Text Mining

Noah Benedict

Department of Mathematical Sciences
Montana State University

May 3, 2019

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Noah Benedict

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date

Jennifer L. Green
Writing Project Advisor

Date

Mark C. Greenwood
Writing Projects Coordinator

Contents

Abstract

Textual information is abundant and easily accessible. Despite this, analysis of text is rarely covered in a statistics program. This paper gives an overview of text mining by discussing a framework that conceptualizes the field as eight areas of practice: text preprocessing, document clustering, document classification, information retrieval, information extract, web mining, natural language processing, and concept extraction. To show some of these concepts in practice, an analysis of survey data from education professional development sessions is then presented, illustrating the use of data preprocessing, exploratory visuals, and a modeling technique called latent Dirichlet allocation.

1 Introduction

Data are prolific and the amount that needs to be explored and analysed is increasing every day. Yet a majority of these data are not stored in structured formats of a database or a spreadsheet (Reamy 2016). Large quantities of information are stored in unstructured formats such as text, causing difficulties in extracting useful information. Text information is easy to create as it is easy for people to pick up a pen or computer for writing, but it is difficult to structure text into common data analysis formats like a spreadsheet. It can be overwhelming to think about all the knowledge that has been left underutilized from text sources because of these difficulties.

In response to these changes, individuals from a variety of disciplines have developed techniques for structuring and analyzing textual data (Miner et al. 2012). These techniques and skills help unlock information from a wider variety of data, but such skills are not typically taught in an undergraduate or graduate statistics curriculum. This document provides a brief overview of what text analytics/text mining is, an overview of different topics related to the field, and a more in-depth look at several techniques applied on an educational dataset.

2 What is Text Mining/Text Analytics?

With so many questions that can be asked of textual data, text mining techniques have been developed by a wide variety of disciplines to tackle their own specific problems that arise (Miner et al. 2012). Consequently, many terms have been developed to describe similar things, creating disputes on the appropriate use of a term. For example, some authors and practitioners (e.g., Aggarwal 2018; Miner et al. 2012) use words like text analytics, text mining, and machine learning from text interchangeably while others see these words as each having their own unique meaning. This document will use the term text mining and text analytics as the former, adopting the general definition that text mining is the process of taking unstructured textual data and extracting useful information through patterns and trends.

3 Practice Areas of Text Mining

This paper gives an overview of different practice fields across text analytics, however authors do not split up aspects of text mining consistently. Miner et al. (2012) relate text mining to the wild west of analytics. Text mining has emerged from many disciplines in which specific applications are needed. Figure 1 displays the seven practice areas of text mining Miner et al. defined and the disciplines that primarily use text analytic methodologies related to that practice area. These breakdowns are commonly seen in the literature with some authors taking a data mining approach to text and primarily discussing clustering and classification techniques (e.g., Berry et al. 2008; Jo 2019; Nedjah et al. 2009), others discussing information retrieval, and others tackling some subset of practice areas.

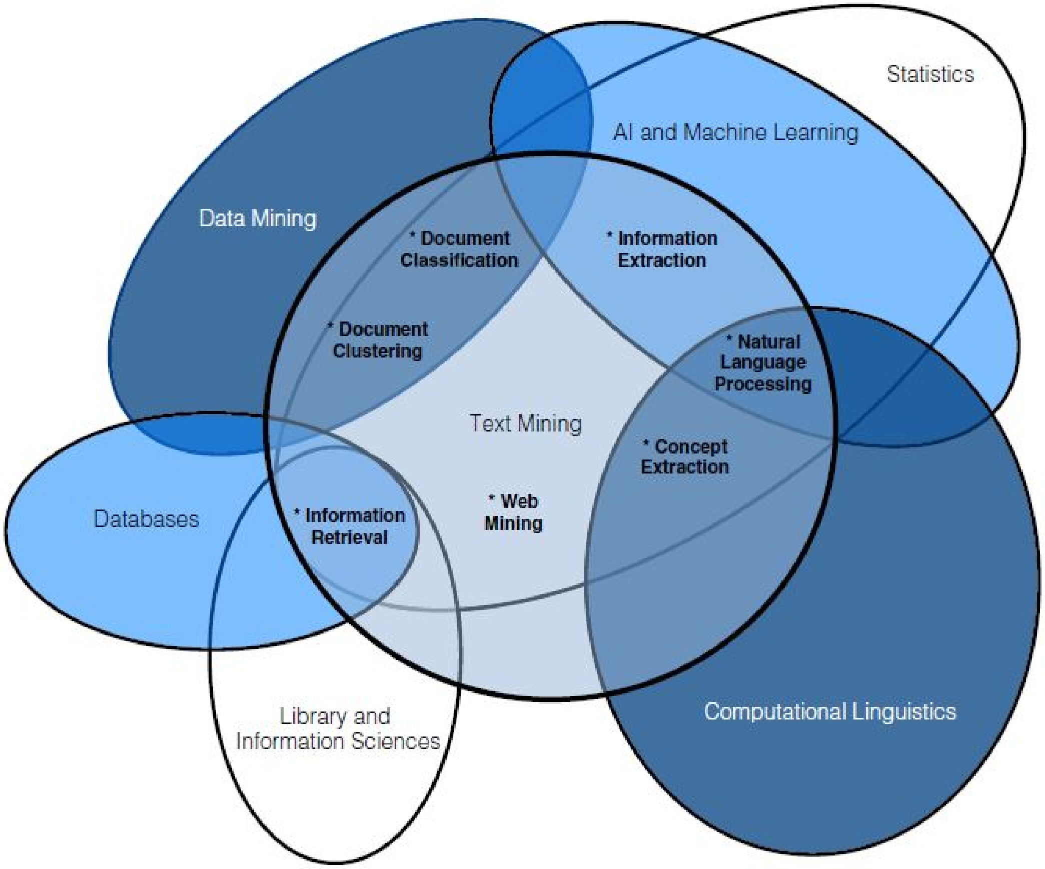


Figure 1: Representation of different disciplinary contributions and uses of text mining practice areas. Adapted from Miner et al., 2012, p. 31.

While many authors define the overarching structure of text analytics as groups of algorithms that are mutually exclusive, Miner et al. (2012) take an approach where practice areas of text mining are fluid and can overlap.

Topics and algorithms used to answer text-related questions may be related to multiple practice areas. A visual representation based on how Miner et al. classify different aspects of text mining and their overlap can be seen in Figure 2 (Miner et al. 2012). For example, algorithms for document ranking share overlapping characteristics from document classification, web mining, and information retrieval.

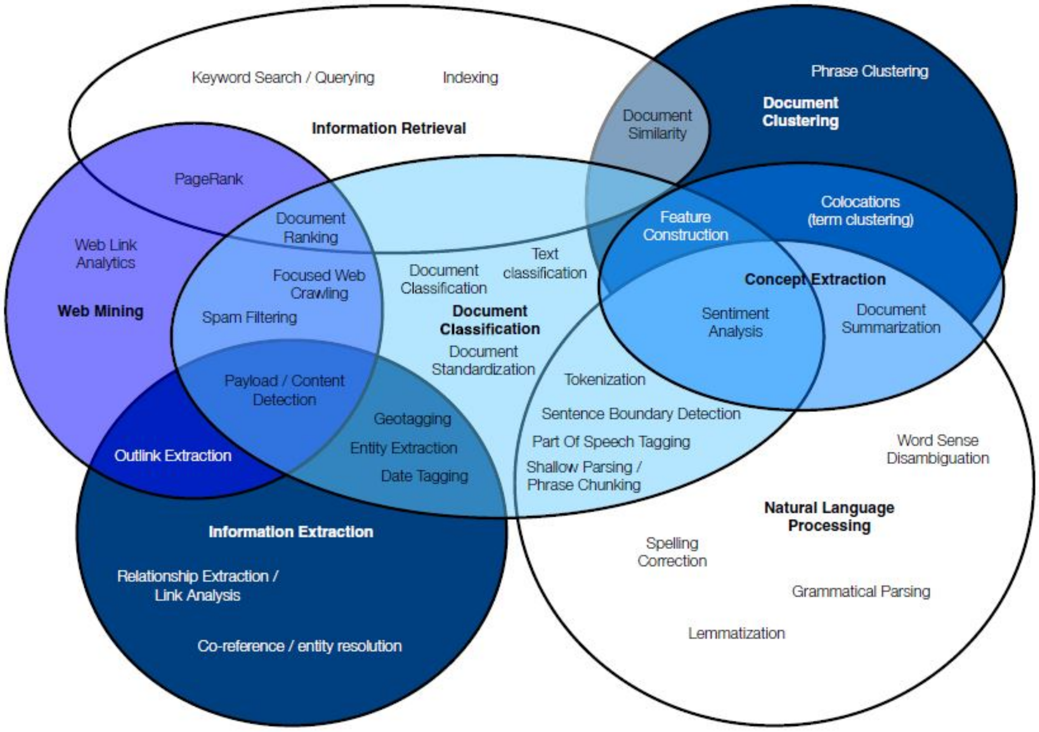


Figure 2: Overlapping connections between practice areas and their relationship to different text mining algorithims. Adapted from Miner et al., 2012, p. 38.

This paper uses the classifications of Miner et al. (2012) in which text mining topics and algorithms are placed under seven different practice areas: document clustering, document classification, information extraction, natural language processing, concept extraction, web mining, and information retrieval.

An eighth practice area, text preprocessing, which is not addressed by Miner et al. is also discussed. For different views on structuring practice areas of text mining please refer to *Machine Learning for Text* (Aggarwal 2018) and *Mining Text Data* (Miner et al. 2012).

3.1 Overview of Practice Areas

This section briefly defines each of the eight practice areas before going into more depth on each topic.

Text Preprocessing - Before analysis of textual data can be done, documents and text have to be collected and processing is needed to turn a stream of words into quantifiable bits for different types of analysis. How text is processed before analysis can have large effects on the results so having an understanding of text preprocessing is important to all practice areas of text mining.

Document Clustering - Document classification looks at finding ways to group words, chunks of textual information, or documents (Miner et al. 2012). These grouping are typically constructed in an unsupervised fashion, where groups are not known before being constructed. The assignment to a group does not need to be a hard yes/no but can be a probabilistic measure (Aggarwal 2018).

Document Classification - Document classification takes textual information and partitions it into known groups or labels (Aggarwal 2018). This is guided by taking data with known labels as a training dataset before classifying unknown groupings of text.

Information Retrieval - This involves finding and retrieving text documents of interest based on keyword searches (Miner et al. 2012). With an abundance of text information available, algorithms need to identify documents of interest and determine the order of their importance to the searcher to make decisions on which documents should be shared first (Aggarwal 2018).

Information Extraction - The process of information extraction involves identifying and extracting entities from unstructured text to construct formal databases (Miner et al. 2012). Entities are real-word objects such as names of people, locations, or dates and times. Identifying entities can provide more interpretable results from text analysis as these real-world concepts contain more meaning than their unidentified counterparts.

Web Mining - Information on the web adds a level of complexity to analysis due to the large quantities of web-related information such as document interconnections through hyperlinks, and the addition of very heterogeneous data (web pages, social media posts, inclusion of multimedia, cross-lingual data) (Aggarwal 2018; Miner et al. 2012). Web mining forms links between these different data sources so all different types of data can be used in conjunction with text to answer questions.

Natural Language Processing - The power of natural language processing comes from the addition of linguistic structure to text analysis. Many text analytic approaches ignore linguistic information such as parts of speech and phrase boundaries, but adding this info to the analysis provides additional insight to help in analysis accuracy or extraction of human context (Miner et al. 2012).

Concept Extraction - Grouping words and phrases using similarities of meaning. The addition of adding semantics and abstraction can help understand attitudes based on certain words found in text or summarize a document based on a common theme (Miner et al. 2012).

3.2 Text Preprocessing

Text preprocessing turns a document's flow of text into the quantifiable chunks needed for analysis, making it an integral part of the text mining process. As the first stage of the text analysis, preprocessing text requires decisions that have a lasting impact on the rest of the analysis. As a note, some consider scraping textual information from databases or the web as part of text prepreocessing, however this will not be covered in this section. Instead, this section will focus on the steps involved after a collection of documents has been digitized and collected. As seen in Figure 3 there are three main steps to turn the unstructured text from documents into a quantifiable list. These steps include tokenization, stemming, and stop word removal (Jo 2019).

The first step in this process is tokenization. This involves taking the stream of text from the documents and breaking it up into meaningful units called tokens. What is considered meaningful depends on the question and how an individual plans on answering that question. Text can be split into sentences, paragraphs, words, n-grams (n-sized clusters of words), or even letters to be considered as tokens (Silge & Robinson 2017). Since most analyses look at token frequency, it can difficult to find connections between long strings of words and letters tend to not have enough meaning associated with them. As

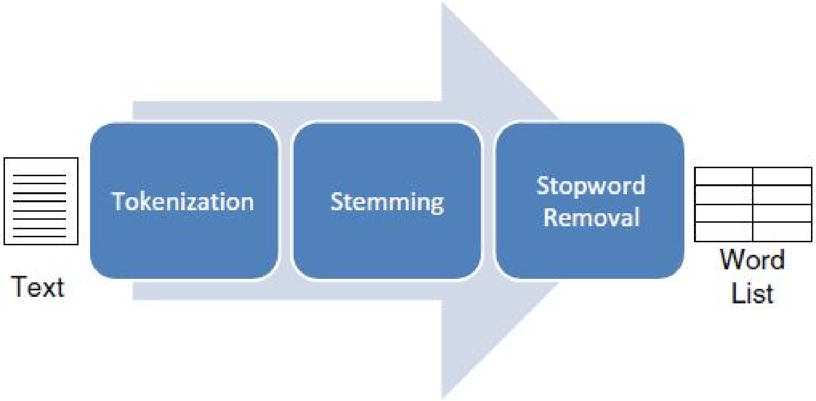


Figure 3: Three stages of text preprocessing to turn streams of text into quantifiable text for analysis. Adapted from Jo, 2019, p. 20.

such, the most commonly used token of analysis is the word, which is common enough to see occurrences across documents but still contains enough human context to pull meaningful insight from an analysis.

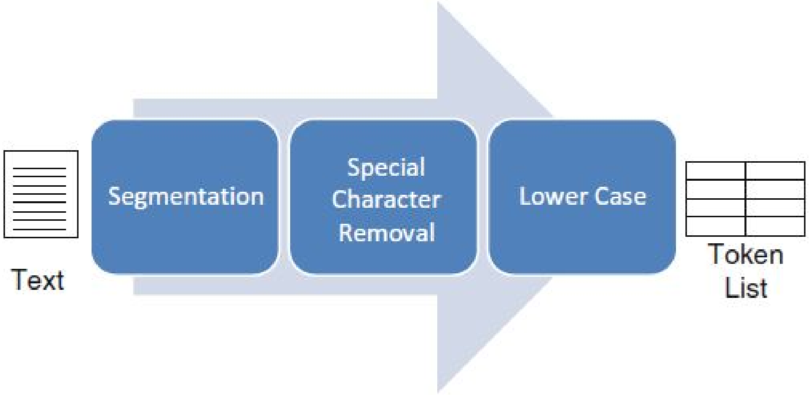


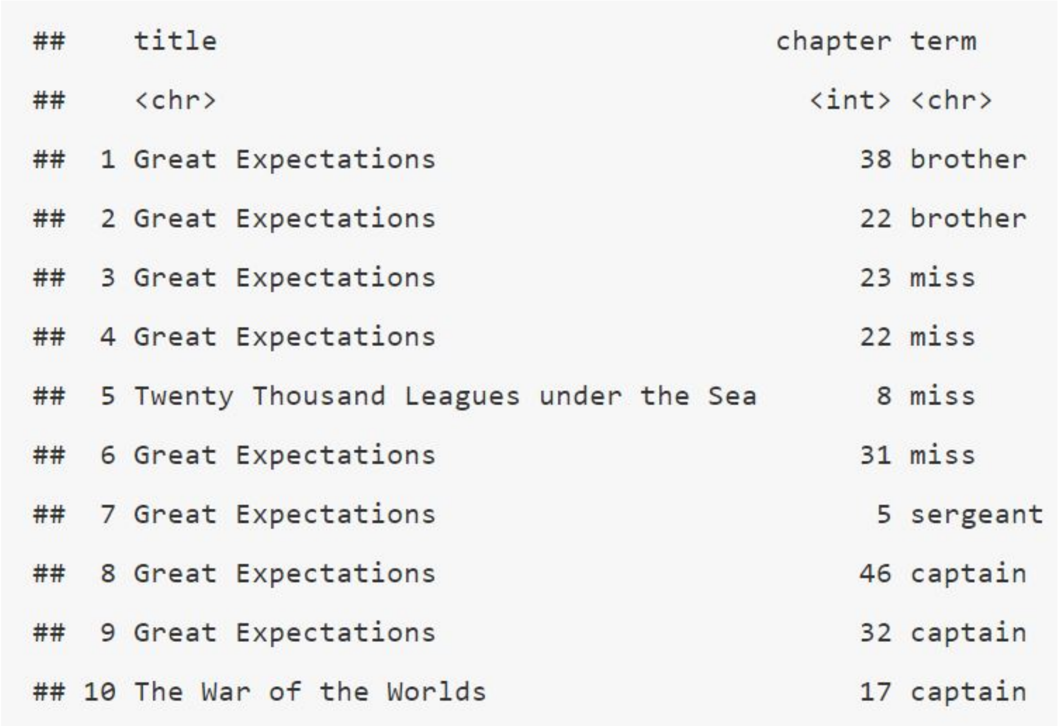
Figure 4: Three parts of the process of turning streams of text into a token list. Adapted from Jo, 2019, p. 22.

There are three considerations to make when turning text into tokens and these are visualized in Figure 4. The first step is segmentation of the text. This looks at how the text is going to be split into tokens. In many languages, but not all, word tokens can be identified by white spaces and sentence tokens identified by punctuation (Jo 2019). However, this process can result in errors unless the parser that splits the text is very complex. Typos may result in words without spaces between them being considered as one token instead of as two. Other types of tokens, such as sentence tokens, can become hard to identify when punctuation is used in abbreviations as well as to end a sentence.

Once the text is segmented, special characters such as hyphens, commas, quotes, and punctuation are removed so that word tokens such as “cats.”, “’cats”, and “cats” are all considered the token “cats”. It is important to note context can be lost when special characters are removed from words like “cat’s” and “cats”’ which give distinctions of cat being plural and possessive, but are treated as the word “cats”.

The last step is to make all characters lowercase. This makes words that begin sentences the same as those in the middle of the sentence. This does remove meaning from words like proper nouns. For example, a musical called “Cats” will now be the same word as Uncle John’s pet “cat”. The final token list may look like Figure 5 where each token has its own row and important information such as the book title and chapter where it can be tracked.

Once the tokens have been separated from the stream of text, the next step is stemming, which aims to identify a common base for words (Jo 2019). While one sentence may refer to multiple “cats” and another refers to a singular “cat”, we would want to be able to identify these sentences as similar since they are referring to the animal known as a cat. As seen in Figure 6, nouns, verbs, and adverbs are simplified to a case common form before being quantified as



```
##    title                                   chapter term
##    <chr>                                     <int> <chr>
##  1 Great Expectations                           38 brother
##  2 Great Expectations                           22 brother
##  3 Great Expectations                           23 miss
##  4 Great Expectations                           22 miss
##  5 Twenty Thousand Leagues under the Sea         8 miss
##  6 Great Expectations                           31 miss
##  7 Great Expectations                            5 sergeant
##  8 Great Expectations                           46 captain
##  9 Great Expectations                           32 captain
## 10 The War of the Worlds                        17 captain
```

Figure 5: Example of a resulting data structure from text preprocessing. Adapted from Silge & Robinson, 2017, ch. 6.2.

a frequency. This removes meaning from the individual words but helps to identify possible trends.

Three common approaches to stemming include semi-automatic lookup tables, suffix stripping, and lemmatization (Aggarwal 2018). Semi-automatic lookup tables are predefined dictionaries that identify all the possible variants of a base word (Aggarwal 2018). For example, when a token such as “eats” or “eating” are identified, the dictionary changes those words to the word “eat” (Aggarwal 2018). A down side to this method is that the work required to build such a dictionary or to modify a dictionary to handle field specific language is difficult. Suffix stripping is a set of rules that remove common word endings like “ing”, “ed”, “ly”, or “s”. While not difficult to implement, issues occur as words like “ate” will not change to “eat”, and “hoping” would have its meaning changed to “hop” (Aggarwal 2018). Lemmatization is a predefined dictionary technique like semi-automatic lookup tables, but it is more complex to find larger underlying base themes of words. For example, words like “better” and “best” may be mapped to the word “good”. Because of the use of complex human meaning to find base words, lemmatization is considered a different process from stemming by some text mining practitioners (Aggarwal 2018).

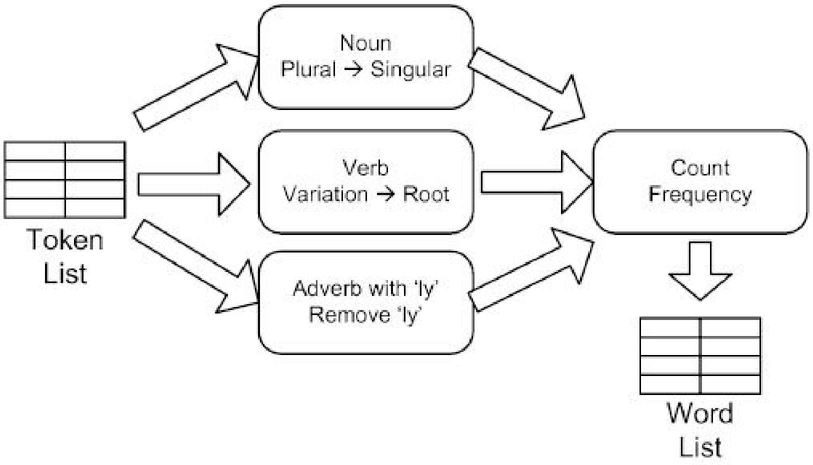


Figure 6: The process of taking a token list and changing tokens to their base form to create a word list. Adapted from Jo, 2019, p. 23.

The last step in preprocessing textual information is stop word removal. Many prepositions, conjunctions, and definite articles lose meaning relevant to answer questions when not in context of surrounding words (Jo 2019). As such, these words are removed as their high frequencies will influence many analyses. A common method to remove these words is by using a stop word list which has defined all words that will be removed. Making these lists can be difficult and using lists that others have made can result in the removal of important field-specific words. Another way to reduce the importance of stop words is to use normalization. Normalization is a process to standardize tokens, changing the weight of the token's frequency (Kumar & Paul 2016). Normalization methods, like term frequency inverse document frequency (tf-idf), will often help lower the weight of these common terms, keeping them in the analysis but with less impact on the results (Jo 2019). Despite normalization reducing the influence of stop words, it has been noticed that in some algorithms like tf-idf the influence of other errors in text such as misspellings can increase (Aggarwal 2018).

Parts of speech tagging is another part of preprocessing text used for natural language processing, as such, more about parts of speech tagging is discussed in the natural language processing section. For more information on text preprocessing (including scraping textual data and parts of speech tagging) reference the following texts:

Aggarwal 2018, p. 17:30; Jo 2019, p. 19:40; Kumar & Paul 2016, ch. 2-3; Silge & Robinson 2017, ch. 1-5; Weiss, Indurkhya, & Paul 2015, p. 13-39

3.3 Document Clustering

The goal of document clustering is to take a set of documents, often called a corpus, and place them into groups of like documents (Aggarwal 2018). Figure 7 displays this overall clustering process where a collection of documents goes through a document organization algorithm, resulting in documents being sorted into five groups. Clustering of documents is similar to document classification where documents are placed into groups, with the distinction that clustering is an unsupervised process where groups are not pre-defined.

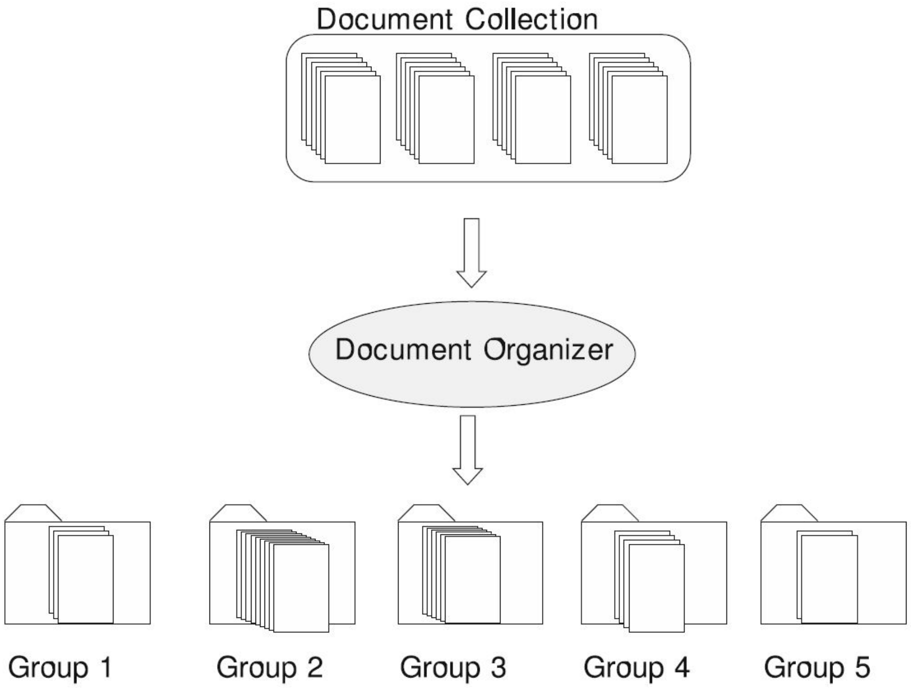


Figure 7: Process of clustering a set of documents into groups. Adapted from Weiss, Indurkhya, & Paul, 2015, p. 8.

Document clustering can use traditional clustering techniques based on word frequencies to compare similarity of documents. This makes document clustering an expansive topic commonly covered in many books and articles. Most of the sources provided below go over a wide range of clustering algorithms, and books like Jo (2019) go over the broad types of clustering. This section will not cover different clustering algorithms, but will discuss unique clustering characteristics of textual data.

There are three unique characteristics of textual data that make clustering more complicated (Aggarwal & Zhai 2012). For one, textual data tends to be very sparse with millions of words, but only a handful of words in a document can be similar to another document. This problem proliferates when working with low word count documents such as social media posts (Aggarwal & Zhai 2012). Another issue is that documents can have widely different word counts (Aggarwal & Zhai 2012). When a common word appears in a short document, it can make up a larger proportion of that document. If that word appeared the same number of times in a very long document, it may not seem very abundant. As such, it can be important to normalize the representation of documents before clustering (Aggarwal & Zhai 2012).The last issue is that words tend to be highly correlated with one another despite the large number of words used in a set of documents. This makes it important that clustering techniques appropriately account for this correlation (Aggarwal & Zhai 2012). For more information about document clustering and different clustering algorithms see the following sources:

Aggarwal 2018, p. 73-111; Aggarwal & Zhai 2012, p. 77-121; Berry et al. 2008, p. 3-108; Jo 2019, p. 183-267; Kumar & Paul 2016, ch. 5; Weiss, Indurkhya, & Paul 2015, p. 97-118

3.4 Document Classification

Document classification is the process of assigning documents to one or more predefined groups (Jo 2019). It is a supervised technique, meaning a model has to be trained on already classified documents before it can start classifying new documents (Aggarwal 2018). While similar to document clustering, a key distinction is that, unlike clustering, the groups to place documents must already be known and some documents must have already been classified (Aggarwal 2018). A framework for document classification can be seen in Figure 8 where a document can be placed in one of three groups depending on which criteria the document meets. Document classification has many practical uses from identifying spam email to classifying news articles to topics for a media organizations website.

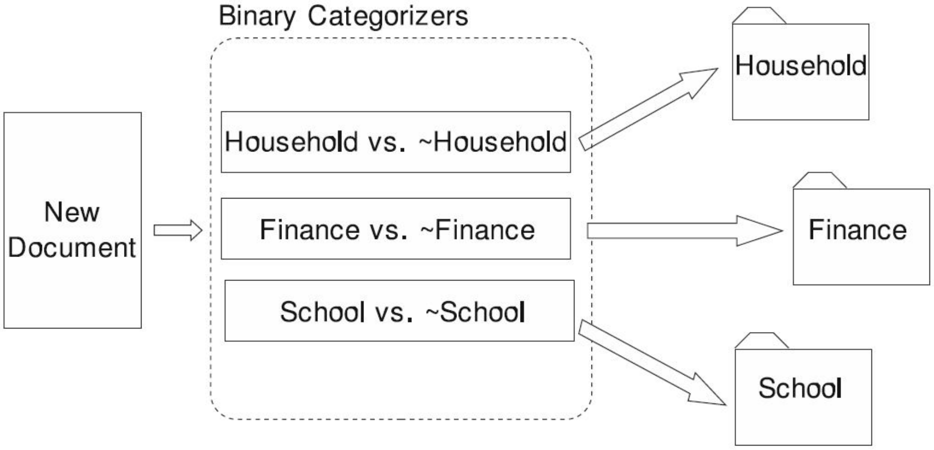


Figure 8: Using binary classifiers to sort new documents into groups. Adapted from Weiss, Indurkhya, & Paul, 2015, p. 6.

An important phase of document classification is the testing and training phase. The training phase looks at qualities of each document to find commonalities among documents classified to a group (Weiss, Indurkhya, & Paul 2015). Words are often the qualities often used to classify text documents. The training phase involves selection of the model as well as decisions about which qualities to include for the classification process (Aggarwal 2018).

What can be difficult when training a document classification model is deciding how many pre-classified documents are enough to train an accurate model. It is not enough to assemble a large collection of documents; one also needs to consider the distribution of these documents across the categories. Some topics may be rare and have very few documents. If there is an obvious difference in this rare group compared to the other groups, a few document may be enough for accurate classification. However if there are no clear defining features of a rare group, more documents need to be collected (Weiss, Indurkhya, & Paul 2015). Another complication is that a representative set of documents to train a model on can change over time as classification topics emerge, disappear, split, or combine resulting in the need for more training articles to keep the classification model up-to-date (Weiss, Indurkhya, & Paul 2015).

After generating a model, it is tested to look at classification accuracy. While high accuracy is desired, it is not everything. A model can be trained to have 100% accuracy on a specific test set of documents, but this may mean the model has been overfit. An overfit model is a model that has been fine tuned for a very specific set of characteristics and will perform poorly on new sets of documents that differ from the test documents (Aggarwal 2018). It is often desirable to have a more generizable classification model. To learn more about document classification and different modeling techniques see the following texts:

Aggarwal 2018, p. 113-258; Aggarwal & Zhai 2012, p. 163-213; Jo 2019, p. 79-156; Kumar & Paul 2016, ch. 6; Weiss, Indurkhya, & Paul 2015, p. 41-79

3.5 Information Retreival

The goal of information retrieval is to provide a query, usually in the form of a set of key words, to identify a set of relevant documents (Aggarwal 2018). Different ways of defining queries and relevant documents can result in varying information retrieval results. For example, a Boolean retrieval can identify which documents contain the query words “Happy” and the word “Birthday” and which documents do not. The documents that do not contain both of those search words will be left alone while the documents that do contain “Happy” and “Birthday” will be brought to the user.

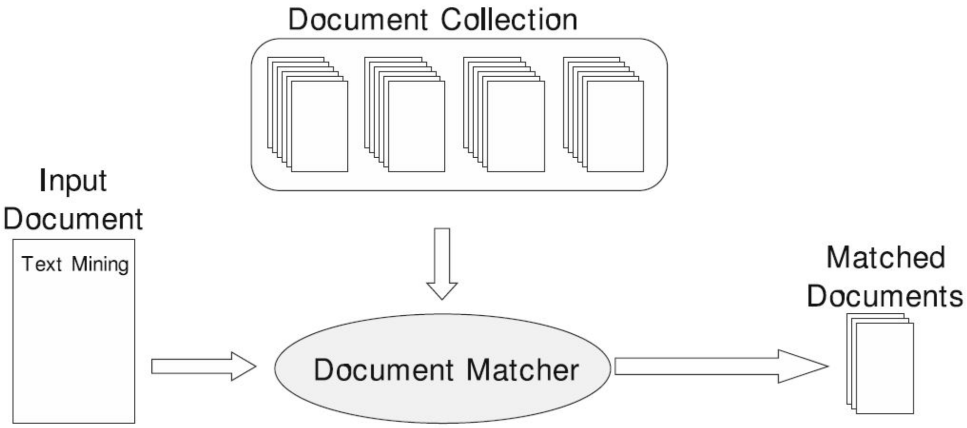


Figure 9: Process of selecting relevant documents from a collection given user text input. Adapted from Weiss, Indurkhya, & Paul, 2015, p. 7.

In large document databases, a query could bring thousands of documents to the user. Because of this problem scoring techniques are used to rank the relevance and quality of the extracted documents (Aggarwal 2018). The documents with the top scores are shown to the individual who provided the query, and if asked for, the next highest scored documents are also provided.

Traditionally, information retrieval techniques only take into consideration the content in a document and do not require a training set of documents. This has been changing with the vast quantities of text documents accessible through the web, resulting in an increased use of supervised information retrieval algorithms and other document information such as hyperlinks to identify relevant information for a query. For more information on information retrieval you can reference these pages of text:

Aggarwal 2018, p. 259-360; Berry et al. 2008, p. 109-147; Kumar & Paul 2016, ch. 7; Weiss, Indurkhya, & Paul 2015, p. 81-95

3.6 Information Extraction

Information extraction can be seen as taking a bunch of documents full of unstructured text and filling out a template (such as a blank spreadsheet or database, Figure 10) with desired information (Weiss, Indurkhya, & Paul 2015). This is not an easy process, for example, if we wanted to extract the names of rock stars to fill in a music data base, we would first need to be able to identify individual names. Once a name such as John Smith has been identified as a rock star, the next difficulty lies in checking which musical group John Smith belongs to and to whom he has connections (e.g. other band members, spouse, etc.)!

These ideas break entity extraction into two main concepts: 1) named entity recognition and 2) relationship extraction (Aggarwal 2018). Named entity recognition is the process of identifying words as concepts, such as a person, organization, or location. Relationship extraction takes these named

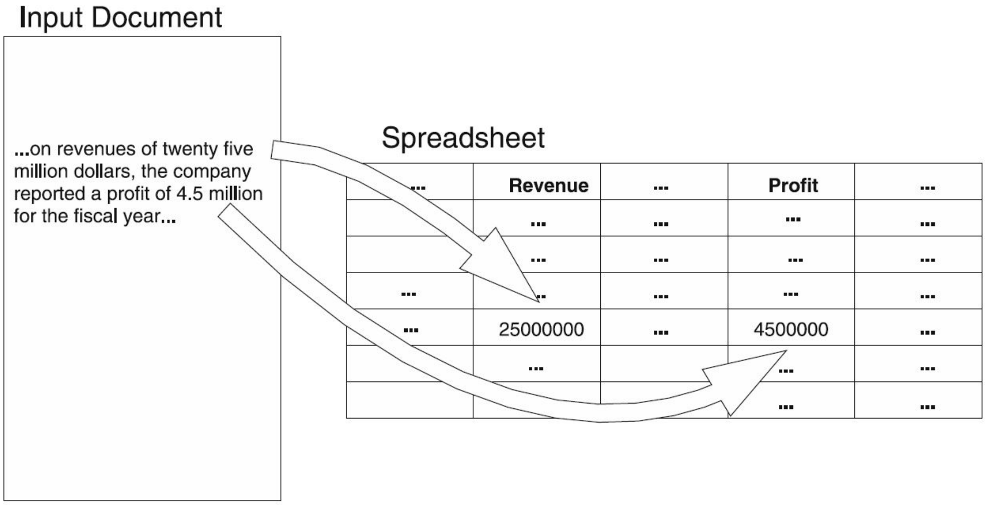

Input Document
...on revenues of twenty five million dollars, the company reported a profit of 4.5 million for the fiscal year...
Spreadsheet
Revenue
Profit
25000000
4500000

Figure 10: Retreiving specific values from a document to place into a spreadsheet. Adapted from Weiss, Indurkhya, & Paul, 2015, p. 9.

entities and tries to identify the relationships among them (Aggarwal 2018). A few examples of this is that “John Smith” has been identified as a person and “The Smiths” has been identified as a band, so a relationship to identify is that John Smith is a member of The Smiths.

Figure 11 displays a visual representation of how documents are used to train different algorithms to identify different entities and their relationships which can then be applied to future sets of documents to extract information. Using a supervised approach to information extraction is common however there are unsupervised information extraction techniques as well (Aggarwal & Zhai 2012). These unsupervised techniques are often called open information extraction (Aggarwal 2018).

An issue that commonly pops up in entity extraction is the concept of co-reference resolution (Aggarwal 2018). Entities can have multiple ways of

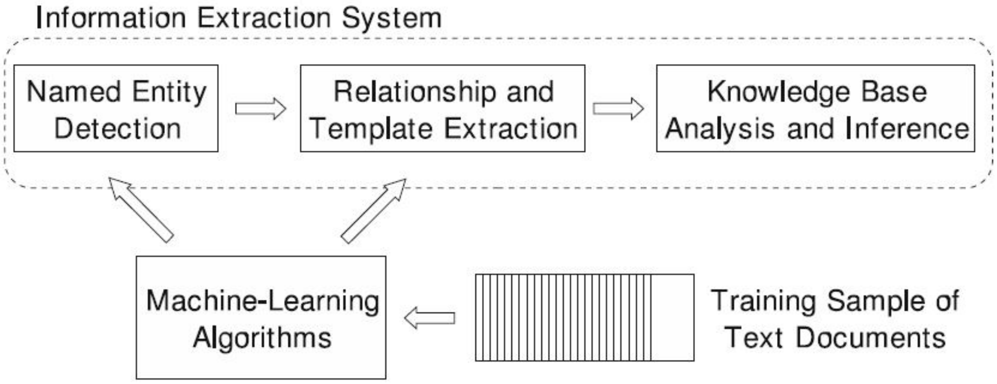

Information Extraction System
Named Entity Detection
Relationship and Template Extraction
Knowledge Base Analysis and Inference
Machine-Learning Algorithms
Training Sample of Text Documents

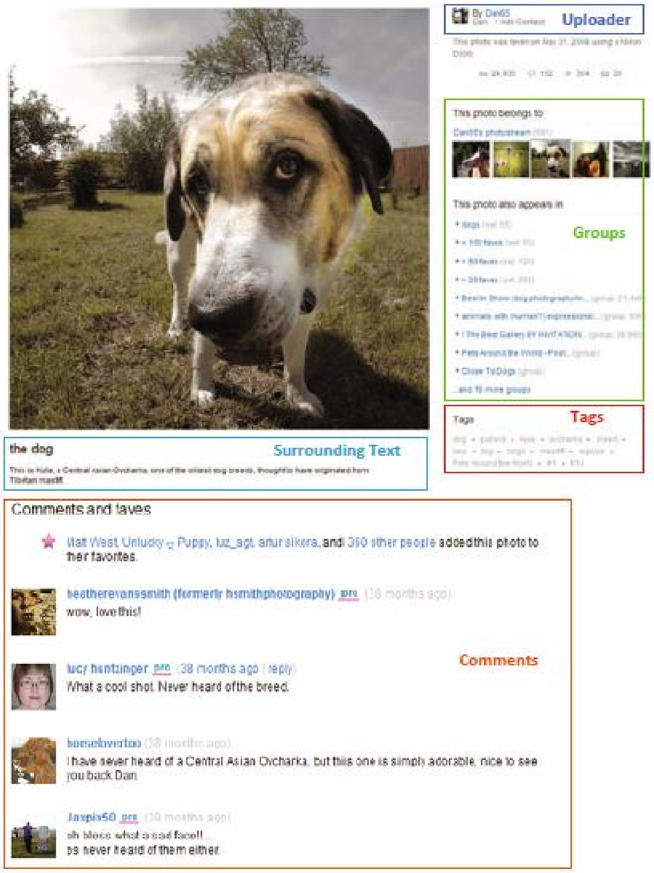
Figure 11: Process of training a model for extracting information from documents. Adapted from Weiss, Indurkhya, & Paul, 2015, p. 121.

being referenced. The rock star John Smith may sometimes be called Johny, or a sentence may use a pronoun to reference John Smith even though the pronoun itself is not a named entity. For more information on information extraction you can reference these pages of text.

Aggarwal 2018, p. 381-434; Aggarwal & Zhai 2012, p. 11-35; Weiss, Indurkhya, & Paul 2015, p. 119-145

3.7 Web Mining

Textual data are often considered messy compared to traditional data because they are not in a clean spreadsheet, but they are usually homogeneous (i.e. the documents are all written in a common language) (Weiss, Indurkhya, & Paul 2015). This is not the case when it comes to working with internet related documents. Websites and social media contain hyper-links connecting text between users to reference other web sources with related content (Aggarwal 2018). It is also common to see documents in multiple languages or to have videos and images included with captions or comments. Analyzing complex hybrid data with these components is the reason for work in the practice area of web mining.



the dog

This is Kula, a Central Asian Ovcharka, one of the oldest dog breeds, thought to have originated from Tibetan mastiff.

Comments and faves

Matt West, Unlucky Puppy, luz_agt, artur sikora, and 350 other people added this photo to their favorites.

heatherevanssmith (formerly hsmithphotography) pro (38 months ago)
wow, love this!

lucy hentzinger pro (38 months ago | reply)
What a cool shot. Never heard of the breed.

horseloverloo (38 months ago)
I have never heard of a Central Asian Ovcharka, but this one is simply adorable. nice to see you back Dan.

Jaxpix50 pro (38 months ago)
oh bless what a sad face!!...
bs never heard of them either.

Figure 12: Example of a webpage that contains different types of information, some of which will be on differing feature spaces. Adapted from Aggarwal & Zhai, 2012, p. 363.

In general people want to answer the same types of questions when working with web/heterogeneous data that they would with more traditional homogeneous text data. These problems involve classification of documents into known groups, clustering documents into similar group, or identifying latent or unknown document qualities with topic modeling. Most of these techniques work best when the object being classified or clustered has all the relevant input information in an single data frame or feature space but that is not always the case (Aggarwal 2018). Information from web data is often stored in different feature spaces. For example, multiple feature spaces could include the relationship between a set of documents and the words used are represented in a document-word matrix (one feature space) and the hyperlinks between the web documents which may be represented as a document-document matrix (another feature space). The challenge is to relate information from many different feature spaces that are difficult or impossible to meld into one feature space. For example, to classify the picture in Figure 12, information from the Uploader, Groups, Tags, Surrounding Text, and Comments would need a cohesive feature space or a network of connections between features to classify the image from the site.

Several common ways to tackle the issue of differing feature spaces include: shared matrix factorization, joint probabilistic modeling, and relationship graphs. Relationship graphs create connections between the documents through similar features such as words, image tags, and user ids (Aggarwal 2018). Once a network is created among the documents network algorithms can be used be used to analyse the data. A simple example of a network of hyper-links between four documents and six words from the documents can be seen in Figure 13. However, as shown in Figure 14, the network complexity increases with more documents and the addition of features such as words from another language. Such complexity is a main concern with this approach because networks can

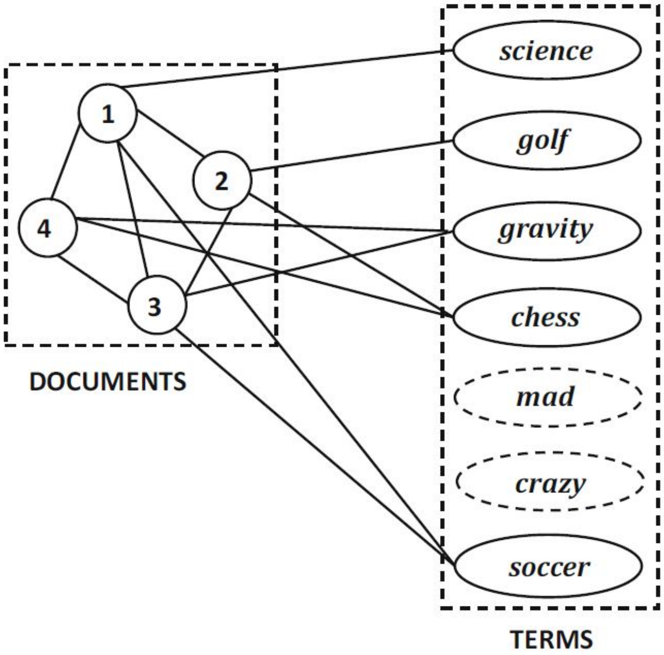


Figure 13: Relationship network that shows connections among documents from hyperlinks, as well as, relationships among words used and documents. Adapted from Aggarwal, 2018, p. 255.

be massive with many documents and features.

Joint probabilistic modeling and shared matrix factorization are used to address the complexity from large relationship graphs. Joint probabilistic modeling puts the different feature spaces on the same scale by modeling each feature space as a probability distribution (Aggarwal 2018). Once all the information is on the same scale, a variety of questions can be answered by looking at the joint distribution between these various probability distributions. Shared matrix factorization simplifies the relationship graphs by reducing the dimensionality of the problem using this approach, the relations from factor spaces are modeled as latent representations, known as shared latent

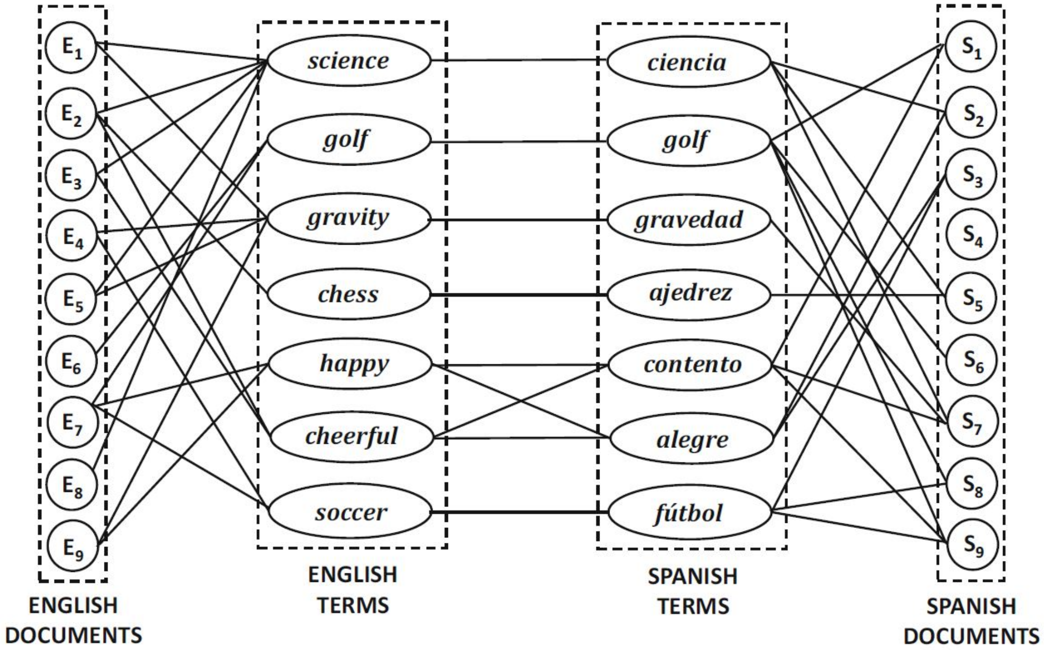


Figure 14: Relationship network connetcting a set of Engligh documents to Spanish documents through the relationship between words from a spanish and english term list. Adapted from Aggarwal, 2018, p. 256.

factors, with a factorization graph. For more information on how to work with heterogeneous data associated with the web see the following resources:

Aggarwal 2018, p. 235-258; Aggarwal & Zhai 2012, p. 361-408; Weiss, Indurkhya, & Paul 2015

3.8 Natural Language Processing

Text mining applications usually take a bag-of-words approach that looks at the occurrence of individual words throughout documents, often saving time in analysis by ignoring linguistic aspects of the text. The aspects of information that are missing from a bag-of-words approach include sentence structure, word order, and context (Kumar & Paul 2016). Natural language processing works to add these pieces of natural human linguistics back into the data at the cost of increased difficulty of data preparation and analysis.

The primary way natural language processing adds linguistics is through parts of speech tagging. Parts of speech tagging works by identifying and labeling words with linguistic properties such as being a noun, verb, or adjective (Kumar & Paul 2016). This information can build on the linguistic structure by identifying how these words form sentences through noun and verb phrases. For example, in Figure 15 words from the sentence “The rabbit ate the carrot” are tagged with parts of speech labels, with the sentence structure appropriately labeled. A verb such as “eat” is labeled with VB and a past tense verb such as “ate” is labeled with VBD. Nouns in the sentence are tagged by NN and common articles are labeled with DT for determiners (Aggarwal 2018). Each noun phrase is labeled with NP, and each verb phrase is labeled with VP. All of this information can be kept and represented in a tree-structure. It is very time intensive to tag parts of speech by hand, so it is common to tag parts of speech using pre-built algorithms that have been trained and modified extensively.

A common use of natural language processing is to aid in clustering and classification, but it is also used in topics such as information extraction, document summarization, opinion mining, text sequence modeling, and event extraction as these methods work to extract human and semantic meaning from text. However, not every algorithm for these topics requires the text to be tagged with parts of speech (Aggarwal 2018). The framework of conceptualizing text mining presented in this paper (see Figure 2) allows for a lot of overlap

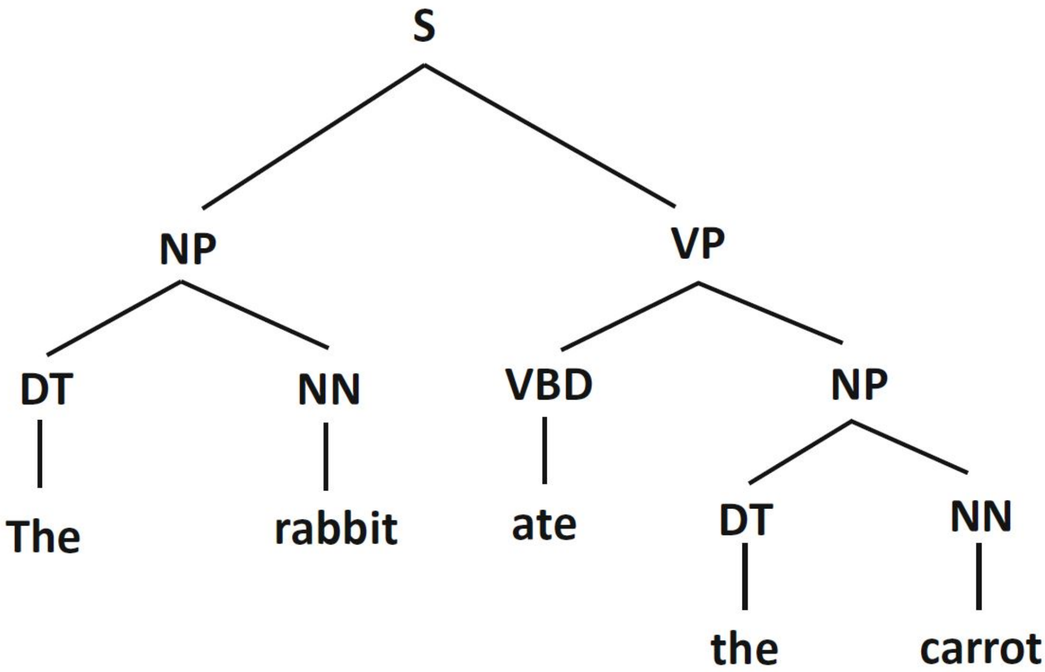


Figure 15: Structure of a sentence represented through a parse tree. Adapted from Aggarwal, 2018, p. 385.

between natural language processing and the other seven practice areas.

Because of the difference in the text preprocessing phase from parts of speech tagging and the fact that there are many frameworks that conceptualize text mining as groups with no overlap, it is common for individuals to consider natural language processing as a separate field from text mining. As such, not a lot of the literature read for this paper discussed algorithms using parts of speech tagged data. For more information on the parts of speech tagging and aspects of natural language processing see the following texts

Aggarwal 2018, p. 384-385; Kumar & Paul 2016, ch. 3

3.9 Concept Extraction

Concept extraction focuses on identifying and extracting human meaning from text. Concept extraction can be a complex process as computer algorithms have a hard time understanding the semantics of the words being used while a human reader has difficulty processing large quantities of documents (Miner et al. 2012). The appropriate combination of the two can result in effective ways to extract meaning from text.

Some common techniques in the area of concept extraction are sentiment analysis, document summarization, topic modeling, text sequence modeling, and synonym identification. Sentiment analysis is used to identify attitudes and feelings expressed in a document or set of documents (Aggarwal 2018). This can range from identifying positive or negative qualities from a product review to exploring the appeal of a political candidate from questionnaires. A way this can be done is by tagging words with specific emotions or concepts, such as “dislike” with negative emotion or “enjoy” with positive, and using these tagged words for classification, clustering, or any other data analysis (Silge & Robinson 2017). The hard part is tagging different words with human emotions. One simple way is to use a list of words that have already been associated with an emotion, but the usefulness of this approach will depend on the quality and context under which the sentiment word list was created. The accuracy of analyzing sentiment through this method can also be improved if word order is maintained so that way a word string such as “not great” will add negative sentiment to a document instead of positive.

The other techniques have different ways of capturing human meaning.

Document summarization aims to take a document (or multiple documents) and extract its overall meaning into a shortened text. In contrast, topic modeling is used to identify hidden, or latent, themes from a group of documents.

Many concept extraction questions and analyses can be addressed using a a natural language processing framework, but concept extraction and natural language processing are still treated as two different practice areas? Natural language processing adds structure to text by adding parts of speech and sentence structure to the quantified text. This added information on linguistic structure is helpful in identifying human meaning in text documents. However, it is not necessary to use this added structure to extract meaning, as it can simply be used to improve algorithms for classification accuracy. Concept extraction is the idea of extracting meaning whether it uses the added structural benefit of natural language processing or not. For more information on different methods for extracting human meaning from text documents see the following texts:

Aggarwal 2018, p. 31-71, p. 305-359, p. 413-452; Aggarwal & Zhai 2012, p. 43-75; p. 415-463; Kumar & Paul 2016, ch. 5; Silge & Robinson 2017, ch. 2, ch. 6

4 Case Study

This case study goes over several aspects of text mining by implementing them on a set of educational data. These aspects include preprocessing of textual data, visualization of word frequencies, and topic modeling using Latent

Dirichlet Allocation. This introduction to the case study gives background context to the educational data used.

The educational data used come from the IMMERSION project that provided professional development to elementary and middle school teachers about implementing mathematical modeling lessons in their classrooms (Burroughs & Fulton 2018). Mathematical modeling is a process in which a real-world problem or scenario is translated by a modeler into a mathematical problem. The modeler finds solutions to the mathematical problem and then interprets them within the context of the real-world problem. This is an ongoing process where the modeler keeps translating the problem between the two worlds, adjusting to improve the solution and its back-translation to the real world (Hirsch & McDuffie 2016).

During the IMMERSION project, the teachers completed a summer course and then developed and implemented a mathematical modeling lesson during the subsequent fall semester. The course helped to inform elementary school educators on what mathematical modeling is; the benefits of implementing mathematical modeling exercises in K-5 classrooms; and how to implement such activities (Burroughs & Fulton 2018). IMMERSION defined mathematical modeling as:

> "Based on student engagement with authentic problems that matter to the classroom community, use mathematical tools to propose solutions to those problems, and involve students as the ones who pose the mathematical question to be asked and addressed" (Burroughs & Fulton 2018, p. 4).

The IMMERSION program was offered to educators for three summers (2015 to 2017) at three different sites (Burroughs & Fulton 2018). Survey data were collected on the educators at the start of the professional development course, at the end of the course, and after using a mathematical modeling lesson in their classroom. At the beginning of the survey, the educators were asked three open-ended questions:

1. What does the phrase "mathematical modeling" mean to you, in the context of your work as an elementary grades teacher? (Please answer in a few sentences or a paragraph.)

2. Please describe a typical day in your mathematics classroom. Excluding the problem you tried in your Teacher Study Group or Lesson Study Group, what kinds of problems or activities have your students worked on? Provide an example from the past month if you can.

3. Have you engaged your students in the practice of mathematical modeling in the past? If yes, briefly describe how. If not, describe things your students have done on the modeling spectrum.

This case study will focus on analysis of responses from the question: "What does the phrase 'mathematical modeling' mean to you, in the context of your work as an elementary grades teacher?"

4.1 Case Study - Preprocessing the Textual Data

This section describes how the textual education data were preprocessed using three different data manipulation processes. For the first process, the streams of text were broken down into lists of words that were used to obtain word counts for data visualization. The second process transformed the text into sets of words to visualize the relationship between a word and the words that came before and after it. The last process manipulated the text information to create a matrix identifying the frequency of words used in each participant's question response. This matrix was necessary for fitting the topic model used in the last section of the case study.

4.1.1 Breaking Streams of Text into Words

The surveys were administered online, allowing the original data to be extracted straight into an excel file. With the survey data already digitized, there was no concern of transcription errors. Despite the data already being stored in an excel file, the data still needed to be manipulated into a format in which the text could be quantified. For this case study a subset of the original dataset was used that included participants from 2015 and 2016 and focused on the question:

1. What does the phrase "mathematical modeling" mean to you, in the context of your work as an elementary grades teacher?

To give a more holistic view of the data manipulation process, a second open-ended survey question was included as it is common that surveys will have more than one open-ended response of interest.

The initial dataset stored each participant's survey responses for a given test in a single row (Figure 16). The variables associated with each row include:

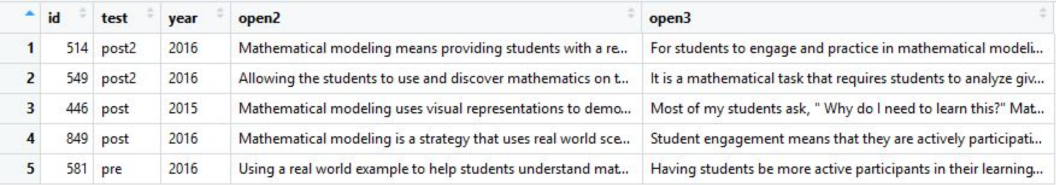
-id - Numeric code unique to each individual who participated in the professional development seminar

-test - Identifier denoting if the responses are from the test given before the professional development session (pre), after the professional development session (post), or during the fall semester after using a mathematical modeling framework lesson in the classroom (post2)

-year - The year in which the test was taken (2015 or 2016)

-open2 - The open-ended response to the question: "What does the phrase 'mathematical modeling' mean to you, in the context of your work as an elementary grades teacher?"

-open3 - The open-ended response to the question: "Describe what it means for students to engage in the practice of mathematical modeling."



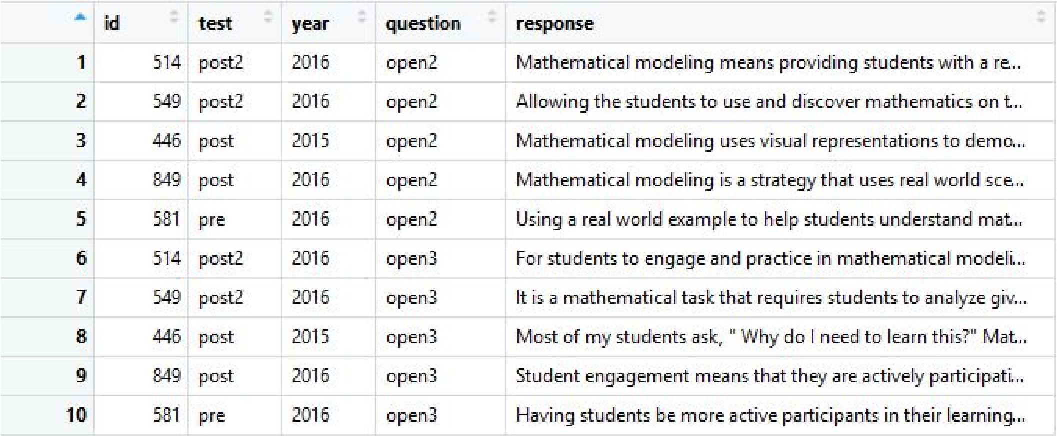
	id	test	year	open2	open3
1	514	post2	2016	Mathematical modeling means providing students with a re...	For students to engage and practice in mathematical modeli...
2	549	post2	2016	Allowing the students to use and discover mathematics on t...	It is a mathematical task that requires students to analyze giv...
3	446	post	2015	Mathematical modeling uses visual representations to demo...	Most of my students ask, " Why do I need to learn this?" Mat...
4	849	post	2016	Mathematical modeling is a strategy that uses real world sce...	Student engagement means that they are actively participati...
5	581	pre	2016	Using a real world example to help students understand mat...	Having students be more active participants in their learning...

Figure 16: Structure of the initial survey data.

A transformation to the data was implemented to expand the data into a long format in which each row contained a participant's textual response to one of the open-ended questions on a given test (Figure 17). The resulting data still have an id, test, and year identifier but include two new columns. These new columns are:

-question - The open-ended response question (open2 or open3)

-response - The textual response

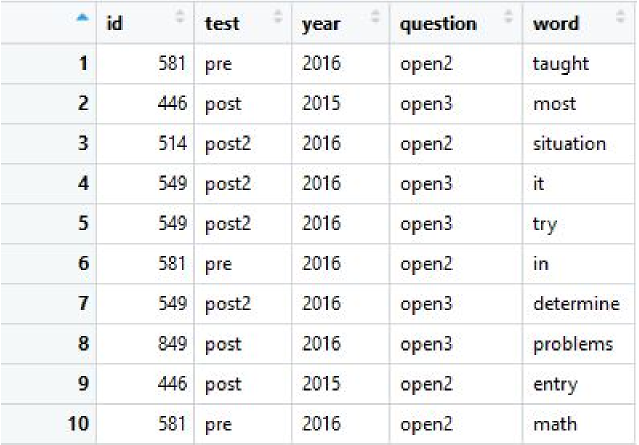


	id	test	year	question	response
1	514	post2	2016	open2	Mathematical modeling means providing students with a re...
2	549	post2	2016	open2	Allowing the students to use and discover mathematics on t...
3	446	post	2015	open2	Mathematical modeling uses visual representations to demo...
4	849	post	2016	open2	Mathematical modeling is a strategy that uses real world sce...
5	581	pre	2016	open2	Using a real world example to help students understand mat...
6	514	post2	2016	open3	For students to engage and practice in mathematical modeli...
7	549	post2	2016	open3	It is a mathematical task that requires students to analyze giv...
8	446	post	2015	open3	Most of my students ask, " Why do I need to learn this?" Mat...
9	849	post	2016	open3	Student engagement means that they are actively participati...
10	581	pre	2016	open3	Having students be more active participants in their learning...

Figure 17: Long format structure of the original data where each row is an individual's response to a single survey question.

Each text response is unique, so the next step was to define what will be considered a token, the observed unit being analyzed. If each response was considered a token, it is unlikely that any two responses are exactly the same, leading to a relatively uneventful data exploration. Sentences, letters, and paragraphs could all have been considered a token, but for this case study each word was treated as a token. It is of interest how educators' language changed across the professional development and this may be most noticeable via changes in words used to describe mathematical modeling.

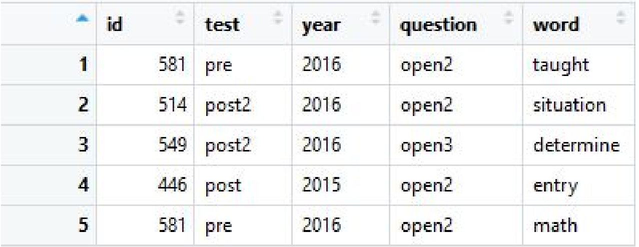
The package `tidytext` (Silge & Robinson 2017) was used to parse out each word of the text responses. This simplified the process, but it is important to consider how the parsing technique may influence the results. Words were identified with spaces being the deliminator and certain types of punctuation such as periods and exclamation marks were removed if they were at the start or end of a parsed word. Also all capital characters were replaced with a lowercase character. The resulting token list can be seen in Figure 18. There are issues and biases that can be imposed through any parsing method and this one is no exception. However, in certain explorations of text data, keeping the order of the parsed words or tagging the words with parts of speech can add valuable information. In this case information on word and sentence order was not maintained in the parsing process as it was not going to be considered in future analyses.



	id	test	year	question	word
1	581	pre	2016	open2	taught
2	446	post	2015	open3	most
3	514	post2	2016	open2	situation
4	549	post2	2016	open3	it
5	549	post2	2016	open3	try
6	581	pre	2016	open2	in
7	549	post2	2016	open3	determine
8	849	post	2016	open3	problems
9	446	post	2015	open2	entry
10	581	pre	2016	open2	math

Figure 18: Example of ten tokens generated from parsing unstructured text.

In the final step to prepare these data, frequent fill words that carry little meaning, called stop words, were removed. Although it would have been ideal to build a stop word list specific to this study, a prebuilt stop word list in the `tidytext` (Silge & Robinson 2017) package was used to identify and remove stop words. Figure 19 displays what tokens remain after the removal of stop words from the token list in Figure 18. There are other methods to minimize the influence of frequent but low meaning words on analysis of text, but use of a stop word list was used for simplicity.



	id	test	year	question	word
1	581	pre	2016	open2	taught
2	514	post2	2016	open2	situation
3	549	post2	2016	open3	determine
4	446	post	2015	open2	entry
5	581	pre	2016	open2	math

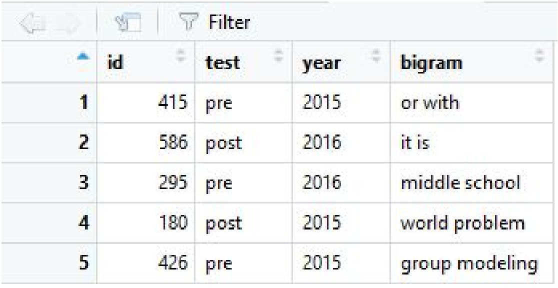
Figure 19: Example of remaining tokens after stop word removal.

4.1.2 Breaking Streams of Text into Bigrams

In the second data manipulation process, another format was created by using a different token instead of words. The token form that was used is known as a n-gram. N-grams break textual units of observation into strings of n words. This allows for examination of word relations through word order that may have alternative meanings when not observed in conjunction with other words. For example, the word “not” and “bad” have a different connotation than “not bad”.

For this situation the n-grams being looked at are bigrams, word strings containing only two words. In the text preprocessing steps, selection of bigram tokens came after turning the data into a long format. As can be seen from the subset of the bigrams found in Figure 20, the word “middle” has a different meaning once we know that it was used in conjunction with the word “school”.

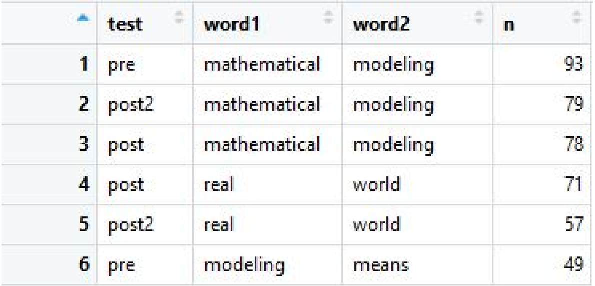
Once the bigrams were extracted the words were split into two variables, maintaining the order of the word usage. Splitting the bigram into the individual words allows for the use of stop word lists for stop word removal. Any bigram that contained at least one stop word was removed. While this does remove combinations of words with low meaning such as “or with”, there may be some



	id	test	year	bigram
1	415	pre	2015	or with
2	586	post	2016	it is
3	295	pre	2016	middle school
4	180	post	2015	world problem
5	426	pre	2015	group modeling

Figure 20: Token list generated after parsing out bigrams.

bigram combinations with meaning that were removed because they contained stop words.



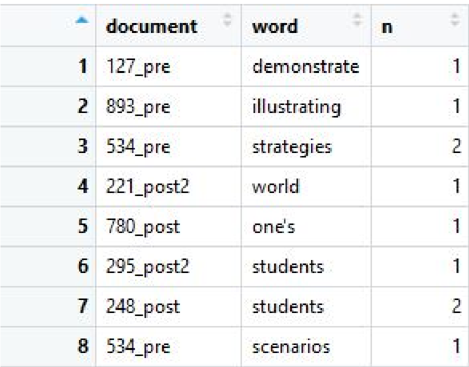
	test	word1	word2	n
1	pre	mathematical	modeling	93
2	post2	mathematical	modeling	79
3	post	mathematical	modeling	78
4	post	real	world	71
5	post2	real	world	57
6	pre	modeling	means	49

Figure 21: Summarization of bigrams with stop words removed.

Once stop words were removed, the data were summarized to look at the frequency of bigrams for each of the three surveys (Figure 21). This allows for comparison of frequently used bigrams between the different surveys. In particular this summarization was used to create a visualization of a bigram network, seen in the data visualization section of the case study.

4.1.3 Turning a Word List into a Document-Term Matrix

The third text structure used in this case study is known as a document-term matrix. A document-term matrix looks at all the words used in a set of texts and for each piece of text, identifies which words were used and their quantity of use; zeros are used for the words that do not occur in the document. The document term matrix was processed from the long format list of terms with stop words removed and thus has the same biases discussed earlier in section 4.1.1.

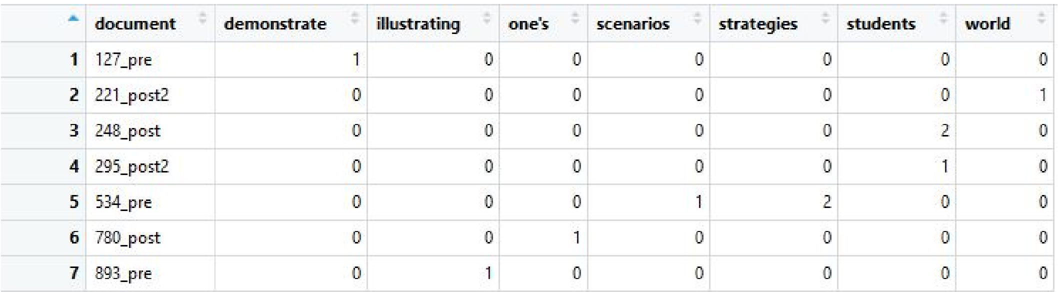


	document	word	n
1	127_pre	demonstrate	1
2	893_pre	illustrating	1
3	534_pre	strategies	2
4	221_post2	world	1
5	780_post	one's	1
6	295_post2	students	1
7	248_post	students	2
8	534_pre	scenarios	1

Figure 22: List of word frequencies per document.

The first step in this text manipulation was to identify the individual document. For these data, an individual's response to one survey (pre, post, or post2) is considered a document. Once an unique identifier for each document was created, word frequencies were generated for each word and document combination. This information was stored in a list as seen in Figure 22 and contains all the information needed to spread the data into matrix format seen in Figure 23.

Having document-term relations in a matrix is needed in some algorithms, such as Latent Dirichlet Allocation, but it is space inefficient. There can be thousands of documents with hundreds of thousands of words used. Many of these words are not used in a majority of documents, leaving a lot of zero word



	document	demonstrate	illustrating	one's	scenarios	strategies	students	world
1	127_pre	1	0	0	0	0	0	0
2	221_post2	0	0	0	0	0	0	1
3	248_post	0	0	0	0	0	2	0
4	295_post2	0	0	0	0	0	1	0
5	534_pre	0	0	0	1	2	0	0
6	780_post	0	0	1	0	0	0	0
7	893_pre	0	1	0	0	0	0	0

Figure 23: Example document term matrix.

frequencies as place holders. Having so many zero word frequencies makes the matrix sparse. As seen from a subset of 7 documents and 7 words, shown in Figure 23, only 8 of the 49 cells contain nonzero information. This same information can be condensed by being stored in a document-term list of 24 elements.

These are just three ways in which textual data can be preprocessed. There are many other ways to process textual data depending on the question being asked and the data provided. The next sections of this case study will conduct analyses that use these three different data structures.

4.2 Case Study - Data Visualization

This case study focused on exploring teachers' responses to the question, "What does the phrase 'mathematical modeling' mean to you, in the context of your work as an elementary grades teacher?" Two different data visualizations were made, one comparing word frequencies between survey responses before and after the professional development seminar and the second exploring a network of bigram relations.

The visual exploration of these data began with plots to compare the language educators used before (pre-survey) compared to after (post) the professional development seminar. The hypothesis was that after a professional development seminar, certain words defining mathematical modeling would change and/or be seen more consistently. In the follow-up survey (post2) it would be ideal to see a similar distribution of words to the post survey if there was retention of the material from the professional development seminar.

A scatterplot was made that compared the frequency of word usage in the pre-professional development survey to that of both the surveys given after the seminar (Figure 24). The proportion of use for a word written on the pre-workshop survey is given along the y-axis, and its proportions of use on the post and post2 professional development surveys are given on the x-axis. The axis is scaled logarithmically, because many words were used infrequently but a few such as “math” and “modeling” were used often. The dotted lines represent the points where word frequencies are the same between two surveys. There was a lot of overlap in word frequencies due to many low word count words seen one to five times, so points were jittered and point opacity was reduced to better visualize these low frequency words.

As shown in Figure 24, several words stuck out and are highlighted in blue. Words above the line such as “concrete” and “models” were frequently used before the professional development seminar but not as frequently after the workshop. Words below the line, such as “question”, “answer”, and “solution”, were more frequently seen in surveys after the developmental seminar then they were before. While the use of “real” and “world” were common before the

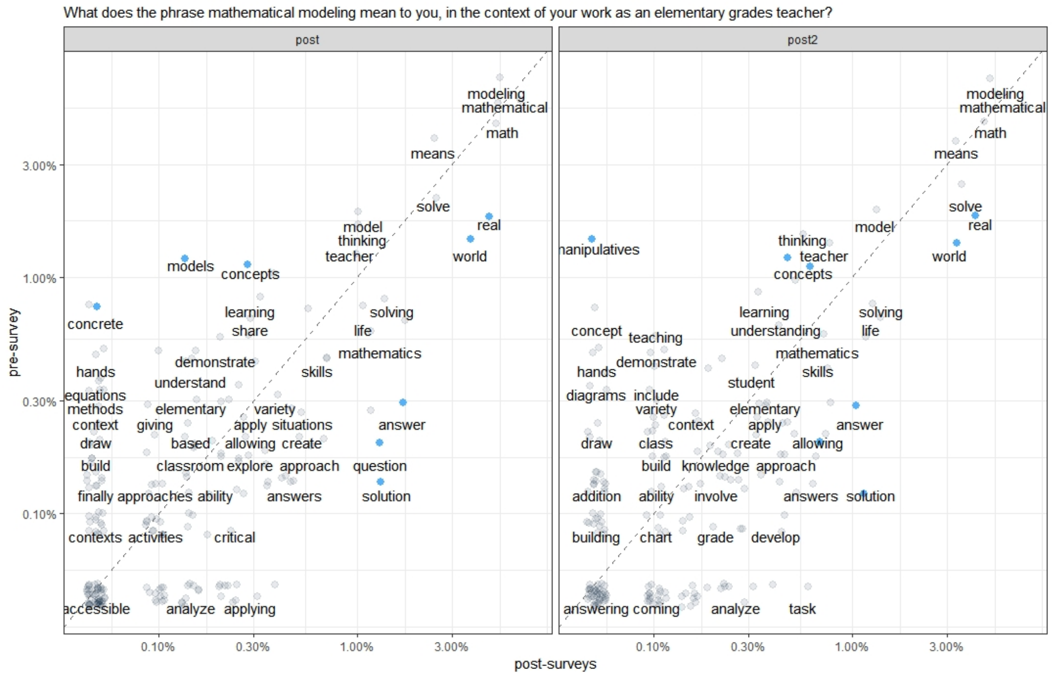


Figure 24: Scatterplot of word frequencies from the reponse to the question "What does the phrase 'mathematical modeling' mean to you, in the context of your work as an elementary grades teacher?" Word frequencies are presented on the x and y axis on a logarithmic scale. The dotted line represents when word frequencies are the same between two surveys.

workshop, they were more frequently used after it. Words from the prompt like “mathematical”, “modeling”, and “means” were used frequently in all of the surveys possibly due to reiterations of the prompt.

Some of the limitations of this text summarization are also apparent. Words such as “concrete” and “manipulatives” were used frequently in the pre-survey, but were used infrequently, if at all, in the post survey, so they do not show on the plot. Also, some words, such as “question” and “model”, may be under represented in their frequency as their plural forms, “questions” and “models”, are also seen frequently. In addition, these words (e.g. “question” or "model") question or model can be referred to as a noun or a verb, which have different meanings and can create problems that will perpetuate in any further data exploration or analysis. This visualization also does not get at how an individual's language changed. It is possible that some individuals had lengthy responses and used particular language that could be misleading when certain words are highlighted in the aggregate.

Another issue is that this visualization does not take into consideration the order and context of the words used. Based on Figure 24, it appears words such as "mathematical", "modeling", "means", "real", and "world" are possibly strings of words that are commonly used together. Understanding words that are commonly used together would give a better understanding of the differences and similarities of language before and after the professional development workshop.

To visualize the relation among words used in conjunction with each other, the data were parsed into bigrams, strings of two words such as "mathematical modeling" or "modeling means". The frequency of bigrams for each survey was displayed as a network where nodes that represent different words are connected by lines (also known as edges) if they were observed being used in order. The direction of the word order is represented by an arrow and more frequent bigrams are represented with a darker line. Only bigrams that were seen five or more times were plotted.

Looking at the bigram network for the survey taken before the professional development workshop (Figure 25) gives some insight on what was observed in the word frequency plot. "Real life" and "real world" were used together but

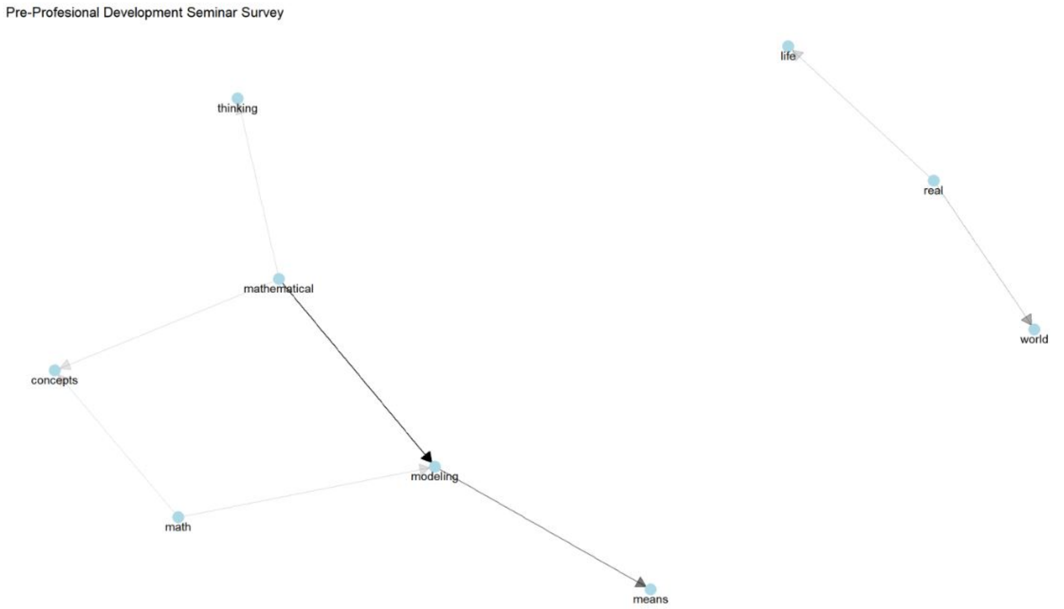


Figure 25: Bigram network for the pre-survey responses where darker lines represent more frequently used bigram connections.

not as frequently as some other phrases. “Mathematical modeling” and “math modeling” were used together a lot, with the bigram “modeling means” seeing frequent but less use than the other bigrams. This could possibly suggest that the phrase “mathematical modeling” is typically being used as a reiteration of the prompt.

There is still a lot not known about these relationships among words. Because this network is only looking at bigrams, it is possible that the phrase “mathematical modeling means” is not being used and “mathematical modeling” and “modeling means” are instead being used in two different contexts. Due to the frequency of these three words, it may be unlikely that such a separation is occurring, but it is still a possibility. It should also be noted that bigrams with stop words being removed may have important meaning that a stop word alone would not have. A phrase like “modeling is” is removed and may have similar context to the prompt in the wording “mathematical modeling is”.

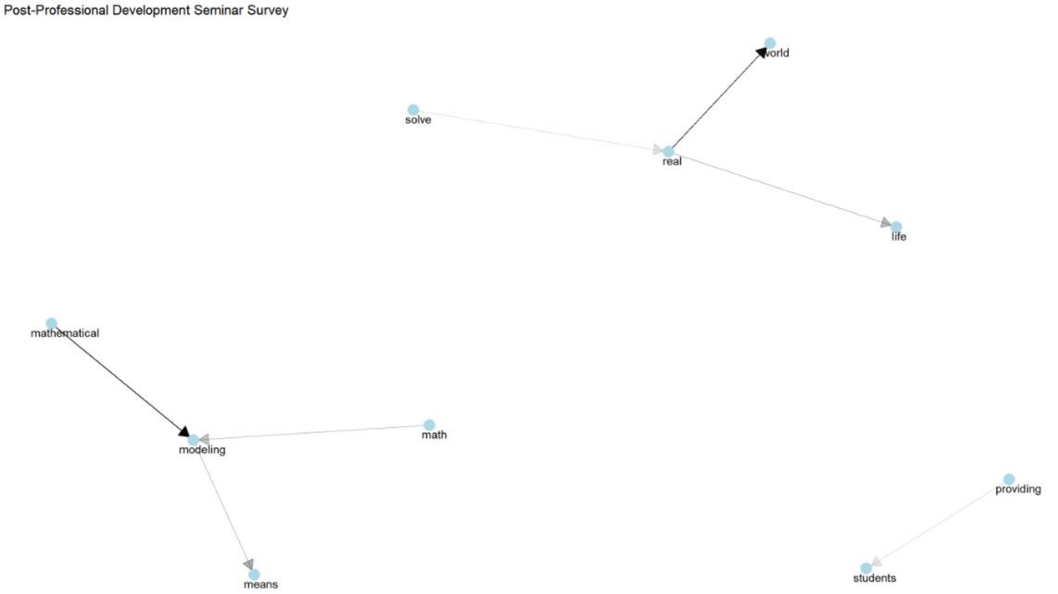


Figure 26: Bigram network for the post-survey responses where darker lines represent more frequently used bigram connections.

To get more insight on how bigram relations changed after the professional development session, we can look at the bigram networks from the two follow-up surveys. What is noticed is that some bigram connection are the same as with the pre-professional development survey. Connections of words like “mathematical” and “math” to the word “modeling” occurred in all three surveys. There is still bigram relationships between the word “real” and the words “life” and “world”. However, in these post-professional development surveys, there is a higher frequency of the bigram “real world”. Based on these observations it is possible that in all surveys many participants were reiterating the question by stating “mathematical modeling means” and that in the post surveys it was common for participants to relate concepts of mathematical modeling in an elementary school setting to involving a connection to the “real world”.

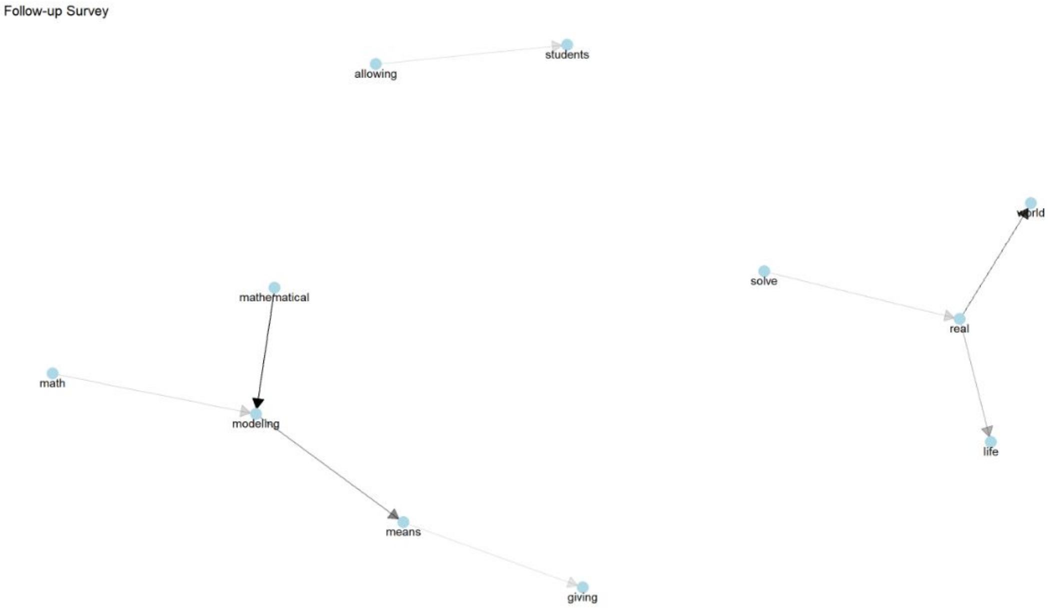


Figure 27: Bigram network for the follow-up survey responses where darker lines represent more frequently used bigram connections.

A few other connections, such as the bigrams like “allowing students”, “providing students”, and “means giving” also appear in the post surveys. It is possible that seeing these patterns has important context in the instructors’ understanding of mathematical modeling from the professional development seminar. However, it is important to remember that all of the exploratory visualizations are just explorations for hypothesis building and that other pieces of information such as reading/coding of the survey responses, subject/literature expertise, and continued studies should be used in conjunction with these visualizations to solidify theories. To understand more about the valuable techniques of exploring textual data through data visualization please reference these books:

Berry & Kogan 2010, p. 105-126; Cao 2016; Silge & Robinson 2017

4.3 Case Study - Latent Dirichlet Allocation

The idea behind Latent Dirichlet Allocation (LDA) is that there is a set of hidden, or latent themes, behind a set of text documents. We do not know what each topic represents, but we can explore the composition of each topic to try and understand its meaning. Using LDA assigns each document in the model as a mixture of these topics, and each topic is defined of a mixture of all the words from the document set (Silge & Robinson 2017). This is a form of soft classification as each document is a composition of all the latent themes instead of being assigned to only one (Aggarwal & Zhai 2012). LDA is a Bayesian technique that works by modeling term frequency in a probabilistic framework (Hornik & Grun 2011). Basing the model off of word frequency comes with the assumption that the order of the words in a document does not matter, possibly resulting in missed information and meaning (Hornik & Grun 2011). Another constraint is that the number of topics being modeled needs to be specified before running the analysis.

There are several pieces to a LDA model. All K topics (ϕ) needs to be assigned a distribution of words that can be modeled by using a multinomial distribution as the sampling model (Aggarwal & Zhai 2012; Hornik & Grun 2011). To draw samples of these distributions, we use a Dirichlet distribution with the parameter β. However, these topics are not the only thing that needs

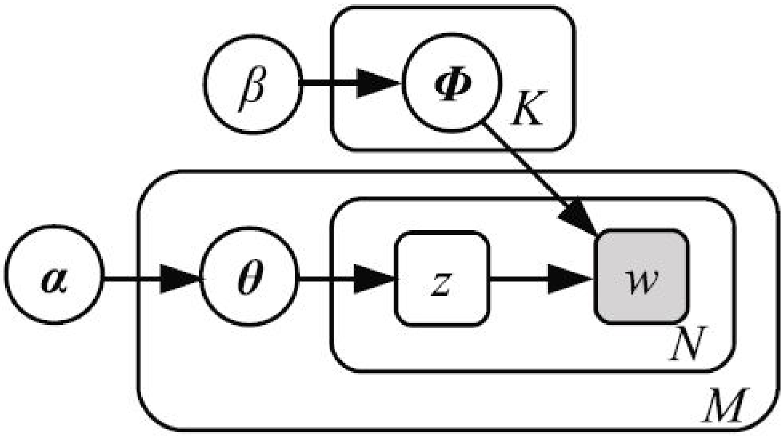


Figure 28: Visualization of an LDA model. Adapted from Aggarwal & Zhai, 2012, p. 142.

to be modeled with a multinomial distribution. All M documents (θ) need to be assigned a distribution of topics. This is also drawn from a Dirichlet distribution with a different parameter which we call α. With all of the background information modeled, it is possible to simulate all N words, and their topic, for every document. This is done by pulling a topic for a word using our multinomial distribution that was drawn for the document. That word is then selected, given the topic selected and that topics multinomial distribution of words. The end result of a single iteration for a simulation is that all N words in every document would be assigned a topic and a word. Using this combined word/topic information, a proportion of word occurrence for each topic can be collected and the proportion of topic occurrence for each document can be recorded.

For this analysis, responses from the question, "What does the phrase 'mathematical modeling' mean to you, in the context of your work as an elementary grades teacher?" were modeled by two latent topics. The hypothesis was that one latent topic would represent documents and words associated primarily with responses from before the professional development seminar while the second topic would represent words and documents for responses after the professional development seminar.

After the model was fit, the results were explored by looking at the words with the five highest probabilities occurring in each topic (Figure 29). Words such as "students" and "mathematical" were both estimated to have a high probability of occurring in the two topics. From these word probabilities, it is difficult to tell if one topic may represent language similar to that used before or after the professional development seminar.

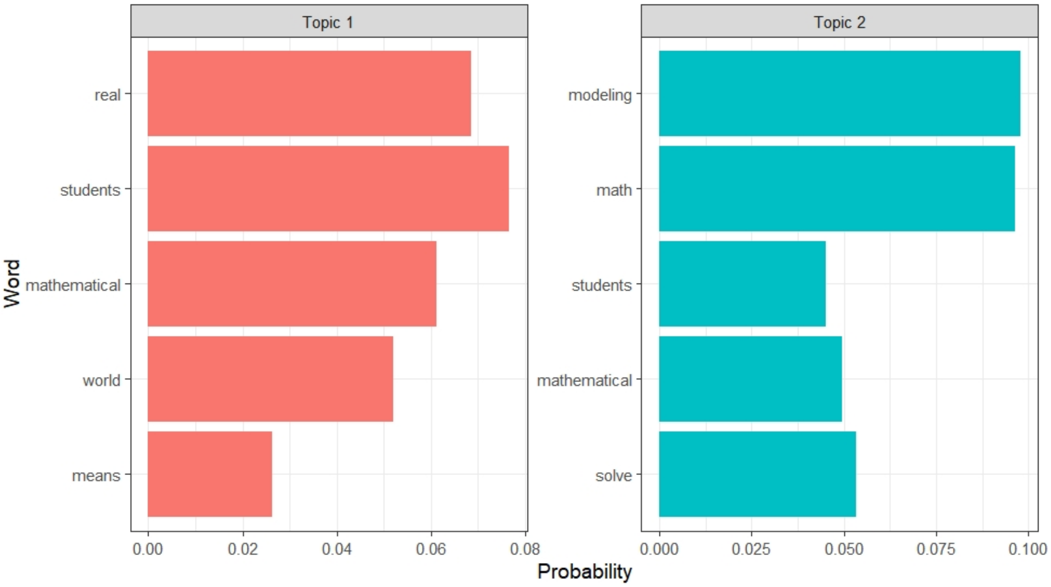


Figure 29: Bar chart of the top five word probabilities for latent topic 1 and 2.

To explore further, ratios of word probabilities (topic 2 / topic 1) between the two topics were plotted (Figure 30). This helps give an understanding of what words have a high probability of belonging to one topic but not the other. The largest and smallest ratio of word probabilities for topic 2 to topic 1 was plotted using a log scale. This visualization displayed information similar to what was observed in the exploratory plots. Words frequently used before the professional development seminars, such as “skills”, “concepts”, and “strategy” had large probabilities of being in topic 2 but not topic 1. Words like “world”, “process”, and “thinking” had high probabilities in topic 1 and were frequently seen in responses after the professional development seminar. This led me to believe that topic 2 may be a latent topic for the pre-survey responses while topic 1 may be a latent topic for survey responses after the professional development seminar.

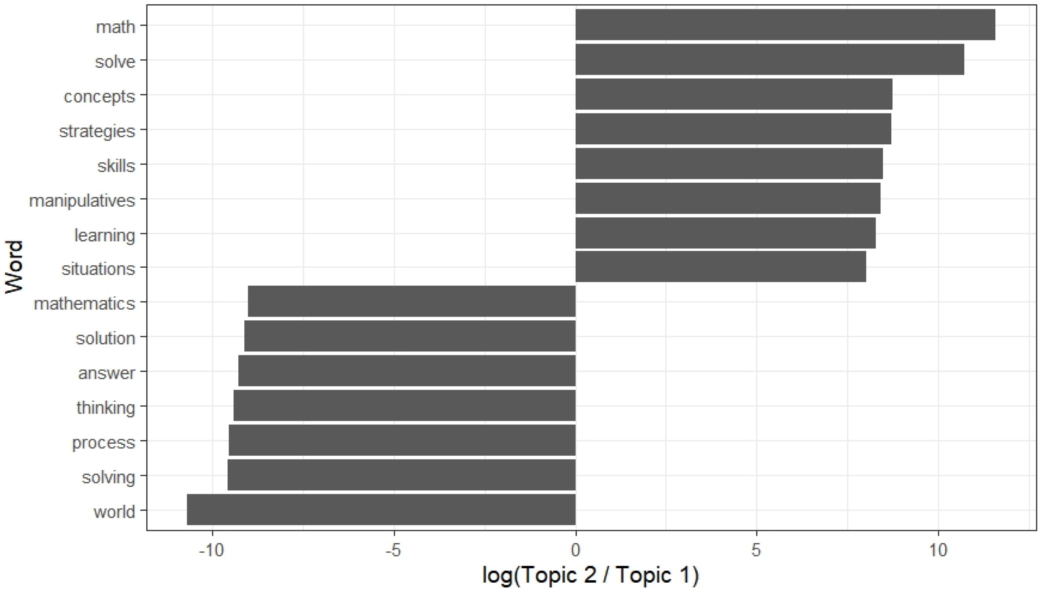


Figure 30: Words with the highest and lowest ratios of word probabilities between latent topic 2 and latent topic 1. Ratio of word probabilities have been transformed logarithmically.

To explore the possible meaning of the topics further, a plot was made to visualize how the different responses were classified as a combination of the two topics (Figure 31). Question responses from the pre, post, and post2 surveys (x-axis) were plotted against the probability of belonging to topic 1 or 2 (y-axis). Each response was colored by the location of the profesional development program.

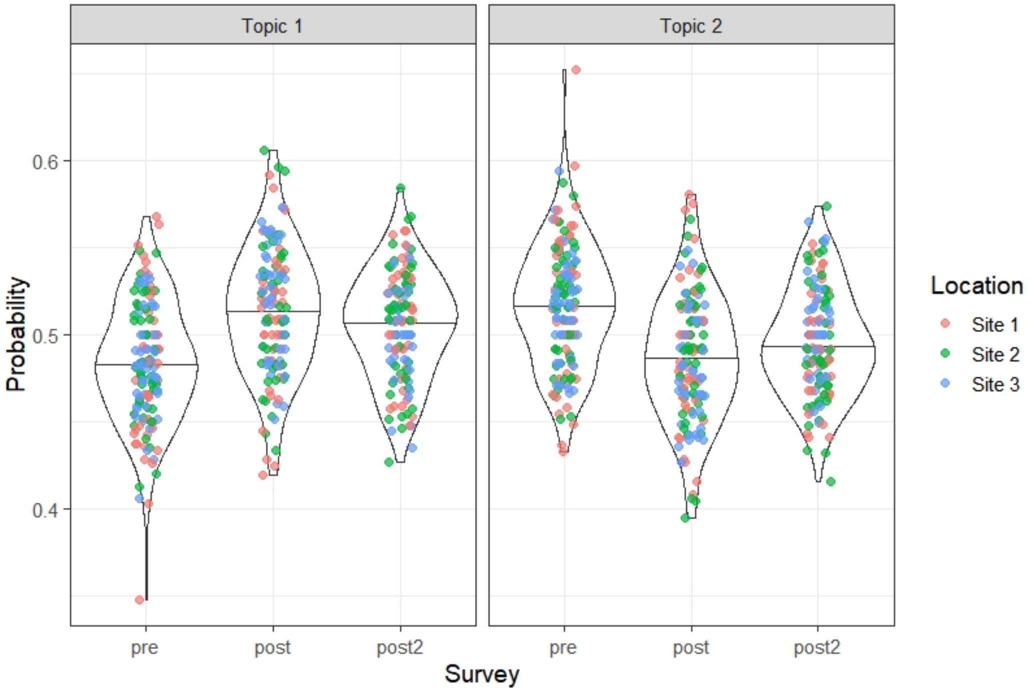


Figure 31: Document probabilities for topic 1 and topic 2. Densities of document probabilities are presented for each survey period with the horizontal line representing the median probability for a given survey.

Looking at the plot shows that the probability of pre-survey responses tend to have a lower median probability of belonging in topic 1 than the other two surveys, but not by much. Considering how variable each set of survey response probabilities are for being drawn from a topic. If there was a clear distinction of topic one representing pre-survey responses, most of the pre-survey responses would have a high probability of belonging to topic 1 while the other two responses would have low probabilities of belonging to topic 1. There is not evidence of that in Figure 31. In addition, there was no clear indication that site is related to the language used to define the two topics.

Based on the results, I was not able to identify before and after professional development survey responses as clear latent topics. There are several reasons in which I believe this happened and more work could be done to explore the topics generated in the future. One possibility is that the topics are capturing something different than before and after professional development survey responses. It would be worthwhile to explore documents with topics with the highest probability of belonging to one topic. The main reason I believe that it is difficult to distinguish individual responses being clearly defined to a topic is the similarity of words used in every response. I believe that certain words were used frequently in all responses because they were mentioned in the prompt. It would be interesting to see if it is also difficult to pick up language differences between the survey responses using qualitative coding methods because of the similarity of words used in both documents.

5 Conclusion

This paper gave an overview of text mining by discussing a framework for conceptualizing the discipline. Text mining can be broken down conceptually into eight practice areas, starting with text preprocessing and moving toward analysis through document clustering, document classification, information retrieval, information extraction, web mining, natural language processing, and concept extraction. While the practice areas are given distinct names, many algorithms and textual problems can overlap concepts from multiple practice areas.

Text mining was applied to a survey question given to educators who went through a professional development seminar on mathematical modeling in an elementary school setting. This case study highlighted the use of techniques to preprocess textual data, visualize the information, and use statistical modeling to explore patterns and generate hypotheses. With textual data being available everywhere, there are many other examples of text mining applications. For more examples of text mining in practice, other cases studies, and practice problems feel free to take a look at the following texts:

Kao & Poteet 2007; Miner et al. 2012, part II; Nedjah et al. 2009; Silge & Robinson 2017, ch. 6-7; Weiss, Indurkhya, & Paul 2015, p. 165-201

6 Appendix: Code

```r
# Initial setup
#load packages
library(readxl) # Used to import data
library(dplyr) # Used for data manipulation
library(ggplot2) # Used for plots
library(tidytext) # Primary text manipulation package
library(tidyr) # Used for conversion between long and wide format
library(igraph) # Used to convert bigrams to network
library(ggraph) # Used to plot bigram network
library(scales) # Gives scale transformations in ggplot
library(topicmodels) # Library for topic modeling (LDA)

# Read in data
math_data_raw <- read_excel("Combined Data Edit.xlsx")

# Set plotting theme
theme_set(theme_bw())


################################################################
####                   Text Preprocessing                   ####
################################################################
```

```
# data cleaning
math_data_factor <- math_data_raw %>%
  mutate(test = factor(test,
                       levels = c("pre", "post", "post2")),
         year = factor(year))

# Text preprocessing for word tokens
math_data_word <- math_data_factor %>%
  select(id, test, year, open2, open3) %>%
  gather(key = question, value = response, open2, open3) %>%
  unnest_tokens(word, response) %>%
  anti_join(stop_words)

# Text preprocessing for bigrams as tokens
bigram_counts_2 <- math_data_factor %>%
  select(id, test, year, open2, open3) %>%
  gather(key = question, value = response, -id, -test, -year) %>%
  unnest_tokens(bigram, response, token = "ngrams", n = 2) %>%
  filter(question == "open2") %>%
  select(-question) %>%
  group_by(test) %>%
  count(bigram) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  ungroup() %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  arrange(desc(n))

# Convert bigram data into a graph dataframe
# Pre
bigram_graph_pre <- bigram_counts_2 %>%
  filter(test == "pre") %>%
  select(-test) %>%
  filter(n > 5) %>%
  graph_from_data_frame()

# Post
bigram_graph_post <- bigram_counts_2 %>%
  filter(test == "post") %>%
  select(-test) %>%
  filter(n > 5) %>%
  graph_from_data_frame()

# Post2
bigram_graph_post2 <- bigram_counts_2 %>%
  filter(test == "post2") %>%
  select(-test) %>%
```

```r
  filter(n > 5) %>%
  graph_from_data_frame()

# Convert data into document-term matrix
# Long format data that identifies document, word, and word count
question_2_word_count <- math_data_word %>%
  filter(question == "open2") %>%
  mutate(document = paste(id,test, sep = "_")) %>%
  count(document, word, sort = T) %>%
  ungroup()

# Convert long format to document-term matrix
question_2_docmat <- question_2_word_count %>%
  cast_dtm(document, word, n)

#################################################################
####                   Text Visualization                    ####
#################################################################

# Word frequency scatterplot
set.seed(532019)
math_data_word %>%
  filter(question == "open2") %>%
  count(test, word) %>%
group_by(test) %>%
mutate(proportion = n / sum(n)) %>%
select(-n) %>%
spread(test, proportion) %>%
gather(test, proportion, post, post2) %>%
 mutate(col = case_when(word == "question" ~ 1,
                        word == "solution" ~ 1,
                        word == "world" ~ 1,
                        word == "real" ~ 1,
                        word == "answer" ~ 1,
                        word == "question" ~ 1,
                        word == "concrete" ~ 1,
                        word == "concepts" ~ 1,
                        word == "models" ~ 1,
                        word == "manipulatives" ~ 1,
                        TRUE ~ 0.1)) %>%
ggplot(aes(x = proportion, y = pre)) +
geom_jitter(aes(alpha = col, color = col),
            size = 2.5, width = .05, height = .05) +
geom_abline(color = "grey40", lty = 2) +
geom_text(aes(label = word),
          check_overlap = TRUE, vjust = 1.5) +
scale_x_log10(labels = percent_format()) +
scale_y_log10(labels = percent_format()) +
  facet_wrap(~test, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "pre-survey", x = "post-surveys",
       subtitle = "What does the phrase mathematical modeling
       mean to you, in the context of your work as an elementary
       grades teacher?")

# Bigram network plots for pre, post, and post2
arrow_tip <- grid::arrow(type = "closed",
                         length = unit(.15, "inches"))

# Pre
set.seed(42)
ggraph(bigram_graph_pre, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = arrow_tip,
                 end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_classic() +
  ggtitle("Pre")

# Post
set.seed(42)
```

```
ggraph(bigram_graph_post, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = arrow_tip,
                 end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_classic() +
  ggtitle("post")

# Post 2
set.seed(42)
ggraph(bigram_graph_post2, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = arrow_tip,
                 end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_classic() +
  ggtitle("post2")


#################################################################
####                   LDA Analysis                          ####
#################################################################
```

```r
# Conduct LDA using gibbs sampler
q2_lda <- LDA(question_2_docmat, k = 2,
              method = "Gibbs", control = list(seed = 42))

# Per topic word probabilities
q2_topics <- tidy(q2_lda, matrix = "beta")

# Get ratio between topics
beta_spread <- q2_topics %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  mutate(log_ratio = log2(topic2 / topic1))

# Plot largest magnitude ratios
beta_spread %>%
  filter(log_ratio >= 8 | log_ratio <= -9) %>%
  mutate(term = factor(term,
                       levels = term[order(log_ratio)])) %>%
  ggplot(aes(x = term, y = log_ratio)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  ylab("log(Topic 2 / Topic 1)") +
  xlab("Word")
```

```r
# Per document probabilities
q2_gamma <- tidy(q2_lda, matrix = "gamma")

# Identify site and survey
q2_gamma <- q2_gamma %>%
  separate(document, c("id", "test"),
           sep = "_", convert = TRUE) %>%
  mutate(location = case_when(
    id >= 100 & id < 400 ~ "Site 1",
    id >= 400 & id < 600 ~ "Site 2",
    TRUE ~ "Site 3"))

# Plot document topic probabilities
q2_gamma %>%
  mutate(test = factor(test,
                       levels =  c("pre", "post", "post2"))) %>%
  mutate(topic = factor(topic,
                        labels = c("Topic 1", "Topic 2"))) %>%
  ggplot(aes( test, gamma)) +
  geom_violin(draw_quantiles = .5) +
  geom_point(aes(color = location), size = 2,
             position = position_jitter( width = .1, height = 0),
             alpha = 0.7) +
  facet_wrap(~ factor(topic)) +
  xlab("Survey") +
  ylab("Probability") +
  guides(color=guide_legend(title="Location"))
```

7 References

Aggarwal, C. C. (2018). *Machine learning for text.* Cham: Springer.

Aggarwal, C. C., & Zhai, C. (2012). *Mining text data.* Boston, MA: Springer.

Berry, M. W., & Castellanos, M. (2008). *Survey of text mining II clustering, classification, and retrieval* (2nd ed.). New York: Springer.

Berry, M. W., & Kogan, J. (2010). *Text mining: Applications and theory.* Chichester, U.K.: Wiley.

Burroughs, E., & Fulton, E. (2018). *IMMERSION: Qualitative research analysis.* Montana State University.

Cao, N. (2016). *Introduction to text visualization.* Retrieved from http://dx.doi.org/10.2991/978-94-6239-186-4

Hirsch, C. R., & McDuffie, A. R. (2016). *Mathematical modeling and modeling mathematics.* Reston, VA: National Council of Teachers of Mathematics.

Hornik, K., & Grün, B. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1-30.

Jo, T. (2019). *Text mining.* Retrieved from https://doi.org/10.1007/978-3-319-91815-0

Kao, A., & Poteet, S. R. (2007). *Natural language processing and text mining.* New York: Springer.

Kumar, A., & Paul, A. (2016). *Mastering text mining with R.* Retrieved from http://proquest.safaribooksonline.com/9781783551811

Miner, G., Elder, J., IV, Fast, A., Hill, T., Nisbet, R., & Delen, D. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Retrieved from http://ebookcentral.proquest.com/lib/montana/detail.action?docID=842198

Nedjah, N., Mourelle, L. de M., Kacprzyk, J., França, F. M. G., & Souza, A. F. de D. (2009). *Intelligent text categorization and clustering*. Berlin: Springer.

Reamy, T. (2016). *Deep text: Using text analytics to conquer information overload, get real value from social media, and add big(ger) text to big data*. Medford, NJ: Information Today, Inc.

Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *JOSS*, *1*(3). https://doi.org/10.21105/joss.00037

Silge, J., & Robinson, D. (2017). *Text mining with R: A tidy approach* (1st ed.). Sebastopol, CA: O'Reilly Media.

Weiss, S. M., Indurkhya, N., & Zhang, T. (2015). *Fundamentals of predictive text mining* (2nd ed.). London: Springer.