6-2022

Innovative Heuristics to Improve the Latent Dirichlet Allocation Methodology for Textual Analysis and a New Modernized Topic Modeling Approach

Jamie T. Zimmerman

INNOVATIVE HEURISTICS TO IMPROVE THE LATENT DIRICHLET ALLOCATION METHODOLOGY FOR TEXTUAL ANALYSIS AND A NEW MODERNIZED TOPIC MODELING APPROACH

DISSERTATION

Jamie T. Zimmermann, Major, USAF

AFIT-ENS-DS-22-J-059

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-DS-22-J-059

INNOVATIVE HEURISTICS TO IMPROVE THE LATENT DIRICHLET ALLOCATION METHODOLOGY FOR TEXTUAL ANALYSIS AND A NEW MODERNIZED TOPIC MODELING APPROACH

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Major Jamie T. Zimmermann

AFIT-ENS-DS-22-J-059

INNOVATIVE HEURISTICS TO IMPROVE THE LATENT DIRICHLET ALLOCATION METHODOLOGY FOR TEXTUAL ANALYSIS AND A NEW MODERNIZED TOPIC MODELING APPROACH

Major Jamie T. Zimmermann

Committee Membership:

Lance E. Champagne, PhD
Chair

Lt Col John M. Dickens, PhD
Member

Dr. Edward D. White, PhD
Member

Dr. Raymond R. Hill, PhD
Member

Adedeji B. Badiru, PhD
Dean, Graduate School of Engineering and Management

AFIT-ENS-DS-22-J-059

Abstract

Natural Language Processing is a complex method of data mining the vast trove of documents created and made available every day. Topic modeling seeks to identify the topics within textual corpora with limited human input into the process to speed analysis. Current topic modeling techniques used in Natural Language Processing have limitations in the pre-processing steps. This dissertation studies topic modeling techniques, those limitations in the pre-processing, and introduces new algorithms to gain improvements from existing topic modeling techniques while being competitive with computational complexity.

This research introduces four contributions to the field of Natural Language Processing and topic modeling. First, this research identifies a requirement for a more robust “stopwords” list and proposes a heuristic for creating a more robust list. Second, a new dimensionality-reduction technique is introduced that exploits the number of words within a document to infer importance to word choice. Third, an algorithm is developed to determine the number of topics within a corpus and is demonstrated using a standard topic modeling data set. These techniques produce a higher quality result from the Latent Dirichlet Allocation topic modeling technique. Fourth, a novel heuristic utilizing Principal Component Analysis is introduced that is capable of determining the number of topics within a corpus that produces stable sets of topic words.

To my daughter and son ~Work hard in silence and let your success be the noise

Acknowledgements

I would like to express my sincere appreciation to my research advisor Dr. Lance Champagne, for his guidance and mentorship through this process, as well as my research committee members, Lt Col John Dickens, Dr. Raymond Hill and Dr. Edward White for their consistent support and insight. I am grateful for the opportunity to work under their advisement

Lastly, I would like to thank my family for their endless support. I could not have done this without you in my corner.

Table of Contents

Page

List of Figures

List of Tables

Page

INNOVATIVE HEURISTICS TO IMPROVE THE LATENT DIRICHLET ALLOCATION METHODOLOGY FOR TEXTUAL ANALYSIS AND A NEW MODERNIZED TOPIC MODELING APPROACH

I. Introduction

1.1 Motivation

In today's world of big data, managers require tools to help fuse and transform raw data streams into actionable information to meet consumer needs and attain a competitive advantage. Information overload occurs when the amount of input exceeds the processing capacity of the system (Solis, 2020). The human mind is a system. The amount of information/data available far exceeds the processing capacity of an individual. In addition, highly contested and resource constrained environments call for the need to have an accurate and timely answer. Technological advancements have aided analysts' ability to collect, process, exploit and disseminate data; however, there are still critical gaps that further research can address.

Topic modeling is a useful technique as it leverages text to help distill data into usable information. However, text is often messy and unstructured, thereby creating challenges for algorithms that require data cleaning and wrangling to create uniform fixed-length inputs and outputs.

Topic modeling is an unsupervised technique (capable of discovering hidden patterns without human intervention) used to provide insight into textual data. Bag of Words (words within the corpus) and Term Frequency-Inverse Document Frequency (word relevancy) are both methods to assist in determining a topic for a document or corpus. However, left unaltered, these types of methods create a cumbersome dimensionality with the bag of words, which often creates unnecessary noise and unintentionally degrades topic modeling processing and output interpretability.

Additionally, despite advancements in the topic modeling realm, selecting the number of topics for the methods to generate still provides a challenge and requires user input. Using current techniques user must select the appropriate number of topics that accurately reflects the documents. This directly affects the overall results of the analysis. If the user chooses to identify too many topics, the information can become saturated and counterproductive. On the other hand, if the user selects a number that is low, the information may not be specific enough for to the decision maker.

A commonality throughout current topic modeling techniques is the requirement for the user to input the number of topics and number of words to output along with each topic. These parameter inputs have a direct impact on the output of the topic model. Furthermore, it requires the user to have a prior knowledge of the dataset in order to select the optimal topic modeling technique for their dataset and to select the correct values for the inputs. If the user is running a topic modeling technique on a dataset, chances are they will not have the insight needed to make an accurate decision for the parameter values. Excessive decision making can lead to decision fatigue impacting the quality of the decision made. Reducing the algorithm input decisions that are user made reduces the decision fatigue, leading to reproducible results and improve overall algorithm performance.

1.2 Dissertation Overview

This dissertation is organized as follows, Chapters II-IV correspond to the four research contributions in the textual analysis domain, formatted at separate papers, and Chapter V summarizes the contributions along with future research recommendations. Table 1. provides the terms used throughout this dissertation and associate definition to enable a common understanding.

Table 1. Terminology

Word	**Definition**
Word	Basic unit of discrete data
Document	Sequence of words
Corpus	A collection of documents
Stopwords	Words that provide little to no value of the meaning of the document, such as “the”
Topics	A natural grouping of words
Stemming	Converting words to their root
Lemmatization	Groups together the inflected form of a word
Tokenize	Splitting sentences and/or phrases into smaller units
Bag of Words (BoW)	N x V word document matrix where N represents the number of documents and V is the number of words

Chapter II examines the dimensions of the Bag of Words used in the Latent Dirichlet Allocation topic modeling technique and identifies a need and method for a dataset customized stopword list. The new dimensionality-reduction technique, called

Prominent Extraction Technique (PET), employs the total number of words within a document set to produce a higher quality result from the Latent Dirichlet Allocation (LDA) topic modeling technique. The result of the technique illustrates that more data is not always better in topic modeling. Additionally, with our novel culling technique, Coherent Utility Process (CUP), we demonstrated the requirement for a robust stopwords list. When CUP is paired with our bag of word dimensionality-reduction procedure (PET), we report a vastly improved output for the Latent Dirichlet Allocation topic model.

Chapter III examines the current methods used to assist the user in determining the number of topics, *k*, as an input for various topic modeling techniques. The existing techniques could provide multiple numbers to the user, requiring the user to decide which is correct. We developed a heuristic that determines the number of topics for the user as an input into the Latent Dirichlet Allocation topic modeling technique based on the covariance matrix of the transposed term-document matrix.

Chapter IV presents a summary of different topic modeling techniques to include Non-negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) and LDA. Additionally, we propose a new topic modeling technique to address the limitations of requiring the user to input parameter values for number of topics and number of terms per topic, into a topic model and provide a stable output. The new technique only requires the user to input the textual data and any respective custom stopwords list the user may need. The number of topics and number of words associated with each topic is determined by the technique.

Chapter V summarizes the contributions made by this dissertation. The assumptions and limitations of the algorithms and results are discussed, and future research recommendations are provided

II. Mitigating Human Bounded Rationality: A Textual Analysis Approach

2.1 Introduction

The proliferation of data accessible in today's business environment far exceeds the processing capacity of a manager, which leads to a well-studied human condition known as bounded rationality (Cuypers et al, 2021; Tiwana, Wang, Keil & Ahluwalia, 2007; Williamson, 1979). Businesses are continuously facing an increased requirement to handle unstructured textual data (Mendoza, Alegría, Maca, Cobos, León, 2015). With advancements in big data, futuristic mental models of manufacturing are bundled into a concept known as Industry 4.0 where rationally bounded managers are sidelined and automated manufacturing informed by data streams prevail (Benitez, Ayala, & Frank, 2018; Lasi, Fettke, Kemper, Feld & Hoffmann, 2014). Identification of important data can assist managers in their decision making of various marketing strategies (Zhao, 2021). Data mining techniques are becoming increasingly popular as the benefits are recognized as being capable of performing multi-dimensional analysis to help assist in decision making (Tseng & Chou, 2006). Concise summaries of information improve knowledge, assisting in informed decision making (Vemprala, Liu & Choo, 2021).

While today's technology has yet to fully achieve the needs of Industry 4.0, we are at the nexus of these two concepts where managers must process an extreme volume of data to meet the rapid pace of mass customization demanded by consumers. Industry 4.0 has created a momentous push to automate decisions, but managers are still necessary to overcome gaps in data interpretation and decision making that the computer cannot yet fully satisfy (Zawadzki & Zywicki, 2016). Managers operate in a strategic environment.

A strategic environment is created when an individual must consider other individuals' actions/reactions and incentives (Hyndman & Menezes, 2021). Recent developments, such as topic modeling are statistical techniques that can bridge this gap through information redux turning what may otherwise be interpreted as noise into something useful that could provide managers and businesses with a competitive advantage.

Topic modeling is a critical component of natural language processing where documents are modeled as a finite mixture of topics (Wallach, 2006). Topic modeling is useful for document clustering and organizing large blocks of text into useful and actionable information. An effective model will identify words with similar meaning and group them together to form a topic. From product reviews to social media data to informational textual products, topic models can be an effective tool to quickly synthesize data into usable information (Hong & Davison, 2010). However, the inclusion of all words in a body of written texts during topic modeling implementation causes excessive computations to occur, thus adding time and an unnecessary computational expense to successfully execute the algorithm. Additionally, many of the topic models require the user to specify *a priori* the number of topics, *k*, contained in the corpus. Unfortunately, this requires the user to have advanced insight into the data, which is often not possible due to its volume and the competing demands on a manager's valuable time. An additional complication manifests when the number of topics selected has a direct negative influence on the overall output of the model. This modeling flaw creates distortions that unintentionally influence the interpretability of the statistical model, thereby marginalizing its managerial utility (Dahal, Kumar & Li, 2019).

Topic modeling is often complicated by several important factors. The length of the data can range from a limited number of characters, such as a tweet on Twitter, to pages of informational data, such as journal articles. The length of the data will influence the technique(s) implemented for topic modeling (Zuo, Wu, Zhang, Lin, Wang & Xu, 2016). Short text suffers from sparsity and noise (Li, Wang, Zhang, Li, Chi & Ouyang, 2018). Noise in textual data is defined as information that does not provide meaning to the overall intent of the document. Noisy text can also be text that distracts from the original meaning or intent of the text. Consequently, the more noise in a document, the less effective topic modeling algorithms tend to be (Li et al, 2018).

Figure 1 is a broad visualization of the topic modeling process as it exists in literature today. The initiation of topic modeling requires textual input. The dataset then proceeds through a pre-treatment step where the data is cleansed. This purification step may include punctuation removal, stopwords elimination, converting words to lower case, stemming and/or lemmatization. During this step, to save time, the user can leverage software packages with pre-identified stopwords, additionally the user may specify their own stopwords, if desired. After textual pre-processing, a topic modeling method is selected and implemented. The output consists of words associated with k topics, where k is the number of topics.

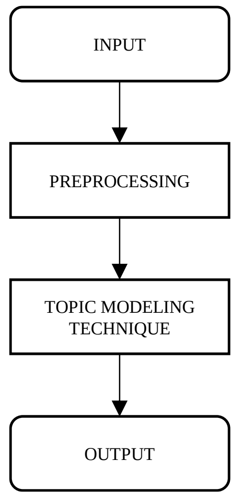


Figure 1. Overview of Traditional AdHoc Topic Modeling

There have been many advancements in the methods of topic modeling (Anthes, 2010; Mustak, Salminen, Plé & Wirtz, 2021). Despite such progress, dimensionality continues to be a challenge for text mining (Singh, Devi, Devi & Mahanta, 2022) leading to overfitting (Yin & Shen, 2018). To overcome this challenge, we offer several compelling contributions to both academics and practitioners. These contributions are CUP, PET, Eigenvalue heuristic for determining *k* and the Zimm Approach. Figure 2 is a visualization of the proposed process for topic modeling presented in this paper.

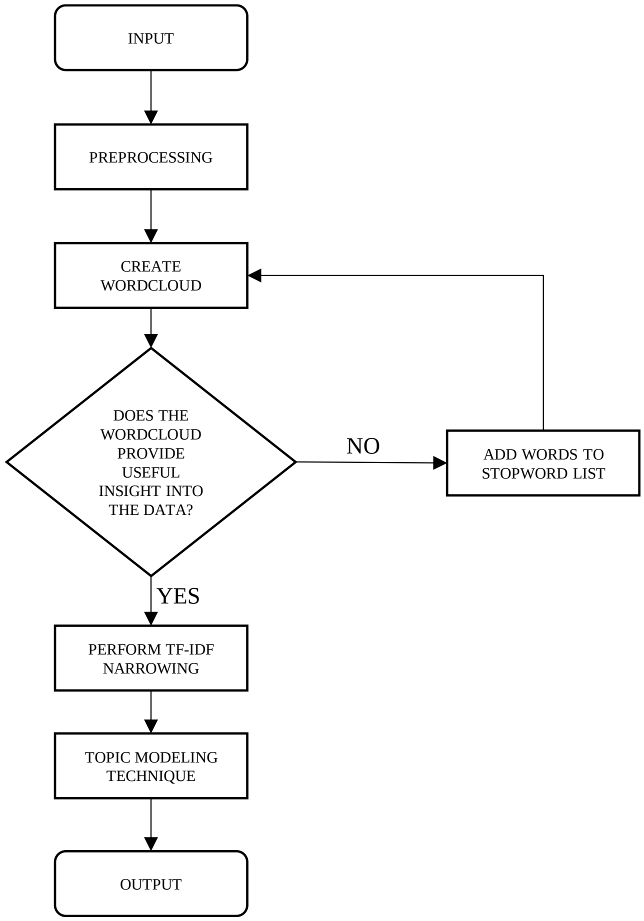


Figure 2. The Proposed Process for Topic Modeling

2.2 Background

Over the years, various literature has indicated that researchers are interested in exploring and applying a variety of machine learning techniques to solve analytic challenges involving textual data. Every word, in a document, may be treated as an attribute (Martins, Monard & Matsubara, 2003). The attribute-value representation may have critical influences on the topic model.

Textual analysis includes various strategies and techniques to transform raw communication data into actional intelligence (Brahma, Goldberg, Zaman & Aloiso, 2021). Text mining is defined as “the application of algorithms and methods from the fields of machine learning and statistics to texts with the goal of finding useful patterns” (Groth & Muntermann, 2011). This section discusses underlying methods that currently exist, which we are going to improve upon, to model topics.

2.2.1 Word Clouds

A word cloud is a visualization tool that allows the user to see the most frequent words in a document/collection of documents. In a word cloud, the size of the word is related to the frequency of the word within the corpus. Chae and Olson (2021) looked at the evolution of topics since 1975. The authors used word clouds as a visualization method to show word changes in abstracts in four time periods: 1975-1985, 1986-1995, 1996-2005 and 2005-2016. The visualization tool successfully illustrated that there were some key changes among the abstracts such as the topics of journals shifting from quantitative modeling methods to supply chain management.

Word clouds can be useful if the user needs to do a quick look to determine if keywords are part of the document(s). However, depending on the context of the information, a word cloud may not accurately capture and communicate important insights about the text. Important concepts about the textual dataset can be left in the shadows if the corpus author favors certain verbiage.

2.2.2 Bag of Words

Bag of Words (BoW) is a representation of the words within a document. It is a vector representation of the document where each element is the normalized number of occurrences of the term in the document (Zhao & Mao, 2017). During the computations, sequential information is not maintained (Lebanon, Mao & Dillon, 2007). BoW is used as an input in many topic modeling techniques, such as LDA.

While the bag of words is used to represent a corpus, there are limited theoretical studies on the properties of the bag of words (Zhang, Jin & Zhou, 2010). BoW suffers from high dimensionality (Zhao et al., 2017). BoW can reach many thousands of potential predictors to assist in topic modeling (Geva & Zahavi, 2014). Passalis and Tefas (2016), Zhao et al. (2017), Ljungberg (2019) and Boulis and Ostendorf (2005) addressed high dimensionality within the textual analysis domain however, their techniques still had room for improvement to be made.

Geva & Zahavi (2014) used preprocessing techniques, such as stemming and stopwords list filtering, to reduce the dimensionality of the BoW. Their technique led to the need to select a specified top number of words. Despite efforts made to improve the bag of words input, a methodology for bounding the Term Frequency-Inverse Document Frequency (TF-IDF) (see Section 2.2.3) technique has not been addressed. This article employs a novel approach to narrow the bag of words used in topic models based on the TF-IDF in addition to introducing a process to select words to create a unique, dataset specific stopwords list.

2.2.3 Term Frequency-Inverse Document Frequency

TF-IDF is a methodology for representing ratio of word counts in a document and indicates the importance of a word to the document and/or corpus. The higher the TF-IDF, the more important the word. To calculate TF-IDF, a count of the number of occurrences of each word in a document (contained in the corpus) is compared to an inverse document frequency count. The inverse document frequency count measures the count of the word in the entire corpus.

2.2.4 Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) model is a generative probabilistic model for the collections of discrete data (Blei, Ng & Jordan, 2003). LDA uses the words in the document to identify the topic(s) that the document belongs to. There are three user inputs into the LDA modeling method: alpha, beta and *k* (Binkley, Heinz, Lawrie & Overfelt, 2014). The output of the LDA model is a list of topics and words with the associated probability that the word belongs to that topic. LDA does not require previous training data and can handle mixed length documents, although for short messages, it needs an aggregation of the messages to avoid data sparsity (Albalawi et al, 2020). The goal of LDA is to find topics for the document collection (Slof, Frasincar, Matsiiako, 2021).

A key assumption of LDA is the bag of words will preserve most of the relevant information (Hoffman, 2001). Additionally, the order of words and sentence structure (i.e., grammatical role of the word) is not considered in the model, therefore word ordering is unimportant (Misra, Cappé & Yvon, 2008). LDA also assumes all documents contain a mixture of topics (Feuerriegel & Pröllochs, 2021), meaning the documents contain assorted topics and the words within the documents are generated from the topics.

LDA has been applied to a wide range of discipline areas when looking at the application of topic modeling. Feuerriegel and Pröllochs (2021) used LDA to study how financial disclosures, across assorted topics, effected stock prices. Chae and Olson (2021) used LDA to understand the topic structure of the *Decision Sciences* journals, correlation of topics and how the topics have evolved since 1975. While LDA is a popular topic modeling technique however, the number of topics, *k*, for the model to identify, must be specified by the user (Fu, Zhuang, Gu, Zhu, Qin & Guo, 2019). This requires the user to have some understanding of the corpus prior to implementing the algorithm. LDA is less prone to overfitting and capable of inferring topics for unobserved documents than other techniques (Yan, Guo, Liu, Cheng & Wang, 2013); therefore, LDA is the topic modeling method of choice for this article.

The rest of the article is organized as follows: discussion on fundamentals of the visualization utilization to create a stopwords list specific to the dataset and TF-IDF narrowing approach in the Methodology section; discussion on the analysis of dataset and results in the Analysis section and finally the conclusions and potential future areas of interest.

2.3 Methodology

There is a low probability that stopwords will contribute to the overall topic modeling of the corpus (Feuerriegel & Pröllochs, 2021). This idea supports the justification for needing a solid stopwords list unique to each dataset. In the proposed topic modeling process the input, textual data, remains the same, and the user/algorithm still performs preprocessing to cleanse the data. Subsequently, a word cloud is created to help identify the main topic and potential subtopics of the dataset. If the word cloud does not consist of excessive noise, then the TF-IDF narrowing technique is performed and fed into the selected topic modeling. If the word cloud contains excessive noise, the user creates a unique stopwords list to assist in noise filtering, which is fed back into the creation of a new word cloud for the user to iteratively examine. This is a novel procedure that we identify as the Coherent Utility Process (CUP).

The CUP is an iterative process that is complete once the user is satisfied that enough noise has been eliminated from the word cloud to generate insights. Additionally, we present a new dimensionality-reduction technique, called the Prominent Extraction Technique (PET), that uses the number of words within a document set to produce a higher quality result from Latent Dirichlet allocation (LDA) or other topic modeling techniques. The resulting dimensionality reduction utilizes the LDA topic modeling in the evaluation criteria to test and analyze the effects of narrowing the BoW based on the Term Frequency-Inverse Document Frequency (TF-IDF) values with the removal of stopwords, utilizing both premade and custom lists. By doing so, this contribution enables managers to effectively right-size the bag of words to achieve a level of utility not previously possible. Discussions of the data, preprocessing, the proposed Coherent

Utility Process (CUP), and the proposed Prominent Extraction Technique (PET) follow in this section.

2.3.1 Data and Preprocessing

Our research used a subset of 20newsgroup, a collection of 11,314 text files of seven subjects, labeled for topics and subtopics. Specifically, we used the baseball topic of the dataset.

We performed common pre-processing steps: lower case, removal of special characters, digits, stopwords (using python preloaded package), stemming (Schofield & Mimno, 2016) and lemmatizing (Balakrishnan & Lloyd-Yemoh, 2014). The most popular stemming algorithm is the Porter Stemmer (Razmi, Zamri, Ghazalli, & Seman, 2021), while the Lancaster Stemmer is a more aggressive stemmer (Razmi et al., 2021); therefore, the Porter Stemmer was utilized. Lemmatizing algorithms are generally slower than stemming because rule-based methods proceed through the corpus to find relevant word associations (Jivani, 2011). The WordNetLemmatizer from the Natural Language ToolKit is used.

After these pre-processing steps, we created word clouds and a BoW for which word frequency and TF-IDF were calculated. These measures are used in the CUP and PET approaches for topic discovery, discussed in the following section.

2.3.2 New Approach Proposal

Some corpora are noisy, meaning they contain information irrelevant to the user specific needs (Rogers, ADrozd & Li, 2017). This noise affects the topic modeling output. The initial step to reduce this noise is a visualization of the word cloud for the data. This visualization will provide the user with a means of identifying words that do not add value in providing insight into the data. CUP is used to identify irrelevant words in the corpus and is used to create a unique, data-specific stopwords list, thereby removing the noise from the dataset. Once the additional irrelevant words are removed, an objective technique for narrowing the BoW, called PET, can more effectively be applied.

Despite the modern sparse techniques, topic discovery is still a challenge due to the high dimensionality of the underlying space (Doshi-Velez, Wallace & Adams, 2015). An approach to reduce the dimensionality provides more accurate results for topic modeling in both a visualization approach and utilizing the LDA topic modeling technique.

According to Eassom (2017) effective keywords should be mentioned every 100 to 200 words in a journal article. Therefore, the total word count divided by 100 and 200 is utilized in the equations for PET. Equations 1 and 2, respectively, show the calculation for lower and upper bounds on word frequency:

$$\frac{w}{200} - (w * .10) \tag{1}$$

$$\frac{w}{100} + (w * .10) \tag{2}$$

where w = the total number of the words in the BoW

Both the lower bound (equation 1) and the upper bound (equation 2) were rounded down and up, respectively, to the nearest whole number. After calculating a lower and upper bound for the word frequency, the minimum and maximum TF-IDF values within that word frequency range was used to create the narrowed/reduced BoW.

A space filling screening design was created varying the percentage of BoW words either added or subtracted from the upper and lower bounds, respectively. The design used percentages from 0 to 20, with increments of 0.025. After completing the analysis, 0.10 provides a reasonable calculation without overestimating the word count bounds used in determining the minimum and maximum TF-IDF values. Therefore, we chose 0.10 when creating the BoW for the LDA topic modeling technique.

2.3.3 Algorithm Evaluation Criteria

The evaluation of true effectiveness of informational retrieval relies on the user expectations and/or needs (Taghva, Borsack, Condit & Erva, 1994). This research used word clouds, coherence score and the overall output of the LDA model as evaluation criteria for algorithm effectiveness.

2.4 Analysis and Results

Topic modeling includes understanding the words within the topics and the similarity between the topics. While there exist a variety of techniques to produce a score, such as the coherence score, these techniques are only part of the overall topic modeling process. The user should be able to interpret, comprehend and formulate the topic(s) of the dataset based on the model output. Too many words produce noise thus adding confusion for the user and topic modeling technique. This analysis illustrates that our novel culling techniques provide more discrimination, with greater dataset interpretability and clarity. Appendix A provides the algorithm for CUP and PET. The TF-IDF files were exported to excel where the narrowing calculations were performed. The narrowing bounds were inputs into the python code.

2.4.1 Results

If a user needs a quick visual for most frequent words in a dataset the word cloud tool provides this capability, since the more frequent a word appears in the corpus, the larger its corresponding representation in the word cloud. The mere frequency of a word may not provide the user with true insight into what important topic(s) are contained within that dataset therefore not all words should be used when creating a final word cloud for a user to use for decision making.

The first step in the proposed process requires the user to create and analyze a word cloud for useability. Figure 3 represents word frequency from the dataset using the full data set and Python's stopwords package for the baseball dataset.

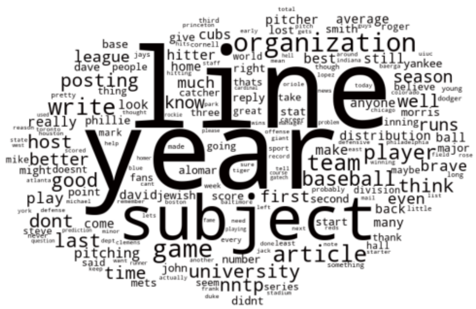


Figure 3. Word Cloud of Baseball Dataset

With a cursory viewing of Figure 3, the general topic of the dataset is not evident because extraneous words relating to the data format (i.e., email) dominate. The words that appear larger are more general words, providing little additional information about the dataset. However, with closer inspection to less prominent words in the cloud, there is an indication that the dataset may be about a sport.

Similarly, we conducted the topic modeling process without the additional TF-IDF narrowing process using only the prestored Python stopwords package. Figure 4 displays the LDA output for the baseball dataset. As was the case with the word cloud, the words assigned by the LDA topic modeling technique do not provide the user clarity into the dataset because general words are dominating the topic-specific words.

Topic: 0
words: ['lines', 'subject', 'organization', 'article', 'game', 'writes', 'university', 'think', 'nntp', 'baseball']
Topic: 1
words: ['subject', 'organization', 'lines', 'players', 'writes', 'baseball', 'good', 'year', 'team', 'university']
Topic: 2
words: ['subject', 'organization', 'lines', 'year', 'article', 'writes', 'would', 'team', 'last', 'good']
Topic: 3
words: ['organization', 'year', 'lines', 'subject', 'article', 'writes', 'dont', 'good', 'team', 'university']
Topic: 4
words: ['lines', 'subject', 'article', 'writes', 'year', 'organization', 'posting', 'baseball', 'game', 'dont']

Figure 4. LDA output for Baseball Dataset

When applying PET to the baseball dataset, the TF-IDF range did not narrow, i.e., the entire BoW were still being used. Therefore, we moved directly into the CUP technique.

By following the CUP technique, the following words were added to the baseball dataset stopwords list:

> from, re, subject, would, organization, university, year, line, better, well, still, like, nntp, think, dont, good, writes, might, know, much, give, article, even, last, anyone, make, time, look, play, season, come, said, great, didnt, back, maybe, going, rally, reply, though, many, years, thats, best, lines, game, team, player.

A word cloud was created to ensure the CUP technique was beneficial to the overall analysis. Figure 5 displays the word cloud for the dataset. Now, because of our culling technique, the user can now identify more insightful details about the datasets prior to PET (TF-IDF narrowing).

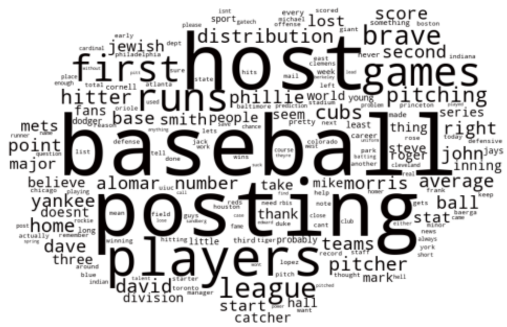


Figure 5. WordCloud for Baseball Dataset using Custom Stopword List

With the employment of our CUP technique, the LDA output has also subjectively increased in fidelity. Figure 6 displays the LDA output for each instance.

Topic: 0
words: ['baseball', 'players', 'host', 'posting', 'games', 'jewish', 'braves', 'cubs', 'pitching', 'could']
Topic: 1
words: ['baseball', 'games', 'posting', 'host', 'david', 'players', 'lost', 'braves', 'philadelphia', 'league']
Topic: 2
words: ['posting', 'host', 'runs', 'first', 'games', 'baseball', 'braves', 'dave', 'david', 'also']
Topic: 3
words: ['host', 'baseball', 'posting', 'players', 'runs', 'games', 'morris', 'pitching', 'first', 'michael']
Topic: 4
words: ['runs', 'baseball', 'first', 'posting', 'games', 'players', 'host', 'league', 'second', 'phillies']

Figure 6. LDA output for Baseball Dataset using the Custom Stopword List

When the user utilizes the unique stopwords list that emerges from the CUP technique the user is provided with more insight into the dataset. To continue providing more details, the unique stopwords list was combined with the BoW dimensionality reduction technique (PET). Prior to PET, the TF-IDF range was [0.000751, 0.047811], after applying PET, the TF-IDF range was narrowed to [0.025573, 0.046691].

Figure 7 shows the results of the word cloud creation after the new process is utilized. For example, the baseball dataset now shows that teams such as braves, cubs, mets and phillies are discussed in the dataset; information that was not prevalent prior to employing our culling techniques (CUP/PET). We believe that when compared to the less filtered word cloud in Figure 7, our culling techniques provide more utility and insight into the dataset.

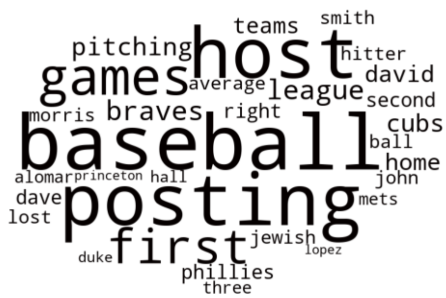


Figure 7. Word cloud when PET applied to Baseball Dataset using CUP

Table 2 displays the coherence scores of two methods. While the coherence score does not directly relate to human interpretability, the coherence score improved with the custom stopwords list and dimensionality reduction technique (CUP/PET). Improving the coherence score by 10.9% paired with the improved ability to gain insight into the dataset makes these two processes look promising.

Table 2. Coherence Scores Comparing the Four Methods

	A	B
Coherence Score	0.5017	0.5563

where,

A = No unique stopword list, no TF-IDF Narrowing

B = Unique stopword list, TF-IDF Narrowing

Figure 8 displays the output when pairing LDA with CUP and PET. The dataset also contains information about specific teams and baseball players. This level of detailed information was not visible in the output in Figure 4.

Topic: 0
words: ['first', 'posting', 'three', 'host', 'david', 'also', 'mets', 'lopez', 'hall', 'could']
Topic: 1
words: ['posting', 'host', 'first', 'baseball', 'braves', 'teams', 'phillies', 'games', 'morris', 'pitching']
Topic: 2
words: ['games', 'average', 'league', 'dave', 'ball', 'john', 'baseball', 'david', 'right', 'hitter']
Topic: 3
words: ['posting', 'host', 'cubs', 'pitching', 'smith', 'duke', 'games', 'braves', 'hall', 'princeton']
Topic: 4
words: ['baseball', 'jewish', 'could', 'alomar', 'home', 'lost', 'also', 'league', 'phillies', 'posting']

Figure 8. LDA Output with PET is applied to Baseball Dataset with CUP

When CUP and PET are applied to the Baseball dataset, the word cloud and LDA output provides more insight into the data, directly stating the baseball players names and teams.

Additionally, Figure 9 shows an overall improvement on the coherence scores for *k* ranging from one through five when using CUP and PET.

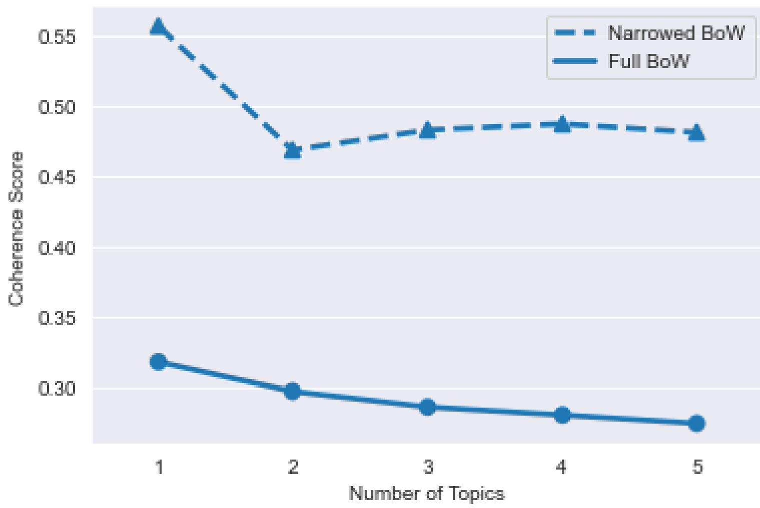


Figure 9. Coherence Score Comparison

2.5 Conclusions

As the amount of textual data available to decision makers continues to increase, textual analysis will become a primary fulcrum for high performing managers. However, as explained in this research, there are many varying factors that can influence the output of the topic model. Most importantly, the quality and quantity of data fed into the models is a critical aspect towards maximizing the value and interpretability of the results. Technological improvements and advanced computing capacity have enabled vast amounts of data to be analyzed quickly; however, as the data becomes more complex and disparate, the quality of inputs can quickly and unintentionally degrade the model outputs. This presents an interesting challenge for data managers and decision makers.

The results of our research answer this important managerial and academic need and serve as a foundational step in this critical area of the topic modelling literature.

In this chapter, we developed and articulated several processes to enhance textual mining. First, we introduced a subprocess for enhancing stopwords, which we identify as CUP. Then, we presented a new dimensionality-reduction technique, we identify as PET, that uses the number of words within a document set to produce a higher quality result from the LDA topic modeling technique. These new culling techniques employ a visualization tool for the user to identify additional stopwords and establish a new upper and lower bound for TF-IDF scores. By doing so, these contributions enable managers to effectively right-size the bag of words to achieve a level of utility not previously attainable.

A brief comparative analysis using our techniques provided a more diverse set of words within each of the *k* topics, which should provide an increased ability to discern specific topics. Our research shows that this result holds for multiple data sets and is therefore promising as a new way to process topics within a body of literature.

III. Heuristic for Determining Number of Topics, *k*

3.1 Introduction

Data science is used to support and improve decision making processes (Coussement, Kristof & Dries Benoit, 2021). The average American adult makes approximately 35,000 decisions a day (Sollisch, 2016). After a while, an individual experiences decision fatigue. Decision fatigue is symptom of ego depletion and/or depleted state of internal resources (Pignatiello, Martin & Hickman Jr., 2020). When decision fatigue occurs, the quality of the decision declines (Hirshleifer, Levi, Lourie & Teoh, 2019). Analysts can experience decision fatigue. This demonstrates the need for more effective heuristics to aid / make routine decisions. Additionally, a more streamline decision making process is imperative for reproducible and stable results.

In a data-driven society, the number of textual datasets continues to grow (Dutta & Gupta, 2022). This growth has led to an increase in information a human is expected to review. Data-driven decision making is a key concept for supporting decisions (Röder, Palmer & Muntermann, 2022). Human beings have limited resources such as the ability process, clean and analyze the various data points affecting decisions. The need to streamline textual analysis techniques continues to grow at an exponential rate.

When discussing document content, topics must first be identified. A topic is identified as a natural grouping of words. The length of the text influences the technique selected for topic modeling (Albalawi, Yeap & Benyoucef, 2020). If the text is short or a single document, a simple word frequency approach may be useful.

A useful topic model is one that models the corpus contents in a stable fashion. Stable meaning that no matter the input representation or model parametrizations, the results are still useful topics (De Waal & Barnard, 2008). In efforts to produce a stable model, parameters need to be optimized for each topic modeling technique. If a modeling technique requires a user to input a parameter, such as *k* (number of topics), this could cause the model to become unstable.

Latent Dirichlet Allocation (LDA) is one topic modeling technique. It utilizes the Dirichlet prior. Gerlach, Peixoto & Altmann (2018) stated that topic models suffer from conceptual and practical problems. Specifically mentioned were, intrinsic methodology to choose the number of topics, a large number of free parameters that may lead to overfitting and no justification (besides mathematical convenience) as to why the Dirichlet prior is utilized in the model. LDA requires the user to specify *k*, the number topics, for the algorithm to generate, requiring significant input from domain experts (Fu, Zhuang, Gu, Zhu, Qin & Guo, 2019).

There have been many advancements in the methods of topic modeling. Despite these advancements, selecting the number of topics for topic modeling methods to create still provides a challenge and requires user input (Kherwa & Bansal, 2020). A user must select the appropriate number of topics that accurately reflects the documents. This directly affects the overall results of the analysis. If the user selects a sparse number of topics, the risk of "too broad" of topic identification occurs however if the user selects a high number of topics, the risk of "over-clustering" is present (Greene, O'Challaghan & Cunningham, 2014). This research develops an eigenvalue heuristic to determine the appropriate number of topics, *k*.

3.2 Background

A common way of modeling topics is to treat each topic as probability distribution over words (Griffiths & Steyvers, 2004). If there are T topics then the probability of the *i*th word, in a given document, is written as

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j) \tag{3}$$

where,

z_i is a latent variable indicating the topics from which the *i*th word was drawn

$P(w_i|z_i = j)$ is the probability of the word w_i under the jth topic

$P(z_i = j)$ is the probability of choosing a word from topics j in the current document.

Two assumptions common throughout most of the models are: 1) k is known and fixed, and 2) the words are infinitely exchangeable as are the topics within the document (Xu, Heller, Ghahramani, 2009). Given the exponential growth of digital datasets and the growth of information extraction (Hogenboom, Frasincar, Kaymak, De Jong & Caron, 2016), many techniques have been developed to determine the number of topics for various topic models. This section discusses techniques used and the respective topic modeling techniques.

3.2.1 Graph Dimensionality Selection Techniques

Graph based dimensionality selection or the number of topics, k, has been used in methods like Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) where the natural indicator is the eigenvalue. Fu et al (2019) showed that SVD and PCA produced comparable results when determining the optimal numbers of topics.

Fu et al (2019) used the elbow point in a scree plot to identify to the optimal number of topics. The elbow method utilizes k-mean clustering on input data for a given number of clusters, *k*. The sum of squared errors is calculated for each cluster. The sum of squared errors is the distance of all data points to their respective cluster center. After plotting the number of clusters by the sum or squared errors, take the point in which the sum of squares decreases abruptly and add one, this is the ideal number of topics. Fu et al (2019) noted their findings was based on specific textual data. The heuristic proposed in this chapter is intended for a variety corpus and is based on the term-document matrix.

PCA is a multivariate technique that extracts information and represents the information as a set of new orthogonal variables called principal components and then display a map that shows pattern(s) of similarity of the observations (Abdi & Williams, 2010). PCA tries to identify major components embedded in the data matrix. This technique reduce noise data since the maximum variation source is selected and the small variations are ignored. In PCA, principal components are exact linear transformations of the data without considering residual error (Péladeau & Davoodi, 2018). The heuristic in this chapter uses PCA.

3.2.2 Bayesian Methods

In 2004, Griffiths and Steyvers used Bayesian model selection to determine the number of topics. A Bayesian classifier assumes all words in the document come from a single class (Griffiths & Steyvers, 2004). This is not always the case. An input can come from multiple classes (Murphy, 2006).

Griffiths & Steyvers (2004) looked at the effects of changing the number of topics, utilizing the Gibbs sampling algorithm. The Gibbs sampling algorithm is a Markov chain Monte Carlo, a stochastic process for computing and updating α and β (Agrawal, Fu, & Menzies, 2018). The Griffiths & Steyvers (2004) dataset was comprised of 28,154 abstracts published in PNAS from 1991 to 2001. In LDA, two other input parameters are α and β. A high α indicates that every document is likely to contain a mixture of most topics and not a single topic. A low α indicates that a document is more likely to represent one or just a few topics. A high β each topic is likely to consider most of the words and not any word specifically. A low β each topic may contain a mixture of only a few words. The value of α and β affect the optimal number of topics therefore during the experiment, $\alpha = 50/k$ and $\beta = 0.1$ were fixed and *k* was varied using Bayesian statistics. The optimal number for *k* is selected based on the log-likelihood of the data.

While Griffiths & Steyvers (2004) proposed an approached to determine *k*, varying *k* and computing/graphing calculations were still required. This requires the user to know a range in which to vary *k* and know how to understand/interrupt the results of the graphs. There is potential for the optimal value of *k* to fall outside of the range in which the user selects to test. Our proposed heuristic does not require comparisons of various computations by varying *k*.

3.2.3 Stability Analysis

Greene et al (2014) proposed a term-centric stability analysis strategy to address the issues of selecting the appropriate number of topics as an input to the Non-negative Matrix Factorization (NMF) topic modeling technique, *k* in [$k_{min,}$ k_{max}]. Let S denote the i^{th} topic produced by the algorithm list R_i, i.e S={R_1,…R_k} where *k* is the number of ranked lists. In NMF this will correspond with the highest ranked values in each column of *k* basis vectors (Green et al, 2014). Jaccard similarity can be used to measure the similarity between two top words of any two topics. If two topics have the same top word then the Jaccard measure would be 1 and if all top words were different then the Jaccard measure would be 0 (Mantyla, Claes, & Farooq, 2018). The Jaccard index does not account for positional information. In other words, terms that are listed at the top of a ranked list will naturally be more relevant to a topic than those at the end of the list (Greene et al, 2004). To alleviate this problem, Greene et al (2014) utilized a ranking distance measure proposed by Fagin et al (2003).

Greene et al (2014) referred to Fagin et al's (2003) approach as the Average Jaccard (AJ) approach. The AJ approach is used to analyze the similarities between a pair of ranked lists (R_i, R_j). AJ is a top-weighted version of the Jaccard index.

$$AJ\left(R_i, R_j\right) = \frac{1}{t}\sum_{d=1}^{t} \gamma_d(R_i, R_j) \tag{4}$$

where,

$$\gamma_d\left(R_i, R_j\right) = \frac{|R_{i,d} \cap R_{j,d}|}{|R_{i,d} \cup R_{j,d}|} \tag{5}$$

produces a value between [0,1]

$$stability(k) = \frac{1}{\tau}\sum_{i=1}^{\tau} agree(S_0, S_i) \tag{6}$$

where,

τ: number of samples of dataset that are construct by randomly selecting a subset of β x n documents without replacement

0≤β≤1 : sampling ratio controlling the number of documents in each sample

$$agree\left(S_x, S_y\right) = \frac{1}{k}\sum_{i=1}^{k}(AJ(R_{xi}, \pi(R_{xi})) \tag{7}$$

where,

$$S_x = \{R_{x1}, \dots, R_{xk}\}$$

$$S_y = \{R_{y1}, \dots, R_{yk}\}$$

A plot of the stability scores is created. The final value of *k* will be based on the peaks of the plot. If more than one peak exists, then that may indicate that the corpus can be associated with more than one topic. If more than one peak exists, then the user still has to make a decision on the value for *k*, thus no longer removing the decision-making requirement.

3.2.4 Coherence Scores and Perplexity

Topic coherence measures are a qualitative approach to automatically uncover the coherence of a topic (Syed & Spruit, 2017). It scores a single topic by measuring the degree of semantic similarity between high scoring words in the topic. The measures assist in differentiating between topics that are semantically interpretable and topics that are artifacts of statistical inferences (Stevens et al, 2012). Topics are “coherent” if all or most of the works are related if they support each other.

Common topic coherence measures are UCI measure (Newman, Noh, Talley, Karimi & Baldwin, 2010), UMass measure (Mimno, Wallach, Talley, Leenders & McCallum, 2011), and Coherence Value (*C_v*) (Rőder, Both and Hinnerburg, 2015). These measurements have been shown to reflect human judgement when referencing topic quality (Stevens et al, 2012). UCI and UMass measures compute the coherence of a topic as the sum of a pairwise distributional similarity scores, as in formula 8,

$$Coherence(V) = \sum_{(v_i, v_j) \in V} score(v_i, v_j, \epsilon) \tag{8}$$

where V is a set of words describing the topics and ϵ is the smoothing factor to guarantee that score returns real numbers. The value of ϵ is set to 1 however Stevens et al (2012) looked at the effects of varying the value. Newman, Lau, Grieser and Baldwin (2010) showed coherence scores based on Pointwise Mutual Information (PMI) and Normalized Pointwise Mutual Information (NPMI) have the highest correlation with human judgement in topic evaluation (Hamzeian, 2021).

The UCI measure defines the score to be a pointwise mutual information (PMI) between two words, as shown in formula 9. It can also be thought of as an external comparison to known semantic evaluations (Stevens et al, 2012).

$$score\left(v_i, v_j, \epsilon\right) = \log \frac{p(v_i, v_j) + \epsilon}{p(v_i, v_j)} \tag{9}$$

The UMass measure defines the score to be based on document co-occurrence (Stevens et al, 2012), as shown in formula 10. This measure uses the counts over the original corpus used to train the topic models, rather than the external corpus as in the UCI measure leading this metric to be more intrinsic in nature.

$$score(v_i, v_j, \epsilon) = \log \frac{D(v_i, v_j) + \epsilon}{D(v_j)} \tag{10}$$

Where $D(x, y)$ counts the number of documents containing x and y words and $D(x)$ counts the number of documents containing x (Stevens et al, 2012).

Aletras and Stevenson (2013) showed NPMI was better than PMI for correlating with human judgement. NPMI reduces the impact of low frequency counts in word co-occurrences thus utilities more reliable estimates (Bouma, 2009) thus leading to the improvement of NPMI over PMI.

Rőder et al (2015) looked at the top word of a topic instead of defining probabilities over word pairs (Hamzeian, 2021). The Coherence Value (C_v) measure combines the indirect cosine measure with the NPMI and the Boolean sliding window (Rőder et al, 2015).

Statistical measure of perplexity or likelihood of test data has been the method of choice for evaluation of topic models (Newman et al, 2010). Zhao et al. (2015) used perplexity scores to assist in determining the optimal number of topics for the LDA model. Perplexity was defined as

$$perplexity(D_{test}) = \exp \{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\} \tag{11}$$

where D is the corpus containing M documents d having N_d words ($d \in \{1,\ldots M\}$). The point in which the rate of the perplexity changed, that was determined to be the optimal number of topics. The perplexity measure does not reflect the semantic coherence of individual topics nor does it provide indication to the user of the topic model's performance. It has been suggested that perplexity measures are contrary to human judgement (Jiang et al, 2017).

While all these methods provided the researchers with promising results, the potential for multiple peaks still exists. Therefore, these techniques still required the user to make a decision on which peak they should select. This chapter introduces a heuristic that removes the requirement for the user to make the decision and provide the number of topics as an immediate input into the Latent Dirichlet Allocation Model.

3.3 Methodology

LDA is the most common used topic modeling method (Zhao et al, 2015). It is a generative probabilistic model with the intent to uncover hidden thematic structures of a corpus (Syed & Spruit, 2017). LDA was recently used by Zamani et al (2020), to assist in the identification in the societal shifts in concerns on COVID-19.

One of the important inputs into the LDA model is *k*, the number of topics for which the model will generate. This variable is a user specified number. If the number for *k* is too high, the topics may merge or be uninterpretable however, if the number for *k* is too low, the topics may be too broad or not enough (Syed & Spruit, 2017). The number of topics effects the overall quality of the LDA model output.

3.3.1 Data and Preprocessing

Our research used a subset of datasets from 20newsgroup, specifically, a varied combination of collection of 11,314 text files of seven subjects, labeled for topics and subtopics. The text documents were put through various pre-processing algorithms for stemming, lemmization, removal of symbols, punctuation and stopwords using preloaded python packages.

3.3.2 New Heuristic Proposal

Röder, Both and Hinneburg (2015) introduce a coherence score measure, C_v, which achieves the highest correlation with all available human topic ranking. LDA was selected at the topic modeling technique and implemented, varying *k* to compute the coherence scores. After the coherence scores are calculated and plotted, the results are compared to the proposed technique in the analysis section.

The coherence score technique requires the user to input *k* to calculate the results, plot the various scores among a user specified number of unique *k*'s and then determine the optimal number of topics. This is resource intensive and requires the user to interpret the plot or output of coherence values. In addition, a couple of challenges are immediate with this approach: 1) what range of *k* should the user specify to test for the optimal *k* and 2) what happens if there exists more than one peak?. Figure 10 shows an example of a coherence score plot where the coherence score peak is the same for values 4 and 8. The user would then have to decide which number to use as an input into the model. The goal is to minimize the decision making required for the user, thus lowering the opportunities for analyst reaching decision fatigue.

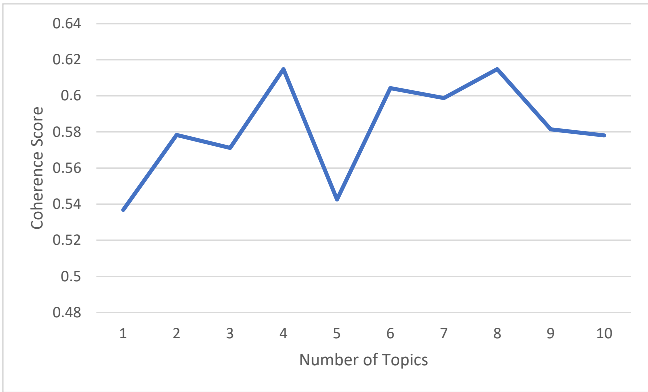


Figure 10. Coherence Score Example, peak at two places

A heuristic using the eigenvalues of the covariance matrix of the term-document matrix is proposed to determine the number of topics. A term-document matrix is a table consisting of a frequency of each term in each document. A row is each term and the columns are each document, while the entry is the frequency of the term in a document.

The proposed heuristic utilizes the term-document matrix, providing an answer that will be fed directly into the LDA topic modeling technique. This eliminates the requirement for a user to manually enter the number of topics and make decisions based on a dataset that he/she may not have insight into.

Initially, looking at the scree plot and finding the point of maximum curve was tested. This approach did not result in accurate results when tested on data that the number of topics were known. The proposed heuristic identifies the number of topics being equated to the number of eigenvalues, of the covariance matrix of the term-document matrix, greater than one. Appendix B provides the algorithm for the eigenvalue heuristic as well as the LDA and coherence score algorithms used in the analysis.

3.4 Analysis and Results

The eigenvalue heuristic was applied to a variety of datasets containing one through five main topics. This research did not look at the possibility of subtopics being identified. This heuristic focused on obtaining a value for *k* as the input parameter into the LDA model.

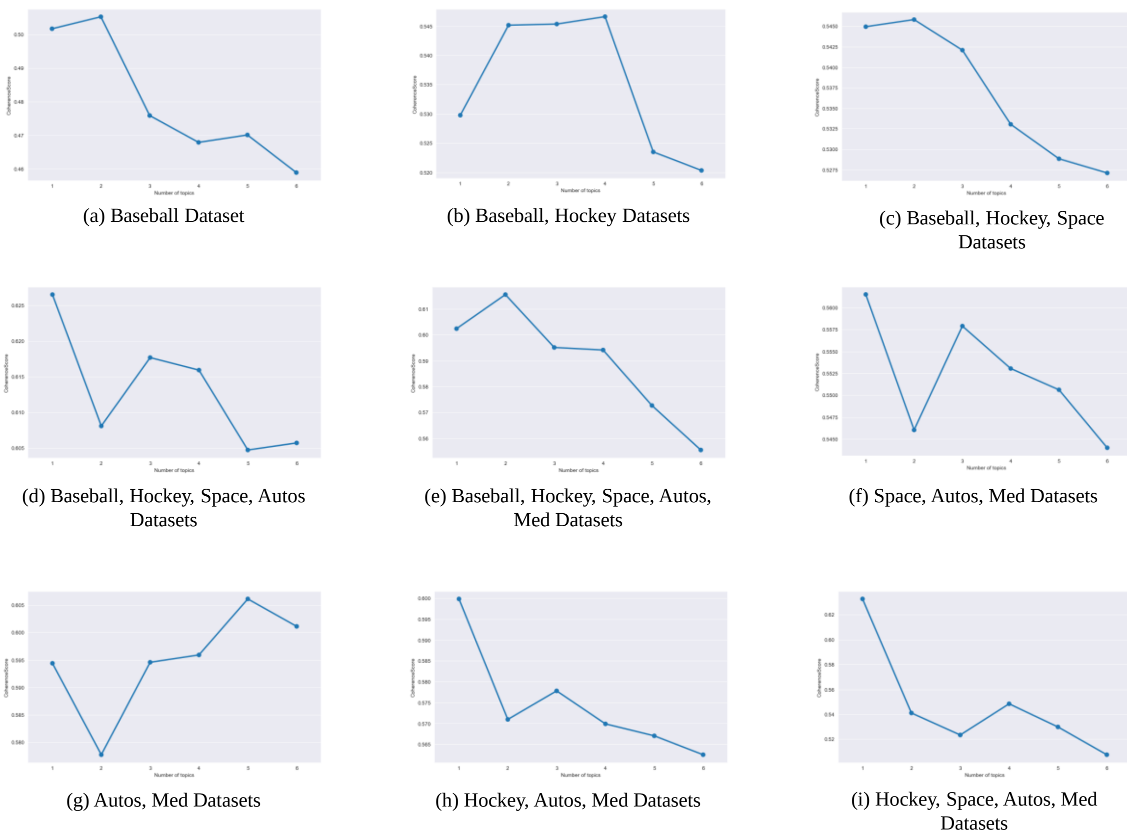
Table 3 displays the results of the eigenvalue heuristic. Additionally, Table 3 shows the number of a topics the user would have selected if utilizing the method of selecting the largest coherence score. Furthermore, Table 3 shows the results when the two methods are used with CUP (from Chapter 2). Approximately 66.7% of the 9 runs, the eigenvalue heuristic produced the correct number of topics verses the coherence score approach leading the user to select the incorrect number of topics for every run. When the eigenvalue heuristic is used with CUP, 77.8% of the 9 runs produced the number of correct number of topics verses 11.1% when using the coherence score approach with CUP.

Table 3. Eigenvalue Heuristic vs Coherence Score

Topic(s)	Number of Topics based on Eigenvalue Heuristic prior to CUP	Number of Topics based on Eigenvalue Heuristic after CUP	Number of Topics based on Coherence Score prior to CUP	Number of Topics based on Coherence Score after CUP
Baseball	1	1	2	1
Baseball, Hockey	3	2	4	1
Baseball, Hockey, Space	3	3	2	1
Baseball, Hockey, Space, Autos	3	3	1	3
Baseball, Hockey, Space, Autos, Med	4	4	2	4
Space, Autos, Med	3	3	1	5
Autos, Med	2	2	5	3

Hockey, Autos, Med	3	3	1	1
Hockey, Space, Autos, Med	4	4	1	5

Figure 11 and 12 show the coherence score plots for LDA prior to and after CUP, respectively. The location of the peak in each line was used to determine the number of topics the user would select when using the coherence score approach.



(a) Baseball Dataset

(b) Baseball, Hockey Datasets

(c) Baseball, Hockey, Space Datasets

(d) Baseball, Hockey, Space, Autos Datasets

(e) Baseball, Hockey, Space, Autos, Med Datasets

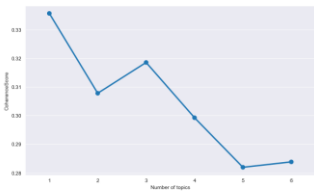
(f) Space, Autos, Med Datasets

(g) Autos, Med Datasets

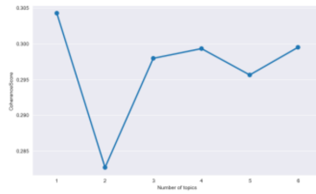
(h) Hockey, Autos, Med Datasets

(i) Hockey, Space, Autos, Med Datasets

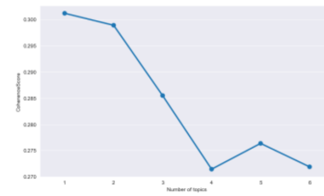
Figure 11. Coherence Score plots prior to CUP



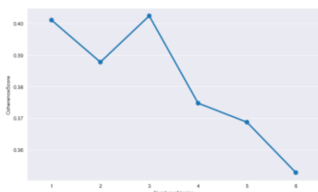
(a) Baseball Dataset



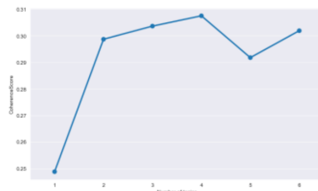
(b) Baseball, Hockey Datasets



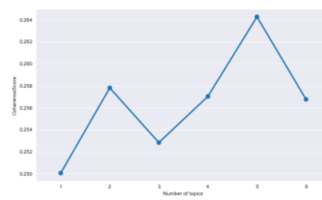
(c) Baseball, Hockey, Space Datasets



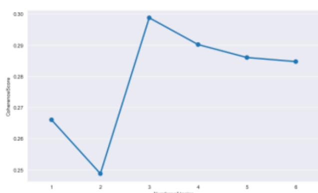
(d) Baseball, Hockey, Space, Autos Datasets



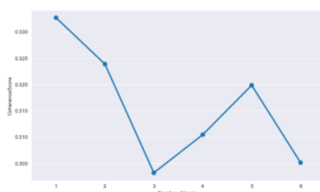
(e) Baseball, Hockey, Space, Autos, Med Datasets



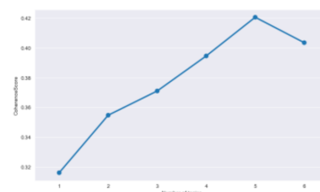
(f) Space, Autos, Med Datasets



(g) Autos, Med Datasets



(h) Hockey, Autos, Med Datasets



(i) Hockey, Space, Autos, Med Datasets

Figure 12. Coherence Score plots after CUP

Both methods, eigenvalue heuristic and coherence score approach, had improved results when paired with our CUP technique from Chapter 2. The eigenvalue heuristic provided a more reliable approach to determining *k* as an input into the LDA topic modeling technique. Since LDA is sensitive to a varying *k*, an effective and reliable approach is critical to increase model stability.

3.5 Conclusions

This chapter provides an eigenvalue heuristic for users to utilize when selecting the number of topics as an input to the LDA topic modeling technique. One of the challenges with determining the number of topics is validating the result is correct. Many factors, such as the writing style of the authors in the various text utilized in the model, will affect an algorithm's capability to produce an accurate result.

When using coherence scores to determine *k*, the user must know a general idea of how many topics the dataset may contain or have a domain expert nearby. The proposed eigenvalue heuristic does not require the user to have any insight into the dataset to have an initial *k* to feed into the LDA model. The eigenvalue heuristic provided a more direct and accurate approach to determining the number of topics when doing LDA.

The LDA topic modeling technique will vary the terms associated with each topic, as *k* varies. This feature is addressed in the next chapter when a new topic modeling technique is proposed.

IV. The Zimm Approach: A New Topic Modeling Technique

4.1 Introduction

Topic modeling allows us to gain insight into unstructured collections of textual data. There are different topic modeling techniques that have been developed. Each of the techniques requires the user to provide some sort of parameter input that can alter the output/analysis. Topic modeling is very popular; however, it is prone to noise sensitivity and instability which results in the results being unreliable (Vayansky & Kumar, 2020). Topic models can include where each document belongs to a single topic (Grimmer, 2010; Quinn et al, 2010) or where each document is a mixture of multiple topics (Blei, Ng & Jordan, 2003).

Topic modeling can help understand content among documents (Lesnikowski et al, 2019). It can provide users a way to see the differences between the publications over time. Topic modeling has been used in areas such as medical sciences (Zhang et al 2017), neuroscience (Koch et al, 2014), software engineering (Thomas et al, 2011), geography (Yin et al, 2011) and political science (Cohen & Ruths, 2013) fields. For example, topic modeling can be used to examine how politicians and policy-makers have adapted or changed their views on different situations.

4.2 Background

There are many topic modeling techniques to include Non-negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). This section provides an overview of those three topic modeling techniques.

There have been many derivatives of the these techniques however the basis still remains and users are required to input the number of topics and number of terms to output with each topic.

4.2.1 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is an unsupervised topic modeling technique (Vayansky & Kumar, 2020). NMF is matrix based and focuses on breaking down the document terms into low-rank factors that represent the bag of words (Shahbazi & Byun, 2020). NMF is capable of performing dimensionality reduction and clustering simultaneously (Albalawi, Yeap & Benyouce, 2020). NMF tries to identify two non-negative matrixes whose product is equal to the original matrix (Cai et al, 2008). Figure 13 shows an illustration of the NMF model for topic modeling. W is a n x d non-negative matrix and H is a d x t non-negative matrix (MacMillan & Wilson, 2017).

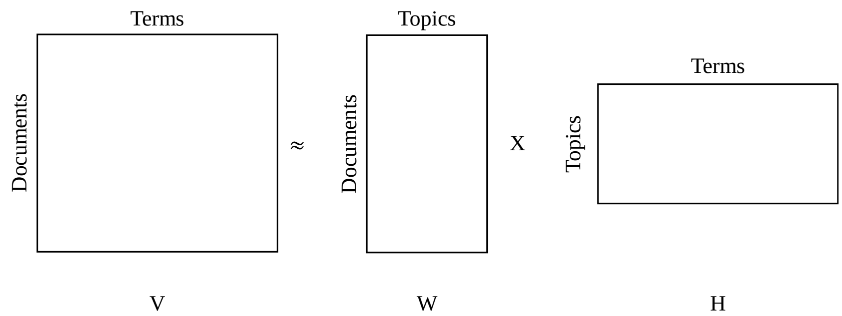


Figure 13. Illustration of NMF for Topic Modeling

NMF does not require prior knowledge to extract meaningful topics however sometimes it provides semantically incorrect results (Albalawi et al, 2020). NMF requires the user to enter the number of topics.

4.2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA), also known as Latent Semantic Index (LSI), can be used for topic modeling on unstructured data. Kulkarni, Apte and Evangelopoulos (2014) applied LSA in the Operations Management field to demonstrate the technique's ability to expose the intellectual structure of a discipline. LSA was selected due to the independence of preconceived notions with the intent to minimize the subject bias in the analysis. The goal of LSA is text representation vector creation to make semantic content (Alghamdi & Alfalqi, 2015). LSA uses singular value decomposition (SVD). SVD can reduce noise thus assisting in improved accuracy (Ozsoy, Alpaslan & Cicekli, 2011). LSA generally performs dimensionality reduction on the term frequency-inverse document frequency vectors. LSA requires the user to enter the number of topics and enter the number of words to output for each topic.

4.2.3 Latent Dirichlet Allocation

LDA is the simplest and most popular statistical topic modeling technique (George & Birla, 2018). The LDA model is a probabilistic model for the collections of discrete data (Blei, Ng & Jordan, 2003). LDA can be either supervised or unsupervised (Vayansky & Kumar, 2020). LDA uses the words in the document to identify the topic(s) that the document belongs to. The output of the LDA model is a list of topics and words with the associated probability that the word belongs to that topic. The basic LDA process can be viewed in Figure 14. The boxes are referred to as plates, the circles represent the variables or parameters and the arrows demonstrate the hierarchy of influence. The K box represents sampling for each topic, the N box represents sampling within each document and the M is the repeated sampling for each document (Vayansky & Kumar, 2020).

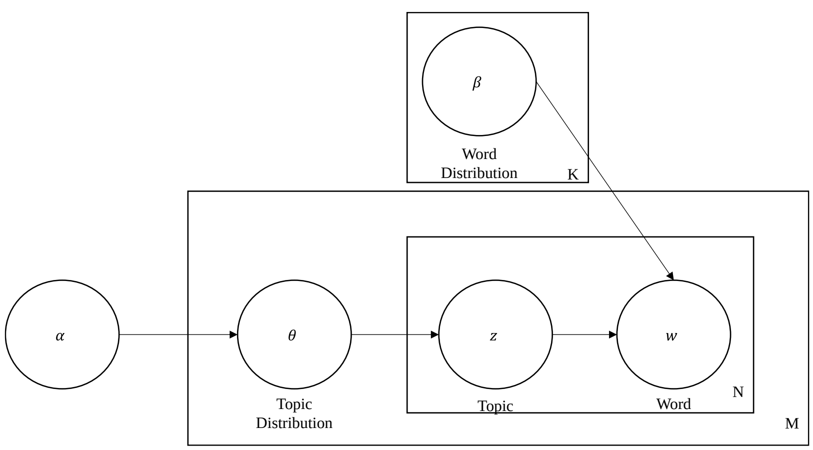


Figure 14. The Basic LDA Process

LDA does not require previous training data and can handle mixed length documents although for short messages, it needs an aggregation of the messages to avoid data sparsity (Albalawi et al, 2020). An example of a short message is a tweet. Tweets are messages on the social media platform Twitter that can be 140 characters long (Ito, Song, Toda, Koike & Oyama, 2015).

LDA utilizes the Dirichlet *priors* therefore it is less prone to overfitting and capable of inferring topics for unobserved documents (Yan et al, 2013). A weakness with using the Dirichlet prior lies within a simple assumption about the data generating process. It is assumed that every mixture model is equally likely, unless a higher-order structure is present (Gerlach, Peixoto, Altmann, 2018).

LDA is based on a nonhierarchical clustering of words (Gerlach et al, 2018). It does not take into consideration the order of the words or the sentence structure therefore the word ordering is unimportant thus creating Bag of Words (Misra, Cappé & Yvon, 2008). A key assumption of LDA is the bag of words will maintain the relevant information (Hoffman, 2001). It assumes all documents contain a mixture of topics (Feuerriegel & Pröllochs, 2021). Additionally, LDA assumes dimensionality of *k* (number of topics) of the Dirichlet distribution is known and fixed (Blei, Ng & Jordan, 2003). In order for *k* to be known, this requires prior knowledge about the contents of the dataset (Hasan et al, 2021).

Aside from the data, there are multiple user inputs into the LDA topic modeling technique: Alpha, Beta and number of topics (*k*) and number of terms per topic. Alpha is the parameter that set the prior on per document topic distribution. A high alpha implies every document is likely to contain a mixture of most topics where as a low alpha implies the document contains fewer topics. For a low alpha, the topic distribution samples are near the corners, near the topics implying the document only has one topic. This number is between not-zero and positive infinity. Beta sets the prior on the per topic word distribution. A high beta implies each topic is likely to consider most of the words and a low beta implies a topic may contain a mixture of just a few words (Binkley, Heinz, Lawrie & Overfelt, 2014). This number is between greater then 0, not inclusive, and positive infinity. The number of topics, *k*, is the number of topics the user wants the algorithm to extract from the corpus. The number of topic terms is the number of terms to be used in the composing of a topic, another user specified parameter. If a user wants to extract themes or concepts, select a high number of topic terms or extract features or terms use a low number of topic terms.

To minimize the amount of user required input, we developed a method that utilizes eigenvalues to determine number of topics and the loadings of the covariance matrix of the term document matrix to determine the number of terms and which terms for each topic.

The technique proposed in this paper does not require the user to input alpha, beta, number of topics, nor number of topic terms. This removes the requirement for prior knowledge of the dataset or access to someone who has knowledge of the dataset.

4.3 Methodology

Factor analysis (FA) is an unsupervised learning method for discovering latent variables. A latent variable is a variable that is inferred rather than directly observed. FA has been used as early as 1963 to extract topics and automatically classify documents

(Péladeau & Davoodi, 2018). Principal Component Analysis (PCA) and FA are similar dimensionality reduction techniques; however, there are some differences. PCA does not generate a model of underlying principal components similar to FA. While both PCA and FA take new dimensions as a hyperparameter, the model for FA should be built again while the change does not affect the principal components already computed in PCA. Therefore, the PCA concept is used in this topic modeling technique

4.3.1 Data and Preprocessing

Our research used the “auto” and “med” files from the 20newsgroup dataset. This led to a dataset size of 1088 text files and 19140 words after preprocessing. The preprocessing included the standard lower casing of letters, removal of punctuation, lemmatizing and stemming. For this dataset, email characters were also removed.

4.3.2 The Zimm Approach

A commonality throughout the literature is the utilization of the full bag of words as inputs to various modeling techniques (Xu, Heller, Ghahramani (2009)). Chapter 2 discussed the need for a stopwords list, beyond the standard preloaded package in Python, custom to a dataset. The identified heuristic was called the Coherent Utility Process (CUP). CUP is utilized in the Zimm Approach, new topic modeling technique, proposed in this chapter.

Chapter 3 proved an approach for determining the number of optimal topics based on eigenvalues greater than one to be an effective heuristic to determine the number of optimal topics. The heuristic was employed in this proposed algorithm. In Chapter 3, we looked at the covariance of the term-document matrix however this algorithm utilizes the covariance matrix of the mean centering data for the transpose of the term-document matrix.

Eigenvalues were computed and the number of topics assigned based on the number of eigenvalues greater than one. The associated eigenvectors were extracted and the loadings were calculated using formula 12.

$$loadings = eigenvector\_subset \ * \ \sqrt{eigenvalue\_subset} \tag{12}$$

where,

eigenvector_subset = the eigenvector associated with the corresponding eigenvalue greater than one

eigenvalue_subset = the eigenvalue, greater than one, that corresponds with the current eigenvector

The loadings for each topic were sorted and plotted. The maximum curvature in each plot was used to identify where the cut off for the terms to be associated with each topic was located. This allowed for the number of terms in each topic to vary. The number of terms for each topic will vary based on the loadings for each topic. The loadings were then mapped back to the term matrix to output terms for the number of topics specified.

4.4 Analysis and Results

A word cloud was initially created in order to implement CUP. Figure 15 displays the word cloud prior to CUP. Figure 16 displays the word cloud after CUP. After creating and implementing the custom stopwords list, the word cloud (in Figure 16) shows us that noise (which previously saturated main ideas of the data) was filtered out.

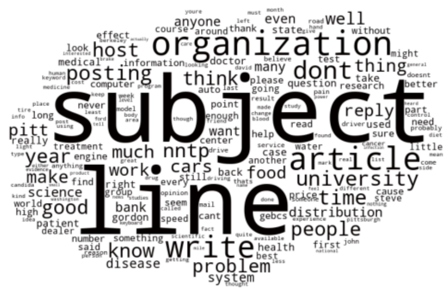


Figure 15. Word cloud of Dataset prior to CUP

Figure 16. Word cloud of Dataset after CUP

Based on the eigenvalues greater than one heuristic, the algorithm stated there were 37 topics. The algorithm was fed a dataset with two main topics, however, there may be subtopics. Additionally, the algorithm was modified to look at the maximum curvature of the scree plot of eigenvalues. This provided a value of 13. The output of the algorithm of k =37 and k=13 were both used for the Zimm Approach and for LDA. The varying of k demonstrated another benefit of this algorithm.

In LDA, when varying k the output varies. The terms in the grouping of each topic will change based on the user specified k. Additionally, with LDA the user must specify the number of terms to output with the topics. The number of terms with the topics will be the same.

For example, if the user selects 10, then there will be ten terms in the output for each topic. Figure 17 shows the LDA output when *k* is 13 and the number of terms is 10. Figure 18 shows the LDA output when *k* is 37 and the number of terms is 10 for each topic. In Figure 18, topic 1, “believe” and “doctor” are listed and not listed in Figure 17, topic 1. The terms will vary when *k* varies in the LDA topic modeling technique.

Topic: 0
words: ['pitt', 'gordon', 'banks', 'science', 'gebcs', 'computer', 'pittsburgh', 'univ', 'soon', 'njxp']
Topic: 1
words: ['health', 'years', 'medical', 'food', 'research', 'back', 'price', 'number', 'little', 'case']
Topic: 2
words: ['banks', 'gordon', 'pitt', 'pain', 'enough', 'right', 'work', 'back', 'cars', 'georgia']
Topic: 3
words: ['years', 'cars', 'water', 'please', 'first', 'back', 'right', 'engine', 'long', 'information']
Topic: 4
words: ['pitt', 'cars', 'gordon', 'right', 'gebcs', 'computer', 'banks', 'water', 'state', 'research']
Topic: 5
words: ['pitt', 'banks', 'gordon', 'cars', 'science', 'computer', 'gebcs', 'need', 'water', 'back']
Topic: 6
words: ['water', 'medical', 'information', 'first', 'health', 'thanks', 'research', 'work', 'washington', 'never']
Topic: 7
words: ['cars', 'science', 'food', 'engine', 'medical', 'back', 'might', 'patients', 'since', 'things']
Topic: 8
words: ['health', 'engine', 'science', 'disease', 'cars', 'without', 'convertible', 'since', 'driving', 'enough']
Topic: 9
words: ['food', 'work', 'years', 'since', 'health', 'pitt', 'never', 'first', 'information', 'science']
Topic: 10
words: ['cars', 'food', 'never', 'doctor', 'first', 'engine', 'without', 'around', 'getting', 'question']
Topic: 11
words: ['cars', 'years', 'science', 'first', 'thats', 'disease', 'since', 'right', 'thanks', 'treatment']
Topic: 12
words: ['cancer', 'right', 'state', 'medical', 'health', 'ohio', 'found', 'system', 'years', 'back']

Figure 17. LDA output with *k*=13

Topic: 0
words: ['pitt', 'gordon', 'banks', 'science', 'gebcs', 'computer', 'pittsburgh', 'soon', 'univ', 'njxp']
Topic: 1
words: ['medical', 'health', 'years', 'food', 'number', 'price', 'case', 'doctor', 'research', 'believe']
Topic: 2
words: ['gordon', 'banks', 'pain', 'weight', 'georgia', 'pitt', 'right', 'work', 'diet', 'need']
Topic: 3
words: ['water', 'polio', 'post', 'patients', 'please', 'systems', 'information', 'engine', 'years', 'cars']
Topic: 4
words: ['pitt', 'gordon', 'cars', 'banks', 'water', 'gebcs', 'computer', 'work', 'radar', 'state']
Topic: 5
words: ['pitt', 'gordon', 'banks', 'science', 'weight', 'case', 'gebcs', 'computer', 'uucp', 'right']
Topic: 6
words: ['water', 'medical', 'first', 'radar', 'science', 'information', 'odometer', 'health', 'group', 'never']
Topic: 7
words: ['science', 'scientific', 'medical', 'might', 'health', 'made', 'since', 'cars', 'patients', 'back']
Topic: 8
words: ['disease', 'skin', 'without', 'health', 'science', 'driving', 'problems', 'oily', 'patients', 'enough']
Topic: 9
words: ['pitt', 'work', 'years', 'science', 'medicine', 'first', 'health', 'banks', 'medical', 'information']
Topic: 10
words: ['cars', 'first', 'medical', 'around', 'food', 'never', 'getting', 'insurance', 'high', 'question']
Topic: 11
words: ['cars', 'years', 'science', 'first', 'yeast', 'thats', 'area', 'right', 'read', 'since']
Topic: 12
words: ['cancer', 'ringing', 'state', 'great', 'health', 'first', 'shift', 'back', 'medical', 'weight']
Topic: 13
words: ['cars', 'integra', 'candida', 'food', 'tires', 'name', 'drive', 'rocks', 'great', 'read']
Topic: 14
words: ['things', 'cars', 'food', 'spot', 'every', 'treatment', 'please', 'question', 'june', 'taste']
Topic: 15
words: ['cars', 'engine', 'state', 'pitt', 'banks', 'gordon', 'question', 'ohio', 'speed', 'years']
Topic: 16
words: ['water', 'mwra', 'dept', 'years', 'food', 'health', 'medical', 'cancer', 'research', 'chinese']
Topic: 17

words: ['pitt', 'banks', 'gordon', 'gebcs', 'science', 'pittsburgh', 'intellect', 'soon', 'skepticism', 'univ']
Topic: 18
words: ['insurance', 'cars', 'cancer', 'medical', 'taurus', 'enough', 'years', 'health', 'looking', 'costs']
Topic: 19
words: ['science', 'shots', 'work', 'send', 'state', 'dyer', 'research', 'steve', 'cars', 'nasa']
Topic: 20
words: ['group', 'food', 'migraine', 'little', 'work', 'thats', 'back', 'corn', 'experience', 'james']
Topic: 21
words: ['years', 'dealer', 'back', 'right', 'world', 'information', 'thanks', 'cars', 'please', 'list']
Topic: 22
words: ['engine', 'steve', 'doctor', 'dyer', 'ultrasound', 'food', 'read', 'back', 'using', 'another']
Topic: 23
words: ['gordon', 'cars', 'pitt', 'never', 'banks', 'help', 'food', 'please', 'science', 'engine']
Topic: 24
words: ['please', 'right', 'food', 'disease', 'system', 'crohns', 'diet', 'foods', 'cars', 'patients']
Topic: 25
words: ['point', 'help', 'medical', 'effect', 'engine', 'disease', 'cars', 'medicine', 'harvard', 'thats']
Topic: 26
words: ['toyota', 'dealer', 'pain', 'back', 'study', 'thanks', 'stanford', 'reading', 'john', 'another']
Topic: 27
words: ['system', 'right', 'needles', 'back', 'world', 'john', 'craig', 'pitt', 'aids', 'state']
Topic: 28
words: ['pitt', 'gordon', 'science', 'banks', 'gebcs', 'pittsburgh', 'computer', 'read', 'john', 'please']
Topic: 29
words: ['pitt', 'years', 'information', 'health', 'research', 'pittsburgh', 'need', 'never', 'washington', 'cancer']
Topic: 30
words: ['saturn', 'harvard', 'honda', 'dyer', 'dealer', 'cars', 'price', 'food', 'road', 'profit']
Topic: 31
words: ['food', 'work', 'state', 'uoknor', 'james', 'research', 'years', 'back', 'cars', 'science']
Topic: 32
words: ['right', 'cars', 'food', 'problems', 'someone', 'drivers', 'science', 'high', 'speed', 'without']
Topic: 33
words: ['years', 'pain', 'insurance', 'back', 'help', 'cars', 'might', 'real', 'first', 'driving']
Topic: 34
words: ['pain', 'back', 'help', 'disease', 'health', 'problems', 'crohns', 'medical', 'information', 'body']
Topic: 35
words: ['list', 'back', 'engine', 'cars', 'science', 'lights', 'computer', 'email', 'mail', 'autos']

Topic: 36
words: ['cars', 'thanks', 'drive', 'side', 'volvo', 'price', 'corn', 'road', 'right', 'mail']

Figure 18. LDA output when *k* = 37

In the Zimm Approach, whether selecting 37 or 13, the first thirteen groups of terms are the same. When varying *k* the words associated with each topic did not change. Therefore, if an individual decided to manually select *k* the output within the topics would not change. Furthermore, the number of terms selected for each output is not consistent and does not require user input, as discussed below.

After extracting each corresponding eigenvector and eigenvalue, the corresponding loading was calculated based on formula (12). The loading values were plotted and the maximum curvature point of each plot was used to determine the number of terms for each topic. Then the vector values were mapped back to the term matrix to produce an output of *k* topics that contains the number of terms determined by the corresponding plot. This method allowed for a varying number of terms per topic since some terms may contribute more to the calculations than others.

Table 4 shows a sample of the output for the Zimm Approach when *k*=13. Table 5 shows a sample of the output for the Zimm Approach when *k*=37. The number of terms per topic varies based on the heuristic of the algorithm however the terms are consistent.

Table 4. Zimm Approach with *k*=13

	Topic1	Topic 2	Topic 3	...	Topic 12	Topic 13
0	cars	cancer	tobacco		requests	pitt
1	pitt	center	water		send	gordon
2	science	research	smokeless		keyboard	banks
3	banks	aids	health		cars	gebcs
4	back	medical	coli		price	requests
5	right	centers	dept		autos	send
6	gordon	comprehensive	case		list	science
7	work	clinical	food		supports	pittsburgh
8	engine	avenue	mwra		shipping	skepticism
9	read	internet	candida		sequence	chastity
10	computer	study	infections		protein	njxp
11	help	melanoma	disease		lists	intellect
12	gebcs	york	outbreak		biology	gebcadre
13	going	vaccines	pitt		contact	shameful
14	thanks	trials	chain		molecular	surrender
15	things	street	gordon		system	soon
16	speed	particles	science		national	univ
17	question	information	aids		phone	computer
18	best	asthma	patients		keys	candida
19	price	particulate	illness		standard	fluids
20	different	researchers	banks		mustangs	weight
21	probably	infected	infection		mailing	brake
22	pain	basic	snuff		candida	exercises
23	never	vaccine	prevalence		computer	lyme
24	enough	hicnet	study		genetic	braking
25	believe	treatment	diarrhea		dragon	program
26	little	april	bloody		mouse	tool
27	doctor	page	yeast		keyboards	typing
28	water	care	restaurant		requestballtown	lists
29	left	test	persons		biological	help
30	thats	education	steve		normal	uucp
31	steve	institute	first		automotive	patients
32	anything	medicine	years		systems	japanese
33	dealer	volume	smoking		chris	windows
34	point	found	users		saturn	medical
35	someone	trial	vitamin		conference	breaks
36	quite	north	evidence		artificial	management
37	without	early	identified		radar	system

38	mail	california	former		knowledge	available
39	might	designated	cause		buttons	manufacturers
40	every	patients	onset		international	tires
41	youre	immune	chewing		intelligence	software
42	though	development	women		addresses	tools
43	around	administration	gebcs		separate	pedal
44	find	made	least		washington	designation
45	driving	newsletter	january		discussion	description
46	getting	institutions	city		analysis	physicians
47	drive	within	meat		school	disease
48	problems	american	found		david	threshold
49	long	schwartz	symptoms		carroll	type
50	doesnt	matter	eating		race	additives
51	great	mice	question		learning	physician
52	another	multiple	patties		prediction	probably
53	opinions	scientists	hamburgers		braille	platforms
54	come	shalala	anti		structure	warns
55	looking	consensus	matched		large	training
56	keep	findings	medical		balltown	body
57	done	lung	public		data	boiling
58	berkeley	msdos	editor		compatible	migraine
59	course	consortium	school		july	bloom
60	keyboard	utah	infected		utah	tire
61	ford	positive	escherichia		intended	number
62	look	rochester	medicine		discussions	brakes
63	tires	last	diet		topics	mustangs
64	actually	east	immune		registration	silicone
65	seems	criteria	services		exotic	questions
66	power	institutes	washington		weltycabot	courses
67	nothing	skin	smoked		welty	fuel
68	diet	seattle	care			fluid
69	candida	ohio	john			version
70	keywords	effects	bloom			often
71	weight	professionals	stool			requestballtown
72	heard	levels	current			yeast
73	autos	programs	sinus			belt
74	front	site	skin			includes
75	maybe	drug				sound
76	else	says				break
77	side	miami				oils

78	post	microgenesys				intervals
79	hard	road				portland
80	check	strong				valve
81	fast	airborne				omen
82	mark	pennsylvania				provide
83	john	published				various
84	brake	emergency				effective
85	tell	cost				sinus
86		experts				language
87		evidence				quack
88		angeles				cases
89		south				calendar
90		physicians				cycle
91		virus				drug
92		ozone				viscosity
93		transgenic				useful
94		clearinghouse				ones
95		tested				richard
96		columbia				gasolines
97		engage				patient
98		vermont				listserv
99		michigan				addresses
100		exposure				rotors
101		virginia				slick
102		project				damage
103		boulevard				technology
104		pollution				gasoline
105		association				equipped
106		respiratory				rebound
107		albert				medicine
108		bitnet				general
109		reports				blood
110		developing				timing
111		sources				programs
112		texas				
113		room				
114		carolina				
115		science				
116		children				
117		tucson				

118		mortality				
119		establishment				
120		experimental				
121		herpesvirus				
122		scientific				
123		secretary				
124		attack				
125		cells				
126		however				
127		genes				
128		arizona				
129		domain				
130		panel				
131		support				

Table 5. Zimm Approach with *k*=37

	Topic 1	Topic 2	Topic 3	….	Topic36	Topic37
0	cars	cancer	tobacco		list	polio
1	pitt	center	water		school	list
2	science	research	smokeless		request	school
3	banks	aids	health		file	carcinogenic
4	back	medical	coli		mailing	smoke
5	right	centers	dept		favorite	patients
6	gordon	comprehensive	case		food	request
7	work	clinical	food		script	meat
8	engine	avenue	mwra		name	motor
9	read	internet	candida		email	post
10	computer	study	infections		mail	mailing
11	help	melanoma	disease		address	mail
12	gebcs	york	outbreak		lists	read
13	going	vaccines	pitt		addresses	wood
14	thanks	trials	chain		owner	file
15	things	street	gordon		photography	tray
16	speed	particles	science		sender	smoked
17	question	information	aids		network	name
18	best	asthma	patients		home	evidence
19	price	particulate	illness		several	script
20	different	researchers	banks		listserv	stuff
21	probably	infected	infection		welty	syndrome

22	pain	basic	snuff		corn	favorite
23	never	vaccine	prevalence		kirlian	risk
24	enough	hicnet	study		probably	lists
25	believe	treatment	diarrhea		pain	grey
26	little	april	bloody		member	charcoal
27	doctor	page	yeast		balltown	chips
28	water	care	restaurant		shell	unpleasant
29	left	test	persons		bounced	heard
30	thats	education	steve		nasa	
31	steve	institute	first		object	
32	anything	medicine	years		echo	
33	dealer	volume	smoking		alias	
34	point	found	users		thanks	
35	someone	trial	vitamin		points	
36	quite	north	evidence		road	
37	without	early	identified		need	
38	mail	california	former		sysadmin	
39	might	designated	cause		energy	
40	every	patients	onset		case	
41	youre	immune	chewing		requestballtown	
42	though	development	women		state	
43	around	administration	gebcs		krillean	
44	find	made	least		members	
45	driving	newsletter	january		around	
46	getting	institutions	city		misc	
47	drive	within	meat		seizures	
48	problems	american	found		possible	
49	long	schwartz	symptoms		systems	
50	doesnt	matter	eating		might	
51	great	mice	question		kids	
52	another	multiple	patties		message	
53	opinions	scientists	hamburgers		errors	
54	come	shalala	anti			
55	looking	consensus	matched			
56	keep	findings	medical			
57	done	lung	public			
58	berkeley	msdos	editor			
59	course	consortium	school			
60	keyboard	utah	infected			
61	ford	positive	escherichia			

62	look	rochester	medicine			
63	tires	last	diet			
64	actually	east	immune			
65	seems	criteria	services			
66	power	institutes	washington			
67	nothing	skin	smoked			
68	diet	seattle	care			
69	candida	ohio	john			
70	keywords	effects	bloom			
71	weight	professionals	stool			
72	heard	levels	current			
73	autos	programs	sinus			
74	front	site	skin			
75	maybe	drug				
76	else	says				
77	side	miami				
78	post	microgenesys				
79	hard	road				
80	check	strong				
81	fast	airborne				
82	mark	pennsylvania				
83	john	published				
84	brake	emergency				
85	tell	cost				
86		experts				
87		evidence				
88		angeles				
89		south				
90		physicians				
91		virus				
92		ozone				
93		transgenic				
94		clearinghouse				
95		tested				
96		columbia				
97		engage				
98		vermont				
99		michigan				
100		exposure				
101		virginia				

102		project				
103		boulevard				
104		pollution				
105		association				
106		respiratory				
107		albert				
108		bitnet				
109		reports				
110		developing				
111		sources				
112		texas				
113		room				
114		carolina				
115		science				
116		children				
117		tucson				
118		mortality				
119		establishment				
120		experimental				
121		herpesvirus				
122		scientific				
123		secretary				
124		attack				
125		cells				
126		however				
127		genes				
128		arizona				
129		domain				
130		panel				
131		support				

Table 4 and Table 5 shows the stability, the core terms do not vary when *k* changes, this approach provides the user in the output. This stability is important when adding additional documents to the corpus. This approach will provide the user a way to compare the impact of the new documents. Appendix C provides the full Zimm Approach algorithm and the LDA algorithm used in this analysis.

4.5 Conclusions

The digital age means textual data is growing at an explosive rate. The human is not capable of keeping up with the content of information available without assistance from machines. There exist many different topic modeling techniques and variations of those techniques.

The existing techniques requires the user to input parameters that has a direct impact on the output of the algorithm. This proposed topic modeling technique does not vary the terms associated with the topic, even if the user varies *k*. The number of terms the algorithm outputs with each term differs from term to term pending on the plot of the loadings. The topic modeling technique proposed in this article removes the requirement for those parameter inputs while providing a more stable output.

V. Conclusions and Recommendations

This research started with exploring various topic modeling techniques and identifying potential areas for improvements. As with any model, the quality of the output is highly dependent on the quality of the input. Throughout the readings a commonality of a use of a standard stopwords package, the use of full bag of words (BoW) is used in the topic modeling techniques and the requirement for the user to input the number of topics, *k*, for the model to populate, exist.

5.1 Conclusions

Chapter 2 identifies the need to have a customized stopwords list for a dataset. The word cloud is used as visualization tool to assist the user in creating the custom stopwords list, the process was called Coherent Utility Process (CUP). This process can be an irritative process to reduce as much noise as possible. Additionally, a technique for identifying a term frequency inverse document frequency (TF-IDF) range, narrowing the BoW used as an input into the Latent Dirichlet Allocation (LDA) topic modeling technique. This technique was called Prominent Extraction Technique (PET). PET is based on the total words used in the document. The CUP and PET approaches allowed the LDA topic modeling technique to achieve a level of utility not previously attainable.

Chapter 3 explores a variety of current methods used to help users determine the number of topics, *k*, for the topic modeling technique to populate. The requirement for the user to select a value for *k*, assumes the user has prior knowledge of the dataset. There are two challenges that exist with the current heuristics that were addressed with our heuristic: 1) In graphical methods, which value should the user select if more than one peak exists? and 2) Users are expected to input different values of *k* to determine optimal scores, what range should the user select to test?. LDA was selected as the topic modeling to use when testing our heuristic. Varying of *k* can cause the output to vary therefore it is important to provide a reliable method for the user to select *k*. Our developed heuristic based on the number of eigenvalues greater than one, using the term document matrix, provided more reliable results when compared to the popular graphing of coherence scores technique.

Finally, Chapter 4 proposes a new topic modeling technique called the Zimm Approach. LDA is a popular topic modeling technique however it requires the user to input the number of topics and the number of terms to output for the topics. In LDA, the number of terms per topic is the same. The Zimm Approach includes CUP, from Chapter 2, and the eigenvalue heuristic, from Chapter 3, while developing a new topic modeling technique. The Zimm Approach does not require the user to select a value for *k* and does not require the use to determine the number of terms for each topic. The new technique allows for a varying number of terms in each topic. Furthermore, an advantage of the Zimm Approach is the stability of the algorithm. If you vary *k*, the terms do not change. For example, if *k*=13 and then the user made *k*=37, the first 13 terms of each topic for all *k*'s will be the same. Whereas, when you vary *k* in LDA the terms the technique outputs will vary.

5.2 Recommendations for Future Research

Topic modeling will continue to be an area of interest and there are many areas for improvement. The techniques in this dissertation used unigrams (single word). Further research could look at bigrams (two words) to expand the concepts.

Additionally, this research focused heavily on the LDA modeling technique. The techniques discussed could be applied among other topic modeling techniques such as Latent Semantic Analysis and Non-Negative Matrix Factorization. If an individual was more focused on LDA, then an algorithm to assist the LDA model in determining the number of terms for each topic would allow more flexibility in the algorithm.

The Zimm Approach outputs the topics and a list of terms for each topic. Future research would include creating a way for the user to visualize the output, other than a list. While the CUP technique retains the human in the data processing loop, requiring decisions to be made about the importance/usefulness of a word, future research should be conducted to create an algorithm to identify the words to enhance the stopwords list, without the need for human entry

Finally, the ultimate metric for evaluating topic modeling outputs is the usability to the user. Coherence Scores fluctuate and do not always align with human interpretability. Further research would develop and/or refine metrics for topic modeling.

Appendix A: Python Code for CUP and PET

```
#Load Packages

import nltk
import numpy as np
import pandas as pd
import re, gensim
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer  #oldest method developed 1979
from nltk.stem import WordNetLemmatizer
from gensim.models.coherencemodel import CoherenceModel
import gensim.corpora as corpora
from tqdm._tqdm_notebook import tqdm

# Plotting tools
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import seaborn as sns

#Import Data
df = pd.read_json('https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json')
#print(df.target_names.unique())

#Filters out rec.sport.hockey files
df = df[df["target_names"].str.contains("rec.sport.baseball")]

#Preprocessing
# Convert to list
data = df.content.values.tolist()

#Remove extra spaces
for i in range(len(data)):
    data[i]=" ".join(data[i].split())


# Remove Emails
data = [re.sub('\b*@\b*\b?', '', sent) for sent in data]

# Remove new line characters
data = [re.sub('\b', ' ', sent) for sent in data]

# Remove distracting single quotes
data = [re.sub("\'", "", sent) for sent in data]

#Remove punctuation
from string import punctuation   #contains !"#$%&'()+,-./:;?@{}[]_^`~
data = [re.sub('['+punctuation+']',' ', sent) for sent in data]


exclude ='\\'
for i in range(len(data)):
```

```
    data[i] = ''.join(sent for sent in data [i] if sent not in exclude)

#Make Lower case
for i in range(len(data)):
    data [i] = data [i].lower()  #Converts to lower case


#Lemmatize
#Data Cleansing
for i in range(len(data)):
    data [i] = ''.join([WordNetLemmatizer().lemmatize(word) for word in data[i]])#Lemmatize

#Stem
#Data Cleansing
for i in range(len(data)):
    data [i] = ''.join([PorterStemmer().stem(word) for word in data[i]]) #Stem

#Remove Numbers
for i in range(len(data)):
    data [i] = ''.join([word for word in data[i] if not word.isdigit()])


#Remove single characters
for i in range(len(data)):
    data [i] = re.sub(r'\b[a-zA-Z]\b',' ',data [i])   # Removes single characters


#Remove words with length of 3 or less
for i in range(len (data)):
    data[i] = re.sub(r'\b\w{1,3}\b','', data[i])

#Remove stopwords
#Stopword list creation
stop_words = stopwords.words("english")
custom_stop_words =['from', 're', 'subject', 'would',
                    'organization', 'university', 'year', 'line',
                    'better', 'well', 'still', 'like', 'nntp', 'think',
                    'dont', 'good', 'writes', 'might', 'know', 'much', 'give',
                    'article', 'even', 'last', 'anyone', 'make', 'time', 'look', 'play',
                    'season', 'come', 'said', 'great', 'didnt', 'back', 'maybe', 'going',
                    'rally', 'reply', 'though', 'many', 'years', 'thats', 'best', 'lines',
                    'game', 'team', 'player']

stop_words = custom_stop_words + stop_words

for i in range(len(data)):
    data [i] = ' '.join([word for word in data[i].split(' ') if word not in stop_words]) #Removes
stopwords
```

```
#Remove extra spaces
for i in range(len(data)):
    data[i]=" ".join(data[i].split())

#Create WordCloud

#change value to black
def black_color_func(word, font_size, position, orientation, random_state=None, **kwargs):
    return ("hsl(0,100%,1%)")

#convert list to string and generate
unique_string=(" ").join(data)
from PIL import Image
background_image=np.array(Image.open('C://Users/jzim2/Desktop/Dissertation/test.jpg'))
wordcloud = WordCloud(prefer_horizontal = 1.0, background_color="white",
mask=background_image, width = 1000, height = 500, collocations =
False).generate(unique_string)

wordcloud.recolor(color_func=black_color_func)
plt.figure(figsize=(15,8))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()


#Tokenize (removing punctuations,
# each sentence into list of words)  Create Dictionary
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes
punctuations

data_words = list(sent_to_words(data))

id2word = corpora.Dictionary(data_words)

#Creating BOW model
wordfreq = {}
for sentence in data:
    tokens = nltk.word_tokenize(sentence)
    for token in tokens:
        if token not in wordfreq.keys():
            wordfreq[token] = 1
        else:
            wordfreq[token] += 1


#Term Frequency (Term Frequency)
```

```
#number of times a word appears in a document
#Calculate TF
BOWCount= len(wordfreq)
tfvalue = {}
for word, count in wordfreq.items():
    tfvalue[word] = count/float(BOWCount)


#Calculate IDF
# measure of how significant that term is in the whole corpus (collection of documents)
#words that appear too often in a document will have lower weights and words that don't appear
too often will have bigger weights

word_idf_values = {}
for token in wordfreq.keys():
    doc_containing_word = 0
    for document in data:
        if token in nltk.word_tokenize(document):
            doc_containing_word += 1
    word_idf_values[token] = np.log(len(data)/(doc_containing_word))


#Extract dictionary values
dict_value = []
for key in word_idf_values.keys() :
    dict_value.append(word_idf_values[key])

#Sort dictionary values
dict_value.sort(reverse=True)


#TF-IDF
#low (near zero) words that occur in many documents in a collecton
#high for words that occur in fewer documents

dict1=tfvalue
dict2=word_idf_values
dict_TFIDF = {k : v * dict2[k] for k, v in dict1.items() if k in dict2}


#Round dict_TFIDF values
# initializing t    4 decimal places
t = 4

# loop to iterate for values
dict_TFIDF_rounded = dict()
for key in dict_TFIDF:

    # rounding to K using round()
```

```
    dict_TFIDF_rounded[key] = round(dict_TFIDF[key], t)


#Export TFIDF values
df = pd.DataFrame(data=dict_TFIDF, index=[0])
df = (df.T)
#print (df)
df.to_excel(r"C:\Users\jzim2\Desktop\Dissertation\Paper1\dict_TFIDF.xlsx")


#Export Word Count
df = pd.DataFrame(data=wordfreq, index=[0])
df = (df.T)
#print (df)
df.to_excel(r"C:\Users\jzim2\Desktop\Dissertation\Paper1\dict1.xlsx")


#Create Dictionary
def sent_to_words(sentences):
    for sentence in sentences:
        # deacc=True removes punctuations
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))

data = list(sent_to_words(data))
id2word = corpora.Dictionary(data)
corpus = [id2word.doc2bow(word) for word in data]


coherenceList_cv = []
num_topics_list = np.arange(1,6)
for num_topics in tqdm(num_topics_list):
    lda_model= gensim.models.LdaModel(alpha= 'auto', eta="auto", corpus=corpus,
id2word=id2word,
                                num_topics=num_topics, random_state=42)
    cv = CoherenceModel(model=lda_model, corpus=corpus,
                        texts=data, dictionary=id2word, coherence='c_v')
    coherenceList_cv.append(cv.get_coherence())
for index, topic in lda_model.show_topics(formatted=False, num_words=10):
    print('Topic: {} \nwords: {}'.format(index, [w[0] for w in topic]))

print(coherenceList_cv)

plotcvData= pd.DataFrame({'Number of topics':num_topics_list,
                        'Full BoW':coherenceList_cv})

f,ax = plt.subplots(figsize=(10,6))
sns.set_style("darkgrid")
plot = sns.pointplot(x='Number of topics',y= 'Full BoW',data=plotcvData)
plot.set_ylabel("Coherence Score")
```

```
#plt.axhline(y=-3.9)
#plt.title('Topic coherence')
plt.show()

#Narrow BoW based on Word Count leading to TFIDF range narrowing
#Calculations completed from Files Exported to Excel
narrowed_BoW = {key : val for key, val in dict_TFIDF.items()
                if val>0.025572513 and val<=0.046691189}

#Extract just word from narrowed BoW
narrowed_BoWterms = list()
for i in narrowed_BoW.keys():
    narrowed_BoWterms.append(i)

#Create Dataset based on narrow words

narrowed_data =[]
narrowed_data = data
for i in range(len(narrowed_data)):
    narrowed_data [i] = ' '.join([word for word in narrowed_data[i] if word in
narrowed_BoWterms])

#Create WordCloud
import matplotlib.pyplot as plt
from wordcloud import WordCloud

#change value to black
def black_color_func(word, font_size, position, orientation, random_state=None, **kwargs):
    return ("hsl(0,100%,1%)")

#convert list to string and generate
unique_string=(" ").join(narrowed_data)
from PIL import Image
background_image=np.array(Image.open('C://Users/jzim2/Desktop/Dissertation/test.jpg'))
wordcloud = WordCloud(prefer_horizontal = 1.0, background_color="white",
mask=background_image, width = 1000, height = 500, collocations =
False).generate(unique_string)

wordcloud.recolor(color_func=black_color_func)
plt.figure(figsize=(15,8))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#Create Dictionary
def sent_to_words(sentences):
    for sentence in sentences:
        # deacc=True removes punctuations
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))
```

```
narrowed_data_list = list(sent_to_words(narrowed_data))
id2word = corpora.Dictionary(narrowed_data_list)
corpus = [id2word.doc2bow(word) for word in narrowed_data_list]

coherenceList_cv_narrowed = []
num_topics_list = np.arange(1,6)
for num_topics in tqdm(num_topics_list):
   lda_model= gensim.models.LdaModel(alpha= 'auto', eta="auto", corpus=corpus,
id2word=id2word,
                                      num_topics=num_topics, random_state=42)
   cv = CoherenceModel(model=lda_model, corpus=corpus,
                        texts=narrowed_data_list, dictionary=id2word, coherence='c_v')
   coherenceList_cv_narrowed.append(cv.get_coherence())
for index, topic in lda_model.show_topics(formatted=False, num_words=10):
   print('Topic: {} \nwords: {}'.format(index, [w[0] for w in topic]))

plotcvData_narrowed = pd.DataFrame({'Number of topics':num_topics_list,
                         'Narrowed BoW':coherenceList_cv_narrowed})

f,ax = plt.subplots(figsize=(10,6))
sns.set_style("darkgrid")
sns.pointplot(x='Number of topics',y= 'Narrowed BoW',data=plotcvData_narrowed)
plt.show()
print(coherenceList_cv_narrowed)

plotcvData_combined = pd.DataFrame({'Number of Topics':num_topics_list,
                         'Narrowed BoW':coherenceList_cv_narrowed,
                         'Full BoW':coherenceList_cv})

Narrowed= sns.pointplot(x='Number of Topics', y= 'Narrowed BoW', data =
plotcvData_combined, linestyles = '--', markers= '^', linewidth = 2.0)
Full = sns.pointplot(x='Number of Topics', y= 'Full BoW', data = plotcvData_combined)
Full.set_ylabel("Coherence Score")
plt.legend(labels = ["Narrowed BoW", "Full BoW"])
plt.show()
```

Appendix B: Python Code for Eigenvalue Heuristic to Determine *k*

```
#Load Packages
import numpy as np
import pandas as pd
import re, gensim
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer  #oldest method developed 1979
from nltk.stem import WordNetLemmatizer
import gensim.corpora as corpora
from gensim.models.coherencemodel import CoherenceModel
from tqdm._tqdm_notebook import tqdm
from sklearn.feature_extraction.text import CountVectorizer
import matplotlib.pyplot as plt
import seaborn as sns

#Import Data
df = pd.read_json('https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json')
#print(df.target_names.unique())

#Assign different files to variables
baseball = df[df["target_names"].str.contains("rec.sport.baseball")]
hockey = df[df["target_names"].str.contains("rec.sport.hockey")]
space = df[df["target_names"].str.contains("sci.space")]
autos = df[df["target_names"].str.contains("rec.autos")]
med = df[df["target_names"].str.contains("sci.med")]

df = [hockey, space, autos, med]
df = pd.concat(df)

#Preprocessing
# Convert to list
data = df.content.values.tolist()

#Remove extra spaces
for i in range(len(data)):
    data[i]=" ".join(data[i].split())

# Remove Emails
data = [re.sub('\b*@\b*\b?', '', sent) for sent in data]

# Remove new line characters
data = [re.sub('\b', ' ', sent) for sent in data]

# Remove distracting single quotes
data = [re.sub("\'", "", sent) for sent in data]

#Remove punctuation
from string import punctuation   #contains !"#$%&'()+,-./:;?@{}[]_^`~
data = [re.sub('['+punctuation+']',' ', sent) for sent in data]
```

```
exclude ='\\'
for i in range(len(data)):
   data[i] = ''.join(sent for sent in data [i] if sent not in exclude)

#Make Lower case
for i in range(len(data)):
   data [i] = data [i].lower()  #Converts to lower case

#Lemmatize
#Data Cleansing
for i in range(len(data)):
   data [i] = ''.join([WordNetLemmatizer().lemmatize(word) for word in data[i]])#Lemmatize

#Stem
#Data Cleansing
for i in range(len(data)):
   data [i] = ''.join([PorterStemmer().stem(word) for word in data[i]]) #Stem

#Remove Numbers
for i in range(len(data)):
   data [i] = ''.join([word for word in data[i] if not word.isdigit()])


#Remove single characters
for i in range(len(data)):
   data [i] = re.sub(r'\b[a-zA-Z]\b',' ',data [i])   # Removes single characters


#Remove words with length of 3 or less
for i in range(len (data)):
   data[i] = re.sub(r'\b\w{1,3}\b','', data[i])

#Remove stopwords
#Stopword list creation
stop_words = stopwords.words("english")
custom_stop_words =['from','re', 'subject', 'would',
'organization','university','year','line','better','well','still', 'like',
                'nntp',
'think','dont','good','writes','might','know','much','give','article','even','last','anyone','make',

'time','look','play','season','come','said','great','didnt','back','maybe','going','really','reply','though',
                'many','years','thats','best','lines','game','team','player']
stop_words = custom_stop_words + stop_words

for i in range(len(data)):
   data [i] = ' '.join([word for word in data[i].split(' ') if word not in stop_words]) #Removes
stopwords

#Remove extra spaces
```

```
for i in range(len(data)):
    data[i]=" ".join(data[i].split())

#Tokenize (removing punctuations,
# each sentence into list of words)  Create Dictionary
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes
punctuations

data_words = list(sent_to_words(data))
id2word = corpora.Dictionary(data_words)

# Count Vectorizer
vect = CountVectorizer()
vects = vect.fit_transform(data)

# Select the rows from the data set

td= pd.DataFrame(vects.todense()).iloc[:len(data)]
td.columns = vect.get_feature_names()
term_document_matrix = td.T
term_document_matrix.columns = ['Doc '+str(i) for i in range(0, len(data))]
term_document_matrix['total_count'] = term_document_matrix.sum(axis=1)
term_document_matrix = term_document_matrix.sort_values(by ='total_count',ascending=False)

term_document_matrix=term_document_matrix

#Mean Centering the Data
#TDM_meaned=term_document_matrix-np.mean(term_document_matrix, axis=0)

#Covariance Matrix
covariance_matrix=np.cov(term_document_matrix, rowvar=False)

#Eigendecomposition of Covariance Matrix
# Using np.linalg.eig function
eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

# Calculating the explained variance on each of components
variance_explained = []
for i in eigen_values:
     variance_explained.append((i/sum(eigen_values))*100)

#print(variance_explained)

# Identifying cumulative variance
cumulative_variance_explained = np.cumsum(variance_explained)
#print(cumulative_variance_explained)
```

```
#Sorting eigenvalues
sorted_index= np.argsort(eigen_values)[::-1]
sorted_eigenvalue=eigen_values[sorted_index]
sorted_eigenvectors = eigen_vectors[:,sorted_index]

total_num_topics= len (eigen_values[eigen_values>1])
print('Number of topics: ', total_num_topics)

#Finding the Elbow   Kneed algorithm finds point of maximum curvature
#!pip install --upgrade kneed
y = sorted_eigenvalue
x= range(1, len(y)+1)
from kneed import KneeLocator
kn = KneeLocator(x, y, curve='convex', direction='decreasing')
print('Number of Components: ', kn.knee)

plt.xlabel('Number of Components')
plt.ylabel('Eigenvalues')
plt.plot(x, y, 'bx-')
plt.xlim(0, 6)
plt.vlines(kn.knee, plt.ylim()[0], plt.ylim()[1], linestyles='dashed')
plt.xticks(range(1,6))
plt.show()



# Coherence score to determine k, number of topics

#Create Dictionary
def sent_to_words(sentences):
    for sentence in sentences:
        # deacc=True removes punctuations
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))

data = list(sent_to_words(data))
id2word = corpora.Dictionary(data)
corpus = [id2word.doc2bow(word) for word in data]

coherenceList_cv = []
num_topics_list = np.arange(1,7)
for num_topics in tqdm(num_topics_list):
    lda_model= gensim.models.LdaModel(alpha= 'auto', eta="auto", corpus=corpus,
id2word=id2word,
                                      num_topics=num_topics, random_state=42)
    cv = CoherenceModel(model=lda_model, corpus=corpus,
                        texts=data, dictionary=id2word, coherence='c_v')
    coherenceList_cv.append(cv.get_coherence())
```

```
plotData = pd.DataFrame({'Number of topics':num_topics_list,
                    'CoherenceScore':coherenceList_cv})
f,ax = plt.subplots(figsize=(10,6))
sns.set_style("darkgrid")
sns.pointplot(x='Number of topics',y= 'CoherenceScore',data=plotData)
plt.show()
```

Appendix C: Python Code for Zimm Approach

```
#Load Packages
import numpy as np
import pandas as pd
import re, gensim
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer  #oldest method developed 1979
from nltk.stem import WordNetLemmatizer
import gensim.corpora as corpora
from string import punctuation   #contains !"#$%&'()+,-./:;?@{}[]_^`~
# Sklearn
from sklearn.feature_extraction.text import CountVectorizer
from pprint import pprint

# Plotting tools
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import seaborn as sns

#Import Data
df = pd.read_json('https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json')
#print(df.target_names.unique())

#Filters out rec.sport.hockey files
#baseball = df[df["target_names"].str.contains("rec.sport.baseball")]
#hockey = df[df["target_names"].str.contains("rec.sport.hockey")]
#space = df[df["target_names"].str.contains("sci.space")]
autos = df[df["target_names"].str.contains("rec.autos")]
med = df[df["target_names"].str.contains("sci.med")]

df = [autos,med]
df = pd.concat(df)

#Preprocessing
# Convert to list
data = df.content.values.tolist()

#Remove extra spaces
for i in range(len(data)):
    data[i]=" ".join(data[i].split())

# Remove Emails
data = [re.sub('\b*@\b*\b?', '', sent) for sent in data]

# Remove new line characters
data = [re.sub('\b', ' ', sent) for sent in data]

# Remove distracting single quotes
data = [re.sub("\'", "", sent) for sent in data]
```

```
#Remove punctuation
data = [re.sub('['+punctuation+']',' ', sent) for sent in data]


exclude ='\\'
for i in range(len(data)):
   data[i] = ''.join(sent for sent in data [i] if sent not in exclude)

#Make Lower case
for i in range(len(data)):
   data [i] = data [i].lower()  #Converts to lower case


#Lemmatize
#Data Cleansing
for i in range(len(data)):
   data [i] = ''.join([WordNetLemmatizer().lemmatize(word) for word in data[i]])#Lemmatize

#Stem
#Data Cleansing
for i in range(len(data)):
   data [i] = ''.join([PorterStemmer().stem(word) for word in data[i]]) #Stem

#Remove Numbers
for i in range(len(data)):
   data [i] = ''.join([word for word in data[i] if not word.isdigit()])


#Remove single characters
for i in range(len(data)):
   data [i] = re.sub(r'\b[a-zA-Z]\b',' ',data [i])   # Removes single characters


#Remove words with length of 3 or less
for i in range(len (data)):
   data[i] = re.sub(r'\b\w{1,3}\b','', data[i])


#Remove stopwords
#Stopword list creation
stop_words = stopwords.words("english")
custom_stop_words =['from','re', 'subject','organization',
               'line','article','write','nntp','know','people',
               'host','dont','think','reply','make','thing','time',
               'distribution','much','well','university','want',
               'anyone','lines','writes','posting','good','even',
               'year','problem','many','really','would','like',
               'also','could','used','take','said','better','still',
              'something','sure','cant']
```

```
stop_words = custom_stop_words + stop_words

for i in range(len(data)):
    data [i] = ' '.join([word for word in data[i].split(' ') if word not in stop_words]) #Removes
stopwords

#Remove extra spaces
for i in range(len(data)):
    data[i]=" ".join(data[i].split())

#Create WordCloud
#change value to black
def black_color_func(word, font_size, position, orientation, random_state=None, **kwargs):
    return ("hsl(0,100%,1%)")

#convert list to string and generate
unique_string=(" ").join(data)
from PIL import Image
background_image=np.array(Image.open('C://Users/jzim2/Desktop/Dissertation/test.jpg'))
wordcloud = WordCloud(prefer_horizontal = 1.0, background_color="white",
mask=background_image, width = 1000, height = 500, collocations =
False).generate(unique_string)
wordcloud.recolor(color_func=black_color_func)
plt.figure(figsize=(15,8))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

# Count Vectorizer
vect = CountVectorizer()
vects = vect.fit_transform(data)

# Select the rows from the data set
td= pd.DataFrame(vects.todense()).iloc[:len(data)]
td.columns = vect.get_feature_names()
term_document_matrix = td.T
term_document_matrix.columns = ['Doc '+str(i) for i in range(0, len(data))]
term_document_matrix['total_count'] = term_document_matrix.sum(axis=1)

term_document_matrix=term_document_matrix.T

#Mean Centering the Data
TDM_meaned=term_document_matrix-np.mean(term_document_matrix, axis=0)

#Covariance Matrix
covariance_matrix=np.cov(TDM_meaned, rowvar=False)

#Eigendecomposition of Covariance Matrix
# Using np.linalg.eig function
```

```
eigen_values, eigen_vectors = np.linalg.eigh(covariance_matrix)


#Retreieve normalized eigenvectors that correspond to eigenvaules greater than 1
#count number of eigenvalues greater than one
num_topics= len (eigen_values[eigen_values>1])

sorted_index = np.argsort(eigen_values)[::-1]
sorted_eigenvalue =eigen_values[sorted_index]
sorted_eigenvectors=eigen_vectors[:,sorted_index]

#Round eigen values to eight places.
sorted_eigenvalue= [np.round(x,8) for x in sorted_eigenvalue]


#Eigenvectors for number of topics
eigenvector_subset=sorted_eigenvectors[:,0:num_topics]
eigenvalue_subset=sorted_eigenvalue[0:num_topics]

loadings= (eigenvector_subset) * np.sqrt(eigenvalue_subset)

loading_matrix=pd.DataFrame(loadings, columns=['Topic{}'.format(i) for i in range(1,
num_topics+1)],
                    index=term_document_matrix.columns)

#Divide Loadings Matrix into individual lists
Component=[]
y = loading_matrix
x= range(1, len(y)+1)

columncount = len(loading_matrix.columns)

for i in range(0,columncount):
   Component_i = loading_matrix.iloc[:,i].copy()
   Component.append(loading_matrix.iloc[:,i].copy())

#Sort Loadings Biggest to Smallest
for i in range(len(Component)):
   Component [i] = Component [i].sort_values(ascending=False)

#Finding the Elbow  for loadings  Kneed algorithm finds point of maximum curvature
#!pip install --upgrade kneed
from kneed import KneeLocator

t=[]
k=[]
for i in range(len(Component)):
   l = range(0, len(Component[i]+1))
```

```
for i in range(len(Component)):
   t = Component [i]
   kn = KneeLocator(l, t, curve='convex', direction='decreasing')
   k.append(kn.knee)
   print('Number of Components: ', kn.knee)

dusty = np.array(k)

#Print Entire Matrix of Words for each component
df = loading_matrix
v = loading_matrix.values
i = loading_matrix.index.values
q = len(x)

y=pd.DataFrame(i[v.argsort(0)[::-1]][:q], columns=df.columns)

#Print Number of entries in each column match array value
#Divide Loadings Matrix into individual lists
FullMatrix=[]
columncount_FullMatrix = len(y.columns)

for i in range(0,columncount_FullMatrix):
   FullMatrix_i = y.iloc[:,i].copy()
   FullMatrix.append(y.iloc[:,i].copy())

#FinalResults=[]
for i in range(len(FullMatrix)):
   row = dusty[i]
   FullMatrix [i]= pd.DataFrame(FullMatrix[i], index=range(row))
   #FinalResults.append(FullMatrix)

#Export the results to Excel, each Topic has its own Tab
from pandas import ExcelWriter

def save_xls(list_dfs, xls_path):
   with ExcelWriter(xls_path) as writer:
      for n, df in enumerate(list_dfs):
         df.to_excel(writer, 'Topic%s' %n)

save_xls(FullMatrix, r'C:\Users\jzim2\Desktop\Dissertation\Paper3\FullMatrix.xls' )

#LDA for Comparison
from gensim.models.coherencemodel import CoherenceModel
from tqdm._tqdm_notebook import tqdm

#Create Dictionary
def sent_to_words(sentences):
   for sentence in sentences:
      # deacc=True removes punctuations
```

```
    yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))

data = list(sent_to_words(data))
id2word = corpora.Dictionary(data)
corpus = [id2word.doc2bow(word) for word in data]


coherenceList_cv = []
num_topics_list = np.arange(1,14)

for num_topics in tqdm(num_topics_list):
  lda_model= gensim.models.LdaModel(alpha= 'auto', eta="auto", corpus=corpus,
id2word=id2word,
                                  num_topics=num_topics, random_state=42)
  cv = CoherenceModel(model=lda_model, corpus=corpus,
                          texts=data, dictionary=id2word, coherence='c_v')
  coherenceList_cv.append(cv.get_coherence())
for index, topic in lda_model.show_topics(formatted=False, num_words=10, num_topics=13):
  print('Topic: {} \nwords: {}'.format(index, [w[0] for w in topic]))

pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]

plotcvData= pd.DataFrame({'Number of topics':num_topics_list,
                      'Full BoW':coherenceList_cv})

f,ax = plt.subplots(figsize=(10,6))
sns.set_style("darkgrid")
plot = sns.pointplot(x='Number of topics',y= 'Full BoW',data=plotcvData)
plot.set_ylabel("Coherence Score")
plt.show()
```

Bibliography

Abdi, H. and Williams, L.J., 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), pp.433-459.

Agrawal, A., Fu, W. and Menzies, T., 2018. What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information and Software Technology*, *98*, pp.74-88.

Albalawi, R., Yeap, T.H. and Benyoucef, M., 2020. Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence*, *3*, p.42.

Aletras, N. and Stevenson, M., 2013, March. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers* (pp. 13-22).

Alghamdi, R. and Alfalqi, K., 2015. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, *6*(1).

Anthes, G., 2010. Topic models vs. unstructured data. *Communications of the ACM*, *53*(12), pp.16-18.

Balakrishnan, V. and Lloyd-Yemoh, E., 2014. Stemming and lemmatization: a comparison of retrieval performances.

Binkley, D., Heinz, D., Lawrie, D. and Overfelt, J., 2014, June. Understanding LDA in source code analysis. In *Proceedings of the 22nd international conference on program comprehension* (pp. 26-36).

Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, pp.993-1022.

Boulis, C. and Ostendorf, M., 2005, April. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In *Proc. of the International Workshop in Feature Selection in Data Mining* (pp. 9-16). Citeseer.

Bouma, G., 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, *30*, pp.31-40.

Brahma, A., Goldberg, D.M., Zaman, N. and Aloiso, M., 2021. Automated mortgage origination delay detection from textual conversations. Decision Support Systems, 140, p.113433.

Cai, D., He, X., Wu, X. and Han, J., 2008, December. Non-negative matrix factorization on manifold. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 63-72). IEEE.

Chae, B. and Olson, D., 2021, A Topical Exploration of the Intellectual Development of *Decision Sciences* 1975–2016. Decision Sciences. https://doi.org/10.1111/deci.12326.

Cohen, R. and Ruths, D., 2013. Classifying political orientation on Twitter: It's not easy!. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 7, No. 1, pp. 91-99).

Coussement, K. and Benoit, D.F., 2021. Interpretable data science for decision making. *Decision Support Systems*, *150*, p.113664.

Cuypers, I.R., Hennart, J.F., Silverman, B.S. and Ertug, G., 2021. Transaction cost theory: Past progress, current challenges, and suggestions for the future. *Academy of Management Annals*, *15*(1), pp.111-150.

Dahal, B., Kumar, S.A. and Li, Z., 2019. Topic modeling and sentiment analysis of global climate change tweets. *Social Network Analysis and Mining*, *9*(1), pp.1-20.

De Waal, A. and Barnard, E., 2008. Evaluating topic models with stability.

Doshi-Velez, F., Wallace, B. and Adams, R., 2015, February. Graph-sparse lda: a topic model with structured sparsity. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29, No. 1).

Eassom, H. (2017). How to Choose Effective Keywords for Your Article. The Wiley Network. https://www.wiley.com/network/researchers/preparing-your-article/how-to-choose-effective-keywords-for-your-article.

Fagin, R., Kumar, R., Sivakumar, D.: Comparing top *k* lists. SIAM Journal on Discrete Mathematics 17(1), 134–160 (2003)

Feuerriegel, S. and Pröllochs, N. (2021), Investor Reaction to Financial Disclosures across Topics: An Application of Latent Dirichlet Allocation*. Decision Sciences, 52: 608-628. https://doi.org/10.1111/deci.12346.

Fu, Q., Zhuang, Y., Gu, J., Zhu, Y., Qin, H., & Guo, X. (2019, December). Search for K: assessing five topic-modeling approaches to 120,000 Canadian articles. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 3640-3647). IEEE.

George, L.E. and Birla, L., 2018, June. A study of topic modeling methods. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 109-113). IEEE.

Gerlach, M., Peixoto, T.P. and Altmann, E.G., 2018. A network approach to topic models. *Science advances*, *4*(7), p.eaaq1360.

Geva, T. and Zahavi, J., 2014. Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news. *Decision support systems*, *57*, pp.212-223.

Greene, D., O'Callaghan, D. and Cunningham, P., 2014, September. How many topics? stability analysis for topic models. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 498-513). Springer, Berlin, Heidelberg.

Griffiths, T.L. and Steyvers, M., 2004. Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*(suppl 1), pp.5228-5235.

Grimmer, J., 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis*, *18*(1), pp.1-35.

Groth, S.S. and Muntermann, J., 2011. An intraday market risk management approach based on textual analysis. *Decision Support Systems*, *50*(4), pp.680-691.

Hamzeian, D., 2021. *Using Machine Learning Algorithms for Finding the Topics of COVID-19 Open Research Dataset Automatically* (Master's thesis, University of Waterloo).

Hasan, M., Rahman, A., Karim, M., Khan, M., Islam, S. and Islam, M., 2021. Normalized approach to find optimal number of topics in Latent Dirichlet Allocation (LDA). In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering* (pp. 341-354). Springer, Singapore.

Hirshleifer, D., Levi, Y., Lourie, B. and Teoh, S.H., 2019. Decision fatigue and heuristic analyst forecasts. *Journal of Financial Economics*, *133*(1), pp.83-98.

Hofmann, T., 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, *42*(1), pp.177-196.

Hogenboom, F., Frasincar, F., Kaymak, U., De Jong, F. and Caron, E., 2016. A survey of event extraction methods from text for decision support systems. *Decision Support Systems*, *85*, pp.12-22.

Hong, L. and Davison, B.D., 2010, July. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics* (pp. 80-88).

Hyndman, K.B. and Menezes, M.B., 2021. Behavioral pitfalls of product proliferation in supply chains: An experimental study. Decision Sciences.

Ito, J., Song, J., Toda, H., Koike, Y. and Oyama, S., 2015, May. Assessment of tweet credibility with LDA features. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 953-958).

Jiang, Y., Song, X., Harrison, J., Quegan, S. and Maynard, D., 2017, September. Comparing Attitudes to Climate Change in the Media using sentiment analysis based on Latent Dirichlet Allocation. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism* (pp. 25-30).

Jivani, A.G., 2011. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6), pp.1930-1938.

Koch, H., Christensen, J.A., Frandsen, R., Zoetmulder, M., Arvastson, L., Christensen, S.R., Jennum, P. and Sorensen, H.B., 2014. Automatic sleep classification using a data-driven topic model reveals latent sleep states. *Journal of neuroscience methods*, *235*, pp.130-137.

Kulkarni, S.S., Apte, U.M. and Evangelopoulos, N.E. (2014), The Use of Latent Semantic Analysis in Operations Management Research. Decision Sciences, 45: 971-994. https://doi.org/10.1111/deci.12095.

Lasi, H., Fettke, P., Kemper, H.G., Feld, T. and Hoffmann, M., 2014. Industry 4.0. *Business & information systems engineering*, *6*(4), pp.239-242.

Lebanon, G., Mao, Y. and Dillon, J., 2007. The Locally Weighted Bag of Words Framework for Document Representation. *Journal of Machine Learning Research*, *8*(10).

Lesnikowski, A., Belfer, E., Rodman, E., Smith, J., Biesbroek, R., Wilkerson, J.D., Ford, J.D. and Berrang-Ford, L., 2019. Frontiers in data analytics for adaptation research: Topic modeling. *Wiley Interdisciplinary Reviews: Climate Change*, *10*(3), p.e576.

Li, X., Wang, Y., Zhang, A., Li, C., Chi, J. and Ouyang, J., 2018. Filtering out the noise in short text topic modeling. *Information Sciences*, *456*, pp.83-96.

Ljungberg, B.F., 2019. Dimensionality reduction for bag-of-words models: PCA vs LSA. *Semanticscholar. org*.

MacMillan, K. and Wilson, J.D., 2017. Topic supervised non-negative matrix factorization. *arXiv preprint arXiv:1706.05084*.

Mantyla, M.V., Claes, M. and Farooq, U., 2018, October. Measuring LDA topic stability from clusters of replicated runs. In *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement* (pp. 1-4).

Martins, C.A., Monard, M.C. and Matsubara, E.T., 2003. Reducing the dimensionality of bag-of-words text representation used by learning algorithms. In *Proc of 3rd IASTED International Conference on Artificial Intelligence and Applications* (pp. 228-233).

Mimno, D., Wallach, H., Talley, E., Leenders, M. and McCallum, A., 2011, July. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 262-272).

Misra, H., Cappé, O. and Yvon, F., 2008, August. Using LDA to detect semantically incoherent documents. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning* (pp. 41-48).

Mendoza, M., Alegría, E., Maca, M., Cobos, C. and León, E., 2015. Multidimensional analysis model for a document warehouse that includes textual measures. Decision Support Systems, 72, pp.44-59.

Murphy, K.P., 2006. Naive bayes classifiers. *University of British Columbia*, *18*(60), pp.1-8.

Mustak, M., Salminen, J., Plé, L. and Wirtz, J., 2021. Artificial intelligence in marketing: Topic modeling, scientometric analysis, and research agenda. *Journal of Business Research*, *124*, pp.389-404.

Newman, D., Lau, J.H., Grieser, K. and Baldwin, T., 2010, June. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics* (pp. 100-108).

Newman, D., Noh, Y., Talley, E., Karimi, S. and Baldwin, T., 2010, June. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries* (pp. 215-224).

Ozsoy, M.G., Alpaslan, F.N. and Cicekli, I., 2011. Text summarization using latent semantic analysis. *Journal of Information Science*, *37*(4), pp.405-417.

Passalis, N. and Tefas, A., 2016. Entropy optimized feature-based bag-of-words representation for information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, *28*(7), pp.1664-1677.

Péladeau, N. and Davoodi, E., 2018, January. Comparison of latent Dirichlet modeling and factor analysis for topic extraction: A lesson of history. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Pignatiello, G.A., Martin, R.J. and Hickman Jr, R.L., 2020. Decision fatigue: A conceptual analysis. *Journal of health psychology*, *25*(1), pp.123-135.

Quinn, K.M., Monroe, B.L., Colaresi, M., Crespin, M.H. and Radev, D.R., 2010. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, *54*(1), pp.209-228.

Razmi, N.A., Zamri, M.Z., Ghazalli, S.S.S. and Seman, N., 2021. Visualizing stemming techniques on online news articles text analytics. *Bulletin of Electrical Engineering and Informatics*, *10*(1), pp.365-373.

Röder, M., Both, A. and Hinneburg, A., 2015, February. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining* (pp. 399-408).

Rogers, A., Drozd, A. and Li, B., 2017, August. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)* (pp. 135-148).

Schofield, A. and Mimno, D., 2016. Comparing apples to apple: The effects of stemmers on topic models. *Transactions of the Association for Computational Linguistics*, *4*, pp.287-300.

Selva86, 2018, 20 NewsGroups Training Data, https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json.

Shahbazi, Z. and Byun, Y.C., 2020. Topic modeling in short-text using non-negative matrix factorization based on deep reinforcement learning. *Journal of Intelligent & Fuzzy Systems*, *39*(1), pp.753-770.

Singh, K.N., Devi, S.D., Devi, H.M. and Mahanta, A.K., 2022. A novel approach for dimension reduction using word embedding: An enhanced text classification approach. International Journal of Information Management Data Insights, 2(1), p.100061.

Slof, D., Frasincar, F. and Matsiiako, V., 2021. A competing risks model based on latent Dirichlet Allocation for predicting churn reasons. Decision Support Systems, 146, p.113541.

Solis, Brian. (2020). Information Overload, Why it Matters and How to Combat it. Retrieved from https://www.interaction-design.org/literature/article/information-overload-why-it-matters-and-how-to-combat-it.

Sollisch J (2016) The cure for decision fatigue. *Wall Street Journal*. https://www.wsj.com/articles/the-cure-for-decision-fatigue-1465596928.

Stevens, K., Kegelmeyer, P., Andrzejewski, D. and Buttler, D., 2012, July. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 952-961).

Syed, S. and Spruit, M., 2017, October. Full-text or abstract? Examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International conference on data science and advanced analytics (DSAA)* (pp. 165-174). IEEE.

Taghva, K., Borsack, J., Condit, A. and Erva, S., 1994. The effects of noisy data on text retrieval. *Journal of the American Society for Information Science*, *45*(1), pp.50-58.

Thomas, S.W., Adams, B., Hassan, A.E. and Blostein, D., 2011, May. Modeling the evolution of topics in source code histories. In *Proceedings of the 8th working conference on mining software repositories* (pp. 173-182).

Tiwana, A., Wang, J., Keil, M. and Ahluwalia, P., 2007. The bounded rationality bias in managerial valuation of real options: Theory and evidence from IT projects. *Decision Sciences*, *38*(1), pp.157-181.

Tseng, F.S. and Chou, A.Y., 2006. The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. Decision Support Systems, 42(2), pp.727-744.

Vayansky, I. and Kumar, S.A., 2020. A review of topic modeling methods. *Information Systems*, *94*, p.101582.

Vemprala, N., Liu, C.Z. and Choo, K.K.R., 2021. From puzzles to portraits: Enhancing situation awareness during natural disasters using a design science approach. Decision Sciences.

Wallach, H.M., 2006, June. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning* (pp. 977-984).

Williamson, O.E., 1979. Transaction-cost economics: the governance of contractual relations. *The journal of Law and Economics*, *22*(2), pp.233-261.

Xu, Y., Heller, K., & Ghahramani, Z. (2009, April). Tree-based inference for Dirichlet process mixtures. In *Artificial Intelligence and Statistics* (pp. 623-630). PMLR.

Yan, X., Guo, J., Liu, S., Cheng, X. and Wang, Y., 2013, May. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *proceedings of the 2013 SIAM International Conference on Data Mining* (pp. 749-757). Society for Industrial and Applied Mathematics.

Yin, Z. and Shen, Y., 2018. On the dimensionality of word embedding. Advances in neural information processing systems, 31.

Zamani, M., Schwartz, H.A., Eichstaedt, J., Guntuku, S.C., Ganesan, A.V., Clouston, S. and Giorgi, S., 2020, November. Understanding weekly COVID-19 concerns through dynamic content-specific LDA topic modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing* (Vol. 2020, p. 193). NIH Public Access.

Zawadzki, P. and Żywicki, K., 2016. Smart product design and production control for effective mass customization in the Industry 4.0 concept. *Management and production engineering review*.

Zhang, Y., Chen, M., Huang, D., Wu, D. and Li, Y., 2017. iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Generation Computer Systems*, *66*, pp.30-35.

Zhang, Y., Jin, R. and Zhou, Z.H., 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, *1*(1-4), pp.43-52.

Zhao, S., 2021. Thumb Up or Down? A Text-Mining Approach of Understanding Consumers through Reviews. Decision Sciences, 52(3), pp.699-719.

Zhao, W., Chen, J.J., Perkins, R., Liu, Z., Ge, W., Ding, Y. and Zou, W., 2015, December. A heuristic approach to determine an appropriate number of topics in topic modeling. In *BMC bioinformatics* (Vol. 16, No. 13, pp. 1-10). BioMed Central.

Zhao, R. and Mao, K., 2017. Fuzzy bag-of-words model for document representation. *IEEE transactions on fuzzy systems*, *26*(2), pp.794-804.

Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., and Xu, K., 2016, August. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2105-2114).

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
13-05-2022	Doctoral Dissertation	September 2019 – May 2022

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Innovative Heuristics to Improve the Latent Dirichlet Allocation Methodology for Textual Analysis and a New Modernized Topic Modeling Approach	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Zimmermann, Jamie, T., Major, USAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765	AFIT-ENS-DS-22-J-059
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Natural Language Processing is a complex method of data mining the vast trove of documents created and made available every day. Topic modeling seeks to identify the topics within textual corpora with limited human input into the process to speed analysis. Current topic modeling techniques used in Natural Language Processing have limitations in the pre-processing steps. This dissertation studies topic modeling techniques, those limitations in the pre-processing, and introduces new algorithms to gain improvements from existing topic modeling techniques while being competitive with computational complexity.

This research introduces four contributions to the field of Natural Language Processing and topic modeling. First, this research identifies a requirement for a more robust "stopwords" list and proposes a heuristic for creating a more robust list. Second, a new dimensionality-reduction technique is introduced that exploits the number of words within a document to infer importance to word choice. Third, an algorithm is developed to determine the number of topics within a corpus and is demonstrated using a standard topic modeling data set. These techniques produce a higher quality result from the Latent Dirichlet Allocation topic modeling technique. Fourth, a novel heuristic utilizing Principal Component Analysis is introduced that is capable of determining the number of topics within a corpus that produces stable sets of topic words.

15. SUBJECT TERMS

Natural Language Processing, Textual Analysis, Term Frequency-Inverse Document Frequency, Latent Dirichlet Allocation, Principal Component Analysis, Word Clouds, Data Mining

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Dr. Lance E Champagne , AFIT/ENS
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	109	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, ext 4646 lance.champagne@afit.edu
U	U	U			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18