpretations for all the themes in the study can be found in APPENDIX 5-G.

To continue with the rest of the results, the trend line calculations need to be presented. Appendix 5-H displays the distance from the trend calculations for 2013. There are two tables in the appendix. The first table shows the calculations that are required to determine the trend line formula. The second table contains the theme, total occurrence, document occurrence, “X * Y”, “X^2” and distance from trend calculations. Appendix 5-I, Appendix 5-J, Appendix 5-K, Appendix 5-L and Appendix 5-M follow the same format, but contain values for 2014, 2015, 2016, 2017 and 2018 respectively.

APPENDIX 5-N contains a table with all the distances from the trend calculations combined into one table. The format for this table is discussed in Chapter 4. A positive value indicates that the research theme is in focus, and a negative value indicates that the research theme is out of focus.

Taking, for example, the theme “machine learning”, one can see that it was in focus for five years and out of focus for one year. The full interpretation of the trend for each theme is attached in APPENDIX 5-O. The table in APPENDIX 5-O has nine columns. The first column indicates the theme. The second column indicates the trend interpretation, which correlates with the same interpretations as the document occurrence interpretations. The next six columns indicate if the theme was in focus for the given year or not. The last column indicates the number of years that the theme was in focus.

Figure 38 displays the top 20% of values in a bar graph. In the graph, the y-axis indicates the distance from the trend value and the x-axis indicates the theme itself. When the theme is out of focus, the bar will appear below the x-axis. To fully identify the trend for each theme presented in APPENDIX 5-N, a sample has been extracted Table 17 from APPENDIX 5-O to discuss each theme in a summarised format. Table 17 follows the same structure as the table in APPENDIX 5-O. There are more themes that are in focus in relation to themes that are out of focus. One can also see that “word removal”, “learning algorithm” and “artificial intelligence” are the only themes that were out of focus for all six years. One can also see that “Naive Bayes”, “Arab text”, “document cluster” and “topic model” are themes that were in focus for all six years. “Machine learning”, “text classifier”, “feature selection”, “sentiment analysis”, “social media”, “social network” and “Arab language” were in focus for five of the six years, while *“*machine learning”, “feature selection” and “Arab language” only dropped out of focus during 2018. “Stop word”, “language processing”, “document frequency”, “Text Processing”, “learning algorithm”, “word removal” and “artificial intelligence” were out of focus for all six years. One can also see that “Arab text”, “document cluster”, “Naïve Bayes”, “topic model” and “training data” were in equilibrium for all six years. None of the themes were constantly increasing or constantly decreasing.

Over the six-year period, 21 of the themes were in focus in 2013, 23 of the themes were in focus in 2014, 19 of the themes were in focus in 2015, 22 of the themes were in focus in 2016, 18 of the themes were in focus in 2017 and 10 of the themes were in focus in 2018. One can see a clear drop in focus for a lot of the themes in 2018.

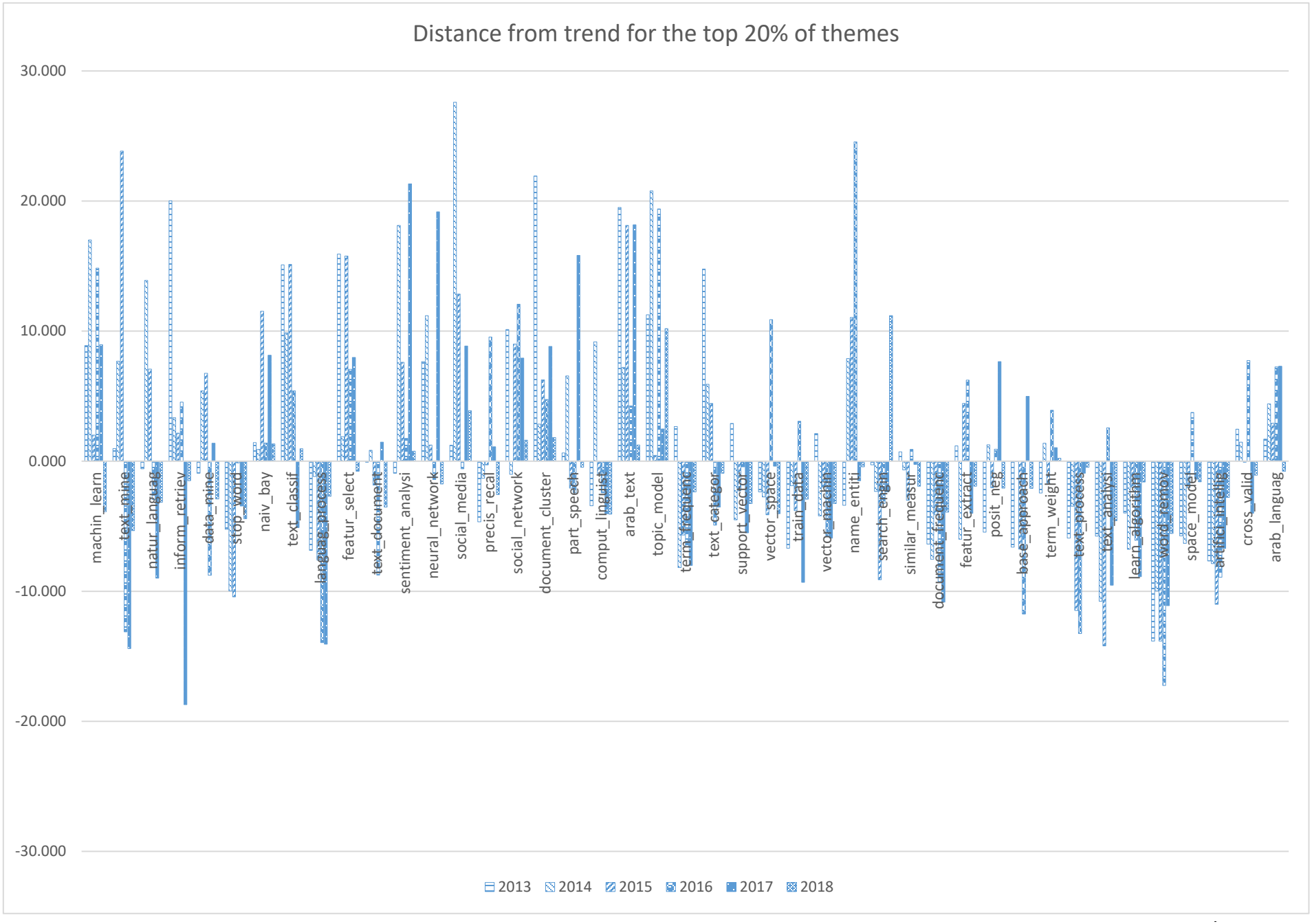


Figure 11: Distance from the trend values for top 20% of themes

Table 17: Trend interpretation for the top 20% of themes

Theme	Interpretation	Focus						
		2013	2014	2015	2016	2017	2018	Count
machin_learn	High peak 2014; low peak 2015; high peak 2016	True	True	True	True	True	False	5
text_mine	High peak 2015; low peak 2017	True	True	True	False	False	False	3
natur_languag	High peak 2014; low peak 2017	False	True	True	False	False	False	2
inform_retriev	Low peak 2015; high peak 2016; low peak 2017	True	True	True	True	False	False	4
data_mine	High peak 2015; low peak 2016; high peak 2017	False	True	True	False	True	False	3
stop_word	Low peak 2015; high peak 2016	False	False	False	False	False	False	0
naiv_bay	Equilibrium	True	True	True	True	True	True	6
text_classif	Low peak 2014; high peak 2015; low peak 2017	True	True	True	True	False	True	5
languag_process	High peak 2014; low peak 2017	False	False	False	False	False	False	0
featur_select	Low peak 2014; high peak 2015	True	True	True	True	True	False	5
text_document	Low peak 2016; high peak 2017	False	True	False	False	True	False	2
sentiment_analysi	High peak 2014; low peak 2016; high peak 2017	False	True	True	True	True	True	5
neural_network	High peak 2014; low peak 2016; high peak 2017	True	True	True	False	True	False	4
social_media	High peak 2014; low peak 2016; high peak 2017	True	True	True	False	True	True	5
precis_recal	High peak 2016	False	False	False	True	True	False	2
social_network	Low peak 2014; high peak 2016	True	False	True	True	True	True	5
document_cluster	Equilibrium	True	True	True	True	True	True	6
part_speech	High peak 2014; low peak 2016; high peak 2017	True	True	False	False	True	False	3
comput_linguist	High peak 2014	False	True	False	False	False	False	1
arab_text	Equilibrium	True	True	True	True	True	True	6
topic_model	Equilibrium	True	True	True	True	True	True	6
term_frequenc	Low peak 2014; high peak 2015; low peak 2017	True	False	False	False	False	False	1
text_categor	Low peak 2016	True	True	True	False	False	False	3
support_vector	Low peak 2014; high peak 2016; low peak 2017	True	False	False	False	False	False	1
vector_space	Low peak 2015; high peak 2016	False	False	False	True	False	False	1

Theme	Interpretation	Focus						
		2013	2014	2015	2016	2017	2018	Count
train_data	Equilibrium	False	False	False	True	False	False	1
vector_machin	Low peak 2014; high peak 2015; low peak 2017	True	False	False	False	False	False	1
name_entiti	High peak 2016; low peak 2017	False	True	True	True	False	False	3
search_engin	Low peak 2015	False	False	False	False	False	True	1
similar_measur	Low peak 2015; high peak 2016	True	False	False	True	False	False	2
document_frequenc	Low peak 2014; high peak 2015; low peak 2017	False	False	False	False	False	False	0
featur_extract	Low peak 2014; high peak 2016; low peak 2017	True	False	True	True	False	False	3
posit_neg	High peak 2014; low peak 2015; high peak 2017	False	True	False	True	True	False	3
base_approach	High peak 2014; low peak 2016; high peak 2017	False	False	False	False	True	False	1
term_weight	High peak 2014; low peak 2015; high peak 2016	False	True	False	True	True	True	4
text_process	High peak 2014; low peak 2016; high peak 2017	False	False	False	False	False	False	0
text_analysi	Low peak 2015; high peak 2016; low peak 2017	False	False	False	True	False	False	1
learn_algorithm	Low peak 2014; high peak 2015; low peak 2017	False	False	False	False	False	False	0
word_remov	High peak 2014; low peak 2016	False	False	False	False	False	False	0
space_model	Low peak 2014; high peak 2016	False	False	False	True	False	False	1
artifici_intellig	Low peak 2015	False	False	False	False	False	False	0
cross_valid	Low peak 2015; high peak 2016; low peak 2017	True	True	False	True	False	False	3
arab_languag	High peak 2014; low peak 201; 5high peak 2016	True	True	True	True	True	False	5

5.5 SUMMARY AND CONCLUSION

The results obtained from the research were presented in this chapter. This meant presenting the document counts for the six-year period, as well as the total occurrence of themes. The quartile ranges of the document occurrences and the total occurrences were also calculated and presented.

The document occurrences for the top 20% of themes were presented in a bar graph. The trend for each theme's document occurrence is indicated and attached in the appendix. Calculations on the distance from the trend for each theme, for each year, were presented. The combined distance from the trend calculation was also presented, and the distance from the trend calculation for the top 20% of themes was presented in a bar graph. Each theme's distance from the trend changes over the six years is indicated and attached in the appendix. The changes to the distance from the trend calculations for the top 20% of themes were illustrated by means of both a bar graph and a table.

From the aforementioned, the average number of articles collected throughout the six years was found to be 57.5. There was a clear drop in the number of articles collected in 2018, which indicated a clear decreasing trend from 2014 onwards.

This chapter further indicated that the highest standard deviation in terms of document occurrence was 15.48. This revealed a large change in the document occurrence value for the theme over the six-year period.

The results presented in this chapter will be analysed in Chapter 6, which focuses on the areas of interest, as identified in the results of this research study.

Chapter Map 6: Discussion of analysis and findings

CHAPTER 6:
DISCUSSION OF ANALYSIS AND FINDINGS

6.1 INTRODUCTION

This chapter analyses the results and findings. It serves as a mediator between the results presented and the conclusion to the research questions set out in Chapter 1. The results presented are analysed in a manner that allows the main research question and the sub-questions set out in Chapter 1 to be answered. They also assist in fulfilling the objectives of the study.

6.2 AIM OF THE CHAPTER

This chapter aims to discuss the trends established from the data presented in Chapter 5.

6.3 SCOPE OF THE CHAPTER

This chapter contains an analysis of the results produced in Chapter 5. It contains examples of the data and how to produce the accurate root word of examples in the documents that were analysed. To achieve the aim of the research, the collected articles are considered by giving them a first glance. Then the changes in focus for the top 20% of themes based on their total occurrence are discussed. The themes are discussed in the sequence of those related to the main topic first, followed by those related to the literature, then the unexpected results and finally uncategorised or other themes. The last set of analyses will be the document occurrences for the top 20% of themes.

6.4 ANALYSIS AND FINDINGS

6.4.1 First glance

From the first glance of the results, the average of the documents gathered between 2013 and 2018 is 57.5. The most articles were gathered in 2014 and the least articles were gathered in 2018. There was a clear drop in the number of articles gathered in 2018. One could speculate that the decline in the number of articles gathered in 2018 could be directly related to Gartner's hype cycle. However, this cannot be confirmed and will require further research to validate the correlation between Gartner's hype cycle and the number of articles collected. However, the drop in the number of articles does mean that there is a reduction in research, which means that there is a reduction in interest in the topic of Stemming Algorithms.

6.4.2 Main topic-related results

There are two main sets of themes on the main topic. The first is "Stemming Algorithms" and the second is "Text Analysis" or "Text Mining" since both these themes show up in the main problem statement. In the results of the research, it was clear that themes related to "Stemming Algorithms" did not show up in the top 20% of themes. However, themes related to "Text Mining", "Text Analysis", "Text Processing" and "sentiment analysis" did show up. Even though four themes related to "Text Analysis" appeared in the top 20% of themes, they only make up 9.3% of the top 20% of themes. This means that 80% of the focus has not been on "Stemming Algorithms" itself, but rather on further extensions of Stemming Algorithms. Since none of the top 20% of themes address the development of Stemming Algorithms, all the themes that appear in the following sections will be in relation to the "application of Stemming Algorithms in Text Analysis."

To analyse the themes related to the main topic, Table 18 presents an analysis of the themes' focus. Table 18 follows the same format as that presented in the previous chapter. Therefore, the researcher will not elaborate on the format again.

Table 18: Topic-related analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
text_mine	High peak 2015 Low peak 2017	True	True	True	False	False	False	3
sentiment_analysi	High peak 2014 Low peak 2016 High peak 2017	False	True	True	True	True	True	5
text_process	High peak 2014 Low peak 2016 High peak 2017	False	False	False	False	False	False	0
text_analysi	Low peak 2015 High peak 2016 Low peak 2017	False	False	False	True	False	False	1

In Table 18, one can see that "sentiment analysis" is present as it is a type of Text Analysis. The theme "Text Mining" indicated that there was a high peak in 2015 and a low peak in 2017; therefore, there was an increase in focus on the theme until 2015. There was a decline in focus until 2017 and another incline following that. However, even with the given trend, the theme seems to have been below the trend line since 2016.

The theme "sentiment analysis" indicates that there was a high peak in 2014, a low peak in 2016 and another high peak in 2017; therefore, this means that there was an increase in interest in the theme until 2014; followed by a decline in focus until 2016 and an incline in focus until 2017; and finally a decline in focus following that. However, even with the given trend, "sentiment analysis" has been above the trend line since 2014, which means that it is a relevant topic of interest. It has also been in focus for five of the six years of the study.

The theme “Text Processing” indicates that there was a high peak in 2014, a low peak in 2016 and a high peak in 2017, which correlates directly with the trend of “sentiment analysis”; therefore, there is a possibility that the two themes are related to each other. However, due to the fact that this theme has been below the trend line for the entire period, it is possible that “sentiment analysis” plays a role in “Text Processing” and that “Text Processing” is a global theme.

Finally, the theme “Text Analysis” indicates that there was a low peak in 2015, a high peak in 2016 and a low peak in 2017; therefore, there was a decline in focus until 2015, followed by an incline until 2016, and then another decline until 2017 when it inclined again. Since the peak was in the same year that the theme was above the trend line, a possible breakthrough could have caught the attention of researchers in 2016.

6.4.3 Literature-related results

To analyse the themes related to the literature, themes have been grouped according to the steps in the Text Analysis process. The first step is data collection. Unfortunately, only one of the themes related to the data collection step were in the top 20% of themes. This indicates that data collection was not a major research concern among the top 20% of themes. The only theme related to data collection is presented in Table 19. The structure of Table 19 follows the format of Table 18; therefore, it will not be discussed again.

Table 19: Data collection-related analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
text_document	Low peak 2016 High peak 2017	False	True	False	False	True	False	2

Table 19 indicates that the theme “text document” had a low peak in 2016 and a high peak in 2017. This means that there was a decline in focus until 2016, followed by an incline until 2017. Finally, a decline was observed from then onwards. The theme “text document” does not seem

to be a major research focus over the six years since it was only in focus (below the trend line) twice during the six-year period.

The second step of the Text Analysis process is the pre-processing step. Table 20 displays the interpretation of the themes' focus over the six-year period. The structure of Table 20 follows the same format as Table 18; therefore, it will not be discussed again.

Table 20: Pre-processing-related analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
stop_word	Low peak 2015 High peak 2016	False	False	False	False	False	False	0
part_speech	High peak 2014 Low peak 2016 High peak 2017	True	True	False	False	True	False	3
word_remov	High peak 2014 Low peak 2016	False	False	False	False	False	False	0

Table 20 contains three themes. These themes only make up 6.97% of the top 20% of themes. With such a low percentage of the top 20% of themes referring to pre-processing, the main focus in the application of Stemming Algorithms was not related to pre-processing. Therefore, pre-processing can be considered to be an established field. Two of the three themes have also been out of focus (above the trend line) for all six years. However, from the analysis on the theme "part of speech", it might still have some relevance as it was in focus (below the trend line) for three of the six years.

To analyse the themes related to techniques of Text Analysis, Table 21 presents an analysis of the themes' focus. Table 21 follows the same format as Table 18; therefore, it will not be elaborated on again.

Table 21: Text Analysis techniques-related analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
natur_languag	High peak 2014 Low peak 2017	False	True	True	False	False	False	2
inform_retriev	Low peak 2015 High peak 2016 Low peak 2017	True	True	True	True	False	False	4
naiv_bay	Equilibrium	True	True	True	True	True	True	6
text_classif	Low peak 2014 High peak 2015 Low peak 2017	True	True	True	True	False	True	5
featur_select	Low peak 2014 High peak 2015	True	True	True	True	True	False	5
precis_recal	High peak 2016	False	False	False	True	True	False	2
document_cluster	Equilibrium	True	True	True	True	True	True	6
topic_model	Equilibrium	True	True	True	True	True	True	6
term_frequenc	Low peak 2014 High peak 2015 Low peak 2017	True	False	False	False	False	False	1
text_categor	Low peak 2016	True	True	True	False	False	False	3
support_vector	Low peak 2014 High peak 2016 Low peak 2017	True	False	False	False	False	False	1
vector_space	Low peak 2015 High peak 2016	False	False	False	True	False	False	1
name_entiti	High peak 2016 Low peak 2017	False	True	True	True	False	False	3
similar_measur	Low peak 2015 High peak 2016	True	False	False	True	False	False	2
document_frequenc	Low peak 2014 High peak 2015 Low peak 2017	False	False	False	False	False	False	0
featur_extract	Low peak 2014 High peak 2016 Low peak 2017	True	False	True	True	False	False	3
posit_neg	High peak 2014 Low peak 2015 High peak 2017	False	True	False	True	True	False	3
base_approach	High peak 2014 Low peak 2016 High peak 2017	False	False	False	False	True	False	1
term_weight	High peak 2014 Low peak 2015 High peak 2016	False	True	False	True	True	True	4
space_model	Low peak 2014 High peak 2016	False	False	False	True	False	False	1

Table 21 indicates that three themes, “Naïve Bayes”, “document cluster” and “topic model”, have constantly been in focus over the six-year period; however, all three these themes have been in equilibrium over the six years. This indicates that there is an alternation between an incline in focus and a decline in focus in research, which indicates that this theme is continuously researched, based on the presented data. Therefore, based on the results, a significant focus for the application of Stemming Algorithms in Text Analysis is on these three themes.

The theme “document frequency” has been out of focus for the entire six years, indicating that the theme is a common understanding or set of knowledge for the application of Stemming Algorithms. The theme falls under information retrieval and clustering.

As mentioned in Chapter 5, there is a clear drop in themes that are in focus in 2018. This indicates that there is a possible correlation between the focus of the themes and the plateau of productivity in Gartner’s hype cycle.

There are 20 themes related to the techniques of Text Analysis. These themes make up 46.5% of the top 20% of themes. Therefore, a major focus on the application of Stemming Algorithms in Text Analysis would be the different techniques of Text Analysis.

From all the themes presented in the literature-related results, the focus on Stemming Algorithms seems to have shifted away from data collection and pre-processing and the major focus during the six-year period was on techniques of Text Analysis. From the techniques of Text Analysis, three themes had major significance: “Naïve Bayes”, “document cluster” and “topic model”. This means that the application of Stemming Algorithms in Text Analysis was highly focused on these three topics of interest.

6.4.4 Unexpected results

In analysing the unexpected results, Table 22 presents the themes that the author had not expected.

Table 22: Unexpected results analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
machin_learn	High peak 2014 Low peak 2015 High peak 2016	True	True	True	True	True	False	5
train_data	Equilibrium	False	False	False	True	False	False	1
vector_machin	Low peak 2014 High peak 2015 Low peak 2017	True	False	False	False	False	False	1
learn_algorithm	Low peak 2014 High peak 2015 Low peak 2017	False	False	False	False	False	False	0
artifici_intellig	Low peak 2015	False	False	False	False	False	False	0

In Table 22, five themes related to artificial intelligence appeared in the top 20% of themes. Even though these themes only make up 11.6% of the top 20%, they still contribute a greater percentage than the “data collection” and “pre-processing” themes.

The theme “machine learning” has been in focus (below the trend line) for five of the six years. The only year that the theme was out of focus was 2018, which could potentially be linked to the plateau of productivity mentioned earlier. Nevertheless, being in focus for five of the six years indicates that it is a major theme for the application of Stemming Algorithms in Text Analysis.

Themes other than “machine learning” in Table 22 have only been in focus for one year (below the trend line). This either indicates that these themes are global concepts or that they are currently not established as breakthroughs. For example, the theme “training data” indicates that

it is in equilibrium, which means that there is an alternation between an incline and a decline in focus, which indicates that this theme is continuously researched, based on the presented data.

Additionally, the theme “artificial intelligence” had a low peak in focus in 2016 and inclined up to 2018. Future research may show how the theme “artificial intelligence” plays a role in the application of Stemming Algorithms.

6.4.5 Uncategorised/other results

Table 23 analyses the other themes that did not fall into any of the categories mentioned above. The structure of Table 23 follows the same format as Table 18; therefore, it will not be discussed again.

Table 23: Uncategorised analysis

Theme	Interpretation	2013	2014	2015	2016	2017	2018	Count
data_mine	High peak 2015 Low peak 2016 High peak 2017	False	True	True	False	True	False	3
languag_process	High peak 2014 Low peak 2017	False	False	False	False	False	False	0
social_media	High peak 2014 Low peak 2016 High peak 2017	True	True	True	False	True	True	5
social_network	Low peak 2014 High peak 2016	True	False	True	True	True	True	5
comput_linguist	High peak 2014	False	True	False	False	False	False	1
arab_text	Equilibrium	True	True	True	True	True	True	6
search_engin	Low peak 2015	False	False	False	False	False	True	1
cross_valid	Low peak 2015 High peak 2016 Low peak 2017	True	True	False	True	False	False	3
arab_languag	High peak 2014 Low peak 2015 High peak 2016	True	True	True	True	True	False	5

There are nine themes in Table 23. The themes “Arab text” and “Arab language” were both in focus (below the trend line) for five or more years. This indicates that the themes were a major

focus for the application of Stemming Algorithms in Text Analysis over the six-year period. Of interest, none of the other languages showed up in the top 20% of themes during the six-year period. However, even with so much emphasis on the language over the past six years, “Arab text” is still under equilibrium, which means that there is an alternation between an incline and a decline in research. Therefore, this theme was continuously researched, based on the presented data, and had no clearly identifiable breakthroughs.

The themes “social media” and “social networks” were both in focus (below the trend line) for five of the six years. Having been in focus for so many years indicates that it is a major application of Stemming Algorithms in the practice of Text Analysis.

The themes “Data Mining”, “language processing”, “computer linguistics”, “search engine” and “cross-validation” have been in focus for three or fewer years over the six-year period. This indicates that even though they were in the top 20% of themes, they did not play a major role in the focus on the application of Stemming Algorithms in Text Analysis. However, there was a low peak for “computer linguistics” in 2014, and research on this theme is on the rise. Future research could investigate this theme further to see if it increases in focus.

6.4.6 Document occurrence-related results

The document occurrence results indicate the number of research articles that include the theme. This shows how much research has been done with or including the theme. It is critical to keep in mind that 2018 had fewer articles related to the application of Stemming Algorithms in Text Analysis. Respectively, none of the top 20% of themes had an increasing trend in document occurrence. The decline in research papers in 2018 can potentially indicate a reduction in research on the topic.

To analyse the document occurrence trends of the top 20% of themes, some tables have been extracted from Chapter 4 and presented again. The first table that is presented again, Table 24, presents document occurrences on themes in the top 20% that display a constantly decreasing trend.

Table 24: Document occurrence decreasing trends

Theme	Interpretation
machin_learn	Decreasing
inform_retriev	Decreasing
arab_text	Decreasing
text_categor	Decreasing
vector_space	Decreasing
space_model	Decreasing

As indicated earlier in this chapter, "machine learning" has been in focus for five years, "information retrieval" has been in focus for four years and "Arab text" has been in focus for six years. Even though these three themes have been in focus for so many years, the number of research papers produced for them has been declining, which means that there is a potential reduced interest in research on the topic.

Table 25 presents the document occurrence themes that are in equilibrium in the top 20% of themes.

Table 25: Document occurrence equilibrium trends

Theme	Interpretation
neural_network	Equilibrium

In Table 25, there is only one theme: "neural networks". This means that the theme "neural networks" is alternating between a breakthrough and a reduction in research, which indicates that this theme is continuously researched, based on the presented data.

Table 26 presents the document occurrence themes in the top 20% of themes that reached a high peak in the year.

Table 26: Document occurrence high peak trends

Theme	Interpretation
natur_languag	High peak 2015
languag_process	High peak 2015
part_speech	High peak 2015
search_engin	High peak 2015
base_approach	High peak 2015
text_analysi	High peak 2015
arab_languag	High peak 2015
stop_word	High peak 2014
text_document	High peak 2014
social_network	High peak 2014
term_frequenc	High peak 2014
similar_measur	High peak 2014
document_frequenc	High peak 2014
posit_neg	High peak 2014
term_weight	High peak 2014
artifici_intellig	High peak 2014

With the high peak analysis of the trends, it is clear that there was an increase in interest on the theme until the peak year (2014/15). From the peak year, there was a reduction in research, and interest in the theme declined. The reduction could also be interpreted as a reduced interest in research based on the assumption that current avenues of research could have been exhausted and new avenues might need to be explored. Even though the theme “social networks” has been in focus for five years, it also revealed a decline in research output in the application of Stemming Algorithms since 2014.

6.5 SUMMARY AND CONCLUSION

In this chapter, the research findings were discussed, and a few main points were discussed. The document counts gathered from the data collection process were presented, and the results were separated into themes to answer the sub-questions in this study. The sections on analysis and findings included the first glance, the main topic-related results, the literature-related results, the unexpected results and uncategorised/other results. The last set of findings analysed the document occurrence results.

All the discussions revolving around the themes in this chapter were in relation to the application of Stemming Algorithms in Text Analysis, unless otherwise specified. Additionally, the themes that were discussed were the themes that appeared in the top 20% of the total occurrence of themes.

From the abovementioned, there appeared to be a major drop in article count in 2018. There is a possible correlation between Gartner's hype cycle and the decline in article count in 2018. Future research will be required to validate this point. This drop in 2018 showed how the document occurrences for each theme decreased in 2018.

With regard to the main topic-related themes, 80% of the focus has not been on "Stemming Algorithms" itself, including further extensions of Stemming Algorithms. However, with regard to themes around "Text Analysis", "sentiment analysis" has been above the trend line since 2014, which makes it a relevant topic of interest. It has also been in focus for five of the six years.

With regard to the literature-related results, revolving already the steps within the Text Analysis process, there was only one theme in the data collection stage of Text Analysis. Three of the themes were in the pre-processing stage. Twenty of the themes were in the techniques stage.

None of the themes were in the knowledge discovery stage. Therefore, the major focus could be in the techniques stage of Text Analysis.

In the data collection stage, the theme “text document” does not seem to be a major focus over the six years since it was only in focus (below the trend line) twice over the six-year period.

In the pre-processing stage, two of the three themes (“stop word”, “part of speech” and “word removal”) have been out of focus (above the trend line) for all six years. However, the theme “part of speech” might still have some relevance as it was in focus (below the trend line) for three of the six years.

In the techniques stage, “Naïve Bayes”, “document cluster” and “topic model” were constantly in focus over the six-year period; however, these themes were continuously researched, based on the presented data. Therefore, based on the results, there was a significant focus on these three themes. The theme “document frequency” is a common understanding or set of knowledge.

The unexpected results make up 11.6% of the top 20% of themes since these results revolve around artificial intelligence. The results reveal that “artificial intelligence”, as a grouping of themes, has been more in focus than “data collection step” or “pre-processing step” as a grouping of themes over the six-year period.

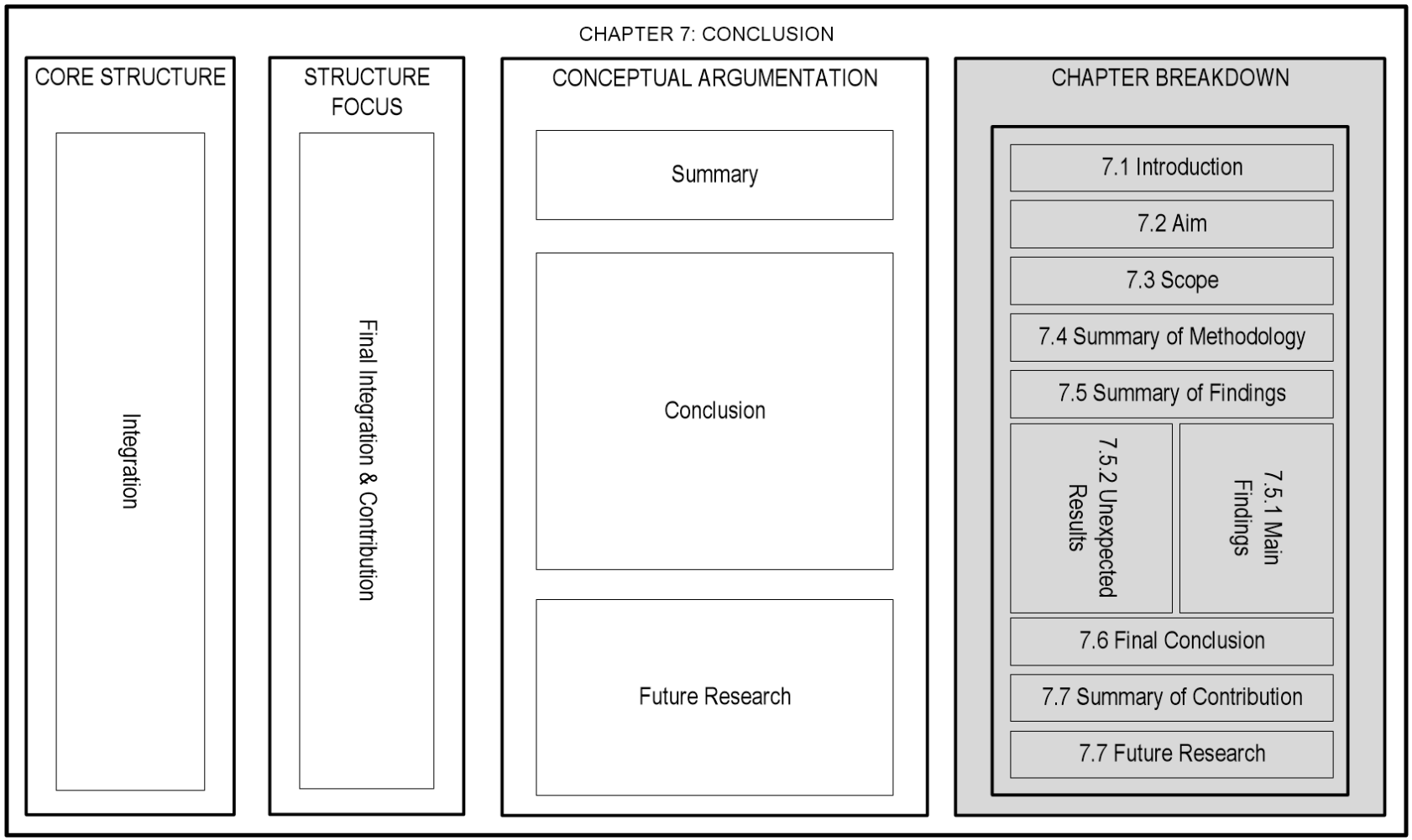
In the unexpected results, “machine learning” is a major focus. The themes “training data”, “vector machine”, “learning algorithm” and “artificial intelligence” are global concepts, or they have currently not established a breakthrough. “Training data” is continuously researched, based on the presented data. “Artificial intelligence” had a low peak in focus in 2016 and inclined until 2018.

Alternatively, with regard to the literature related to Stemming Algorithms that is directly involved in the Text Mining process, a few themes have been constantly out of focus throughout the six years. These are “stop word”, “word removal”, “remove stop” and “document frequency”. Three of the four themes fall under the pre-processing stage of Text Mining. This shows that the association of Stemming Algorithms with the rest of pre-processing has been established as a global concept.

With regard to the unexpected results, “Arab text” and “Arab language” have been a major focus. None of the other languages showed up in the top 20% of themes during the six-year period. “Arab text” is continuously researched, based on the presented data, and has no clearly identifiable breakthroughs. The themes “social media” and “social networks” were found to be a major application. The themes “Data Mining”, “language processing”, “computer linguistics”, “search engine” and “cross-validation” were in the top 20% of themes; however, they do not play a major role in the focus of the application of Stemming Algorithms in Text Analysis.

With regard to the document occurrence analysis, the number of research papers produced for “machine learning”, “information retrieval” and “Arab text” has been declining, indicating a potential reduction in research on these topics. “Neural networks” is a theme that is continuously researched, based on the presented data. “Nature language”, “language processing”, “part of speech”, “search engine”, “based approach”, “Text Analysis”, “Arab language”, ”stop word”, “text document”, “social network”, “term frequency”, “similar measure”, “document frequency”, “positive negative”, “term weight” and “artificial intelligence” revealed a constant incline in research until 2014/15, followed by a decline since then. The decline in these topics suggests a reduction in research, where all the knowledge pertaining to the topic has already been exhausted.

In the next chapter, the research is concluded by answering the questions set out in Chapter 1.



Chapter Map 7*:* Conclusion

CHAPTER 7:
CONCLUSION

7.1 INTRODUCTION

Text Mining is done to understand the main concepts within a body of text. Understanding the main concepts within a text can aid decision-making within a company. The Stemming Algorithm is part of the Text Mining process.

The purpose of this study was to aggregate the existing research that caters for the Stemming Algorithm as part of Text Analysis between 2013 and 2018. Stemming Algorithms are used in Text Analysis. They improve the process of Text Analysis by indexing various forms of a word as one word instead of different words. Indexing the words from the collected sources in such a way speeds up the Text Mining process.

The purpose of this study was to determine Stemming Algorithm trends. These trends provide a better understanding of the interest in research on a particular theme, as well as the decline in interest on a particular theme over the years. Further research can be conducted on the reasons for the reduced interest in the different focus areas of Stemming Algorithms, as well as on potential problems related to Stemming Algorithms. By tying the results from the previous chapters together, this chapter will present the conclusion to this research.

7.2 AIM OF THE CHAPTER

This chapter aims to conclude the study, summarise the methodology used, analyse the findings, and present possible topics for future research.

7.3 SCOPE OF THE CHAPTER

To achieve the aims of the research, this chapter will summarise the methodology and findings. This will include a discussion of both the main results and the unexpected results, its contribution to the field of study and future research. The final conclusion will answer the main research question.

7.4 SUMMARY OF THE METHODOLOGY USED

This subsection gives a general overview of the methodology applied to the research conducted for this study.

Methodologically, a systematic review was followed according to the stages of the research process described by Okoli and Schabram (2010), accompanied by the knowledge and understanding obtained from Attride-Stirling (2001). According to the methodology proposed by Okoli and Schabram (2010), there are four main stages in the research process: planning, selection, extraction and reporting.

To follow the steps suggested by Okoli and Schabram (2010), Google Scholar was selected as a platform to collect data. Google Scholar provided the capability to set and restrict variables that are mentioned throughout the systematic literature review process. By using Google Scholar, the researcher was able to restrict the possible access databases, specify only empirical research (journal articles), specify only pdf files, specify only articles written in English, and use terms related to the theme of a topic, such as “Stemming Algorithms”, “stemmer”, “Text Mining” and “Text Analysis”. However, Google Scholar was only able to return a list of results and not download the articles.

WebHarvy was used to identify each Google Scholar result and download the respective articles. Once an article had been downloaded and placed in its respective year, its theme was identified by means of a Text Analysis process.

The Text Analysis process that was used was adapted from Aryal et al. (2015), whose research identified themes from healthcare information systems research. This research was similarly adapted to themes from Text Analysis articles, and included data collection, pre-processing and the production of a word list. The data collection was done by means of WebHarvy and not according to the method of Aryal et al. (2015) as the outcome of their research differed from that performed in this research study. The steps of aggregating concepts and reducing them to common research terms were done according to the study of Aryal et al. (2015) as their research focused on clusters of words. However, this research used linear regression to analyse the word list.

To analyse the results, the calculations and interpretations of Kerlinger and Lee (2000) were carried out, where the correlation between two factors was compared. This interpretation assisted in answering the questions set out in the problem statement.

7.5 SUMMARY AND FINDINGS

This section discusses the main findings of the research, as well as the unexpected results. The main findings answer the sub-questions set out in Chapter 1. The unexpected results discuss some results that were not expected as part of the literature study.

7.5.1 Main findings

The main findings set out to answer the sub-questions of the problem statement stated in Chapter 1, which lead to answering the main research question.

The first research sub-question is: *"What are the characteristics of Stemming Algorithms?"*

Chapter 3 discusses that the characteristics of Stemming Algorithms are precision, uniqueness, finiteness, input, output, generality, determining the stem of a word and removing the suffix.

The second research sub-question is: *"What are the problems associated with Stemming Algorithms in text and Text Analysis?"*

Chapter 3 discusses how none of the reviewed Stemming Algorithms presented have the combined advantage that every algorithm can provide. Seventy per cent of the Stemming Algorithms evaluated are unable to be adapted to other languages. The algorithms that can cater for different Stemming Algorithms have accuracy issues. All the algorithms have at least one of the following as a disadvantage: language dependency, resource consumption, accuracy, performance duration and incapable of prefix stripping.

The problems that have been identified are related to three of the characteristics of Stemming Algorithms: fitness, generality and determining the stem of a word. Fitness refers to the resource requirement to accomplish the performance of the algorithm. Generality refers to the ability of the algorithm to adapt to other languages. Determining the stem of a word refers to the accuracy of the stem.

The third research sub-question is: *"What is the application of common Stemming Algorithms in Text Analysis?"*

Chapter 2 indicated that "Stemming Algorithms" fall within the pre-processing step of Text Analysis. Therefore, one can conclude that the direct application of Stemming Algorithms is to fulfil the pre-processing step of Text Analysis. The purpose of the pre-processing step in the Text

Mining process is to provide more accurate results. Since Stemming Algorithms work in favour of pre-processing text, one can conclude that the application of Stemming Algorithms also assists in providing more accurate results in the Text Mining process. Chapter 2 also indicated the applications or fields of application of Text Analysis. Stemming Algorithms indirectly fulfil the purpose of Text Analysis by forming part of the pre-processing step of Text Analysis.

The application of a Stemming Algorithm assists in improving the functionality or development of other parts of the Text Mining process and the application of Text Analysis in a wide range of fields.

The fourth research sub-question is: *"How have Stemming Algorithms been applied and changed in Text Mining over the past six years (2013 to 2018)?"*

To answer the fourth sub-question, the application of Stemming Algorithms can be seen to be based on the resulting themes of the research. Results of the research is presented in Chapter 5 and discussed in Chapter 6. Based on the Pareto Principle, the top 20% of the themes represent 80% of the content. Therefore, 80% of the application of Stemming Algorithms in Text Analysis can be seen in the top 20% of the themes identified during the six-year period. The resulting themes show that one theme pertains to data collection, four themes pertain to the pre-processing step of Text Mining, and 20 themes pertain to the processing step with applied techniques. Five themes pertain to artificial intelligence, and eight themes pertain to fields of application.

The top 20% of themes are "text mine", "sentiment analysis", "text process", "Text Analysis", "text document", "stop word", "part of speech", "word removal", "natural language", "information retrieval", "Naive Bayes", "text classifier", "feature selection", "precision and recall", "document cluster", "topic model", "term frequency", "text category", "support vector", "vector space", "named entity", "similar measure", "document frequency", "feature extraction", "positive negative", "based

approach", "term weight", "space model", "machine learning", "train data", "vector machine", "learning algorithm", "artificial intelligence", "data mine", "language process", "social media", "social network", "computer linguists", "Arab text", "search engine", "cross valid" and "Arab language".

From the top 20% of themes, it is clear that themes related directly to "stem*"* in any way are absent. Therefore, the top 80% focus on Stemming Algorithms is no longer on the extension or enhancement of the Stemming Algorithm itself. The focus has shifted to its application.

Due to the fact that the top 20% of themes all relate to the application of Stemming Algorithm Text Analysis, the discussions revolving around the themes under this sub-question related to the application of Stemming Algorithms in Text Analysis.

With regard to the themes related to the main topic, 80% of the focus has not been on "Stemming Algorithms" itself, including further extensions of Stemming Algorithms. However, results show that "sentiment analysis" remains a relevant topic of interest. Alternatively, 80% of the focus is on the application of Stemming Algorithms in Text Analysis.

Results show that the application of Stemming Algorithms in Text Analysis has moved away from "text document", "stop word", "part of speech" and "word removal". However, the theme "part of speech" might still have some relevance as it was in focus (below the trend line) for three of the six years. Results also show that "Naïve Bayes", "document cluster" and "topic model" were constantly in focus over the six-year period; however, these themes are continuously researched, based on the presented data. Therefore, based on the results, a significant focus is on these three themes. The theme "document frequency" is a common understanding or set of knowledge.

Further results show that “Arab text” and “Arab language” have been a major focus. “Arab text” is continuously research based on the presented data and has no clearly identifiable breakthroughs. “Social media” and “social networks” are a major application. The themes “Data Mining”, “language processing”, “computer linguistics”, “search engine” and “cross-validation” were in the top 20% of themes; however, they do not play a major role in the focus of the application of Stemming Algorithms in Text Analysis.

With regard to the document occurrence analysis, the number of research papers produced for “machine learning”, “information retrieval” and “Arab text” has been declining, indicating a potential reduced interest in research. “Neural networks” is a theme that is continuously researched, based on the presented data. A constant incline in research was found for the themes “nature language”, “language processing”, “part of speech”, “search engine”, “based approach”, “Text Analysis”, “Arab language”, “stop word”, “text document”, “social network”, “term frequency”, “similar measure”, “document frequency”, “positive negative”, “term weight” and “artificial intelligence” until 2014/15, followed by a decline since then. The decline in these topics suggests a reduction in research interest.

Results derived from the gathered data indicate that the trend of the application of Stemming Algorithms in Text Analysis started to decrease in correlation to Gartner’s hype cycle. The reason for this would require additional research.

7.5.2 Unexpected results

Interesting results include the increase in the application of Stemming Algorithms within the domain of decision making. Although the overall occurrence of this theme is not high in the sample set, it can still represent how Stemming Algorithm is reaching a plateau of productivity to aid

business. There is also a fluctuating, yet high interest in social media as an application of Stemming Algorithms.

A major unexpected set of themes revolved around “artificial Intelligence” when one considers the increase in interest in this theme. This answers the question: *“What are the advances in Stemming Algorithms over the past six years?”*

Results show that “machine learning” is also a major focus. The themes, “training data”, “vector machine”, “learning algorithm” and “artificial intelligence” are global concepts, or they have currently not established a breakthrough. “Training data” is continuously researched, based on the presented data. “Artificial intelligence” had a low peak in focus in 2016, but inclined up to 2018.

From the results, four of the themes identified related to “artificial intelligence”. This shows that 11.6% of the themes in the top 20% of themes relate to artificial intelligence. This indicates a significant valuation in relation to the application of Stemming Algorithms. Therefore, it can be concluded that the application of Stemming Algorithms or the extension of Stemming Algorithms is moving towards artificial intelligence. In terms of artificial intelligence, it is concluded that “machine learning” has been in focus for five of the six years between 2013 and 2018.

7.6 FINAL CONCLUSIONS

The final conclusions of this research answer the question: *“What are the advances in the application of Stemming Algorithms in Text Analysis over the past six years (2013 to 2018)?”*

With the given answers to the sub-questions in the summary and findings, the answer to the research question is that the main concepts of the theme itself have been out of focus for the past six years, whereas the applications of Stemming Algorithms have been developing and increasing

over the same period. This confirms that Stemming Algorithms have reached a plateau of productivity and/or barrier to further development. To support that point, the themes “social media” and “social networks” have been a significant focus over the six-year period. This shows that there is a significant application of Stemming Algorithms in “social media” over the six-year period. Even though this has been identified, the Stemming Algorithm itself still has problems that have not been solved, such as the fact that 70% of the Stemming Algorithms evaluated are unable to be adapted to other languages. The algorithms that can cater for different languages have accuracy issues. This indicates that there is potential for future research.

Other than the far application of Stemming Algorithms, an improvement in the other components in the process of Text Mining and Text Analysis has been observed. From the results, it is evident that the development of topic models in relation to the application of Stemming Algorithms has been significant over the six-year period.

7.7 SUMMARY OF CONTRIBUTIONS

From a theoretical perspective, the consolidated characteristics of Text Analysis and Stemming Algorithms have been defined for future research.

From a methodological perspective, a combined method of Text Analysis and systematic review has been designed to analyse trends of different themes over a given period of time. This methodology can be used again for future research.

From a practical perspective, researchers now have a direction for further research since this research concluded the unaccomplished capabilities of the existing Stemming Algorithms.

7.8 FUTURE RESEARCH

From a theoretical perspective, researchers can investigate and validate the correlation between the number of articles collected in relation to Gartner's hype cycle and develop methods to predict Gartner's hype cycle based on the number of collectable journal articles.

From a methodological perspective, a design and creation algorithm can be developed to determine the themes that are irrelevant to the given context so that they can be automatically removed and not require further human intervention or investigation. Additionally, future research could develop an algorithm capable of selecting the most plausible origination of a stemmed theme from the collected data so that less human intervention would be required.

From a practical perspective, researchers can investigate each theme qualitatively and evaluate each theme within the data set to determine the reasons behind such trends. As 2,207 themes have been identified, this will take quite a lot of work. Furthermore, one can validate this research's methodology on the conclusion of "a decrease in research on Stemming Algorithms over the six-year period". Even though the decrease in research for many algorithmic fields is a usual trend, it cannot be assumed without proven research.

Future research includes the design and development of information architecture to support the systematic structuring of the Stemming Algorithm knowledge gap. Future research also includes the design of a Stemming Algorithm that would automatically and responsively adapt to historical changes in languages. Finally, without looking into historical changes in languages, future research can also include a dynamic Stemming Algorithm that is language independent, and which also has high levels of accuracy.

BIBLIOGRAPHY

Aguiar, C. Z., Cury, D., & Zouaq, A. (2016). Automatic construction of concept maps from texts. Paper presented at the Concept Mapping Conference (CMC), Tallinn, Estonia. Retrieved from http://cmc.ihmc.us/cmc2016papers/cmc2016-p90.pdf [GS SEARCH].

Akasereh, M. (2015). *Multilingual and domain-specific IR.* Doctoral thesis, Université de Neuchâtel, Neuchâtel, Switzerland. Retrieved from http://doc.rero.ch/record/255758/files/00002460.pdf.

Akhondi, S. A., Klenner, A. G., Tyrchan, C., Manchala, A. K., Boppana, K., Lowe, D., . . . Kors, J. A. (2014). Annotated chemical patent corpus: A gold standard for Text Mining. *PloS One, 9*(9), e107477.

Altman, E. I., & Saunders, A. (1997). Credit risk measurement: Developments over the last 20 years. *Journal of Banking and Finance, 21*(11–12), 1721–1742.

Ameka, F. K. (2016). The uselessness of the useful: Language standardisation and variation in multilingual contexts. In Tieken-Boon van Ostade, I. & Percy, C. (eds) *Prescription and tradition in language: Establishing standards across time and space, 165*, 59–71. Bristol: Multilingual Matters.

Aryal, A., Gallivan, M., & Tao, Y. Y. (2015). Using latent semantic analysis to identify themes in IS healthcare research. Paper presented at the 21th Americas Conference on Information Systems (AMCIS), Fajardo, Puerto Rico.

Attride-Stirling, J. (2001). Thematic networks: An analytic tool for qualitative research. *Qualitative Research, 1*(3), 385–405.

Bakhtin, M. (1977). The problem of the text (An essay in philosophical analysis). *Soviet Studies in Literature, 14*(1), 3–33.

Basiri, M. E., Ghasem-Aghaee, N., & Reza, A. (2017). Lexicon-based sentiment analysis in Persian. *Current and Future Developments in Artificial Intelligence, 30*, 154–183.

Beale, S. (2018). *Gagging for it: Irony, innuendo and the politics of subversion in women's comic performance on the post-1880 London music-hall stage and its resonance in contemporary practice.* Middlesex University, London, UK.

Berry, M. W. (2004). *Survey of Text Mining*. New York, NY: Springer.

Bessmertny, I., Platonov, A., Poleschuk, E., & Pengyu, M. (2016). Syntactic Text Analysis without a dictionary. Paper presented at the Application of Information and Communication Technologies (AICT), Baku, Azerbaijan.

Bhatia, R. (2013). *Matrix analysis, 169*. Berlin: Springer Science and Business Media.

Brown, G., & Yule, G. (1983). *Discourse analysis.* Cambridge: Cambridge University Press.

Brown, R. W. (1957). Linguistic determinism and the part of speech. *The Journal of Abnormal and Social Psychology, 55*(1), 1.

Cai, T., Giannopoulos, A. A., Yu, S., Kelil, T., Ripley, B., Kumamaru, K. K., . . . Mitsouras, D. (2016). Natural language processing technologies in radiology research and clinical applications. *Radiographics, 36*(1), 176–191.

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine, 9*(2), 48–57.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 161–175.

Arsanjani, A., Hailpern, B., Martin, J., & Tarr, P. L. (2003). Web services: Promises and compromises. *ACM Queue, 1*(1), 48-58.

Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice, 283*. Boston, MA: Addison-Wesley Reading.

Darlington, R. B., & Hayes, A. F. (2016). *Regression analysis and linear models: Concepts, applications, and implementation*. New York, NY: Guilford Publications.

Dawson, J. (1974). Suffix removal and word conflation. *ALLC Bulletin, 2*(3), 33–46.

Heudecker, N. (2013). Gartner's hype cycle for big data, 2013. Retrieved from http://https//www.gartner.com/doc/2574616/hype-cycle-big-data.

Dohare, S., Karnick, H., & Gupta, V. (2017). *Text summarization using abstract meaning representation.* Cornell University, Ithaca, NY.

Dredze, M., Paul, M. J., Bergsma, S., & Tran, H. (2013). Carmen: A twitter geolocation system with applications to public health. Paper presented at the AAAI Workshop on Expanding the Boundaries of Health Informatics using AI (HIAI), Palo Alto, CA, USA.

Drozdova, N., Utochkin, D., & Kleppe, I. A. (2017). Web 2.0: Online communities or bla-bla land? *Advances in Consumer Research, 45*, 415–418.

Eldridge, R. T. (2017). Review of "Irony and idealism: Rereading Schlegel, Hegel, and Kierkegaard" by F. Rush. *European Journal of Philosophy, 25*(4), 1228–1231.

Elragal, A., & Haddara, M. (2014). Big data analytics: A Text Mining-based literature analysis. *Proceedings of the Norsk Konferanse for Organisasjoners bruk av IT (Nokobit)*, *22*(1).

Ertek, G., Tapucu, D., & Arin, I. (2013). Text Mining with rapidminer. *RapidMiner: Data Mining use cases and business analytics applications*, *241*.

Feldman, R., & Sanger, J. (2007). *The Text Mining handbook: Advanced approaches in analyzing unstructured data.* Cambridge: Cambridge University Press.

Fisicaro, C., & Gauvin, L. (2018). *Part of speech tagging hidden Markov models vs recurrent neural networks.* Master's dissertation, University of Turin, Turin, Italy.

Gaikwad, S. V., Chaugule, A., & Patil, P. (2014). Text Mining methods and techniques. *International Journal of Computer Applications, 85*(17).

Goeminne, M., & Mens, T. (2011). Evidence for the Pareto Principle in open source software activity. Paper presented at the the Joint Porceedings of the 1st International Workshop on Model Driven Software Maintenance and 5th International Workshop on Software Quality and Maintainability, London, UK.

Goldwater, S. (2016). ANLP Lecture 6 N-gram models and smoothing. School of Informatics, University of Edinburgh, Eidenburgh, Scotland.

Graepel, T., Herbrich, R., Bollmann-Sdorra, P., & Obermayer, K. (1999). Classification on pairwise proximity data. Paper presented at the Advances in Neural Information Processing Systems (NIPS) Conference, Denver, CO, USA.

Gupta, G., & Malhotra, S. (2015). Text documents tokenization for word frequency count using rapid miner (taking resume as an example). *International Journal of Computer Applications, 975*, 8887.

Gurusamy, V., & Nandhini, K. (2017). Stemming techniques for Tamil language. *International Journal of Computer Science and Engineering Technology, 8*(6), 225–231.

Haddaway, N. R. (2015). The use of web-scraping software in searching for grey literature. *Grey Journal, 11*(3), 186–190.

Hariharan, R., Hore, B., Li, C., & Mehrotra, S. (2007). Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems. Paper presented at the 19th International Conference on Scientific and Statistical Database Management (SSDBM), Calgary, Canada.

Harris, R. L. (2000). *Information graphics: A comprehensive illustrated reference.* Oxford: Oxford University Press.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*(3), 283–304.

Huddleston, R., & Pullum, G. (2005). The Cambridge grammar of the English language. *Zeitschrift für Anglistik und Amerikanistik, 53*(2), 193–194.

Ismailov, A., Jalil, M. A., Abdullah, Z., & Rahim, N. A. (2016). A comparative study of Stemming Algorithms for use with the Uzbek language. Paper presented at the 3rd International Conference on Computer and Information Sciences (ICCOINS), Seri Iskandar, Malaysia.

Janetzko, D. (2016). Nonreactive data collection online. *The SAGE handbook of online research methods*, *76*.

Jivani, A. G. (2011). A comparative study of Stemming Algorithms. *International Journal of Computer Technology Applications, 2*(6), 1930–1938.

Johnson, H. L., Bretonnel Cohen, K., & Hunter, L. (2007). A fault model for ontology mapping, alignment, and linking systems. *Biocomputing 2007,* 233–244. Singapore: World Scientific.

Kandogan, E., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., & Zhu, H. (2006). Avatar semantic search: A database approach to information retrieval. Paper

presented at the 2006 ACM SIGMOD International Conference on Management of Data, Chicago, IL, USA.

Karthika, S., & Sairam, N. (2015). A naive Bayesian classifier for educational qualification. *Indian Journal of Science and Technology, 8*(16).

Kerlinger, F. N., & Lee, H. B. (2000). *Foundations of behavioral research,* 4th ed. Fort Worth, TX: Harcourts College Publishers.

Kharlamov, A. A., Yermolenko, T. V., & Zhonin, A. A. (2013). Text understanding as interpretation of predicative structure strings of main text's sentences as result of pragmatic analysis (combination of linguistic and statistic approaches). Paper presented at the 15th International Conference on Speech and Computer, Pilsen, Czech Republic.

Krensky, P., & Hare, J. (2018). Hype cycle for data science and machine learning. Retrieved from https://www.gartner.com/doc/3883664?ref=unauthreader&srcId=1-4730952011.

Kucher, K., & Kerren, A. (2015). Text visualization techniques: Taxonomy, visual survey, and community insights. Paper presented at the IEEE Pacific Visualization Symposium (PacificVis) 2015, Hangzhou, China.

Labrinidis, A., & Jagadish, H. V. (2012). Challenges and opportunities with big data. *Proceedings of the VLDB Endowment, 5*(12), 2032–2033.

Lee, H., Surdeanu, M., & Jurafsky, D. (2017). A scaffolding approach to coreference resolution integrating statistical and rule-based models. *Natural Language Engineering, 23*(5), 733–762.

Lin, N. (2017). Building a network theory of social capital. *Social capital,* 3–28. Abingdon: Routledge.

Linden, A., & Fenn, J. (2003). Understanding Gartner's hype cycles. Strategic Analysis Report No. R-20-1971, Gartner Inc.

Lovins, J. B. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics, 11*(1–2), 22–31.

Majumder, P., Mitra, M., Parui, S. K., Kole, G., Mitra, P., & Datta, K. (2007). YASS: Yet another suffix stripper. *ACM Transactions on Information Systems, 25*(4), 18.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition, and productivity.* New York, NY: McKinsey Global Institute.

Marres, N., & Weltevrede, E. (2013). Scraping the social? Issues in live social research. *Journal of Cultural Economy, 6*(3), 313–335.

Mathiak, B., & Eckstein, S. (2004). Five steps to Text Mining in biomedical literature. Paper presented at the 2nd European Workshop on Data Mining and Text Mining in Bioinformatics, Pisa, Italy.

Mayfield, J., & McNamee, P. (2003). Single n-gram stemming. Paper presented at the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, New York, NY, USA.

McCusker, K., & Gunaydin, S. (2015). Research using qualitative, quantitative or mixed methods and choice based on the research. *Perfusion, 30*(7), 537–542.

Minhas, S. Z. (2016). *A corpus driven computational intelligence framework for deception detection in financial text.* Doctoral dissertation, University of Stirling, Scotland. Retrieved from https://dspace.stir.ac.uk/bitstream/1893/25345/1/FINAL-%20MAIN.pdf.

Moral, C., De Antonio, A., Imbert, R., & Ramírez, J. (2014). A survey of Stemming Algorithms in information retrieval. *Information Research: An International Electronic Journal, 19*(1), 1–14.

Müller, H-M., Kenny, E. E., & Sternberg, P. W. (2004). Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biology, 2*(11), e309.

Müller, O., Junglas, I., Debortoli, S., & Vom Brocke, J. (2016). Using Text Analytics to derive customer service management benefits from unstructured data. *MIS Quarterly Executive, 15*(4), 243–258.

Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2014). *Automated data collection with R: A practical guide to web scraping and Text Mining.* Hoboken, NJ: Wiley.

Naumov, I., & Vykhovanets, V. (2016). Using syntactic Text Analysis to estimate educational tasks' difficulty and complexity. *Automation and Remote Control, 77*(1), 159–178.

Neuendorf, K. A. (2016). *The content analysis guidebook.* London: Sage.

Nielsen, F. Å. (2011). *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs.* Cornell University, Ithaca, NY, USA.

Norouzizadeh Dezfouli, F., Dehghantanha, A., Eterovic-Soric, B., & Choo, K-K. R. (2016). Investigating social networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms. *Australian Journal of Forensic Sciences, 48*(4), 469–488.

Nowak, B. A., Nowicki, R. K., Woźniak, M., & Napoli, C. (2015). Multi-class nearest neighbour classifier for incomplete data handling. Paper presented at the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland.

Oates, B. J. (2005). *Researching information systems and computing*. London: Sage.

Okoli, C., & Schabram, K. (2010). A guide to conducting a systematic literature review of information systems research. *Sprouts: Working Papers on Information Systems, 10*(26), 1–51.

Papadopoulos, T., Gunasekaran, A., Dubey, R., Altay, N., Childe, S. J., & Fosso-Wamba, S. (2017). The role of big data in explaining disaster resilience in supply chains for sustainability. *Journal of Cleaner Production, 142*, 1108–1118.

Parekh, M. B., Saksena, A., Ringshia, A., & Chaudhari, S. (2017). Simulation of a two head disk scheduling algorithm: An algorithm determining the algorithm to be imposed on the heads based on the nature of track requests. Paper presented at the 2nd International Conference on Electrical, Computer and Communication Technologies (ICECCT), Chennai, India.

Pimpalshende, A., & Mahajan, A. (2017). Test model for stop word removal of devnagari text documents based on finite automata. Paper presented at the 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India.

Porter, A. L., & Cunningham, S. W. (2005). *Text Mining exploring new technologies for competitive advantage*. Hoboken, NJ: Wiley.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Priandini, N., Zaman, B., & Purwanti, E. (2017). Categorizing document by fuzzy C-means and K-nearest neighbors approach. Paper presented at the AIP Conference, Surabaya, Indonesia.

Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big Data, 1*(1), 51–59.

Rajadesingan, A., Zafarani, R., & Liu, H. (2015). Sarcasm detection on Twitter: A behavioral modeling approach. Paper presented at the 8th ACM International Conference on Web Search and Data Mining, Shanghai, China.

Rani, S. R., Ramesh, B., Anusha, M., & Sathiaseelan, J. (2015). Evaluation of stemming techniques for text classification. *International Journal of Computer Science and Mobile Computing, 4*(3), 165–171.

Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (1999). EDGAR: Extraction of drugs, genes and relations from the biomedical literature. *Biocomputing 2000,* 517–528. Singapore: World Scientific.

Rittenburg, T. L., Gladney, G. A., & Stephenson, T. (2016). The effects of euphemism usage in business contexts. *Journal of Business Ethics, 137*(2), 315–320.

Ristoski, P., Bizer, C., & Paulheim, H. (2015). Mining the web of linked data with rapidminer. *Web Semantics: Science, Services and Agents on the World Wide Web, 35*, 142–151.

Sanders, R. (1987). The Pareto Principle: Its use and abuse. *Journal of Services Marketing, 1*(2), 37–40.

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. Paper presented at the International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA.

Saharia, N., Konwar, K. M., Sharma, U., & Kalita, J. K. (2013). An improved stemming approach using HMM for a highly inflectional language. Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics, Samos, Greece.

Schellens, P., & Maes, A. (2000). Taalbeheersing als communicatiewetenschap. Een overzicht van theorievorming, onderzoek en toepassingen. In Braet, A. (ed.) *Tekstontwerp,* 29–60. Bussum: Coutinho.

Scott, J. (2017). *Social network analysis*. London: Sage.

Shaw, M. J., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and Data Mining for marketing. *Decision Support Systems, 31*(1), 127–137.

Silva, C., & Ribeiro, B. (2003). The importance of stop word removal on recall values in text categorization. Paper presented at the International Joint Conference on Neural Networks, Portland, OR, USA.

Sirsat, S. R., Chavan, V., & Mahalle, H. S. (2013). Strength and accuracy analysis of affix removal Stemming Algorithms. *International Journal of Computer Science and Information Technologies, 4*(2), 265–269.

Song, Y-Y., & Ying, L. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry, 27*(2), 130.

Štajner, S., & Hulpus, I. (2018). Automatic assessment of conceptual text complexity using knowledge graphs. Paper presented at the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA.

Stell, G. (2015). Towards an integrated approach to structural and conversational code-switching through macrosociolinguistic factors. *Code-Switching Between Structural and Sociolinguistic Perspectives, 43*, 117.

Stone, P. J., & Hunt, E. B. (1963). A computer approach to content analysis: Studies using the general inquirer system. Paper presented at the Spring Joint Computer Conference, Detroit, MI, USA.

Sugumar, R. (2018). Improved performance of stemming using efficient stemmer algorithm for information retrieval. *Journal of Global Research in Computer Science, 9*(5), 1–5.

Tan, A-H. (1999). Text Mining: The state of the art and the challenges. Paper presented at the PAKDD 1999 Workshop on Knowledge Disocvery from Advanced Databases.

Thelwall, M. (2017). The heart and soul of the web? Sentiment strength detection in the social web with SentiStrength. *Cyberemotions,* 119–134. Berlin: Springer.

Uramoto, N., Matsuzawa, H., Nagano, T., Murakami, A., Takeuchi, H., & Takeda, K. (2004). A text-mining system for knowledge discovery from biomedical documents. *IBM Systems Journal, 43*(3), 516–533.

Usai, A., Pironti, M., Mital, M., & Aouina Mejri, C. (2018). Knowledge discovery out of text data: A systematic review via Text Mining. *Journal of Knowledge Management, 22*(1), 1471–1488

Van Deventer, J. P. (2014). *The fundamental building blocks of organisational knowledge management-a statistical evaluation.* University of Pretoria, Pretoria, South Africa.

Van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation, 33*(2), 106–119.

Verwer, S., Eyraud, R., & De La Higuera, C. (2014). PAutomaC: A probabilistic automata and hidden Markov models learning competition. *Machine Learning, 96*(1–2), 129–154.

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for Text Mining – an overview. *International Journal of Computer Science and Communication Networks, 5*(1), 7–16.

Violos, J., Tserpes, K., Papaoikonomou, A., Kardara, M., & Varvarigou, T. (2014). Clustering documents using the 3-gram graph representation model. Paper presented at the 18th Panhellenic Conference on Informatics, New York, NY, USA.

Westgate, M. J., Barton, P. S., Pierson, J. C., & Lindenmayer, D. B. (2015). Text Analysis tools for identification of emerging topics and research gaps in conservation science. *Conservation Biology, 29*(6), 1606–1614.

Widdowson, H. G. (2008). *Text, context, pretext: Critical issues in discourse analysis, 12*. Hoboken, NJ: Wiley.

Wijaya, V., Erwin, A., Galinium, M., & Muliady, W. (2013). Automatic mood classification of Indonesian tweets using linguistic approach. Paper presented at the International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia.

Zhu, L., Gao, S., Pan, S. J., Li, H., Deng, D., & Shahabi, C. (2015). The Pareto Principle is everywhere: Finding informative sentences for opinion summarization through leader detection. *Recommendation and search in social networks,* 165–187. Berlin: Springer.

APPENDIX

The appendices can be found at the following URL: **https://tinyurl.com/y3g3wcj4**