LDaRM: A Technique for Improving Knowledge Discovery in Large Text Corpora

---

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment
of the requirements for the degree

Master of Science

by

Davis Christopher Loose

August 2020

APPROVAL SHEET

This Thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Author Signature: _____

This Thesis has been read and approved by the examining committee:

Advisor: Cody Fleming

Committee Member: Peter Beling

Committee Member: Barry Horowitz

Committee Member: _____

Committee Member: _____

Committee Member: _____

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

August 2020

Systems Engineering and Environment
Charlottesville, VA Summer 2020

LDaRM:

A Technique for Improving Knowledge Discovery in Large Text Corpora

Davis Loose

Supervisor: Cody Fleming

A thesis submitted to The faculty of the Graduate School of Engineering and Applied Science, University of Virginia in partial fulfillment of the requirements for the degree of Master of Science.

University of Virginia

Acknowledgements

I would like to thank Professor Cody Fleming for his support and guidance throughout the development of this thesis. His feedback was paramount to pushing this from just an idea into research.

I would like to thank Professor Peter Beling for his input and support. His assistance was critical for making a technical idea actionable.

I would like to thank Professor Barry Horowitz for always asking the right questions to come to the right conclusions. Without his input this research may have wandered aimlessly, indefinitely.

I would like to thank Professor Rafael Alvarado for alerting me to the history and depth of language studies and linguistics as a digital discipline. Without his course, much of this work may have been a solution in search of a problem.

University of Virginia

Charlottesville, VA, July 2020

Davis Loose

Abstract

Latent Dirichlet allocation – association rule mining (LDaRM) is a methodology for uncovering latent semantic information from a set of documents. Latent Dirichlet allocation (LDA) is a form of topic modeling, a statistical learning technique that clusters and organizes key words in a set of documents into a set of *topics*, which represent the underlying themes in a corpus. While powerful, LDA often produces topics that are difficult for a human to understand, and larger topic models become cumbersome for an individual to analyze.

Association rule mining (ARM) is a data mining technique for uncovering interesting patterns in data that enable a user to explore new knowledge domains or adjust behavior in response to new information. However, if one were to use ARM on a text corpus, the run-time to find interesting patterns would be cost prohibitive.

LDaRM combines these two techniques to address the weaknesses in each. LDA reduces the dimensionality of the underlying data for ARM, while ARM improves user comprehension of topic models developed using LDA. LDaRM is able to uncover interesting rules that an analyst could use as a springboard to more in-depth analysis of a corpus. In particular, LDaRM is useful for contextualizing named entities - proper nouns that may have ambiguous definitions or are highly context dependent. Finally, LDaRM is an effective tool for identifying terms that confound topics and make it difficult for individuals to interpret topic model results.

Keywords – latent Dirichlet allocation, association rule mining, LDaRM, exploratory text analytics, text mining, data mining

Contents

List of Figures

List of Tables

1 Introduction

The amount of text data available for analysis has rapidly proliferated since the advent of the internet. Newspapers publish articles online, academic journals electronically publish articles, and novels in the public domain are available for download. As such, there is an increased desire for methods that enable an individual to quickly learn about large text corpora without reading each individual text. For example, text summarization is a popular machine learning process through which individual documents are distilled into major themes (Nenkova and McKeown, 2011). Other techniques such as word2vec and t-SNE reveal hidden relationships among words in a corpus (Rong, 2014; Maaten and Hinton, 2008). Regardless of the chosen technique, these methods all seek to provide previously unknown information to a user.

1.1 Exploratory Text Analytics

Exploratory text analytics (ETA) is a machine learning approach for uncovering latent cognitive, cultural, and social content from texts (Hu and Liu, 2012). Latent content may include the underlying themes, concepts, events, or sentiments that exist within a corpus of text documents. A corpus may include books, newspaper articles, academic papers, blogs, or social media posts. The methods used for ETA are typically unsupervised; they are tools to support human interpretation of large bodies of text. One may use hierarchical clustering to associate several documents (Sahoo et al., 2006); classify patents using principal component analysis (Kaur and Sapra, 2013); or apply sentiment analysis to trace narrative arcs in fiction (Gao

et al., 2016). An ETA technique of particular interest is topic modeling.

1.1.1 Topic Modeling

Topic modeling is an unsupervised text analytics method used to discover underlying themes in a set of documents (Blei et al., 2003; Griffiths and Steyvers, 2004; Hannigan et al., 2019). More specifically, topic modeling clusters words and documents in terms of their relative frequency of occurrence. However, the output of topic models is often difficult to interpret and provides few actionable insights (Kumar et al., 2019). Much research has been conducted to enhance topic models through improved computational performance, enhanced readability, and the inclusion of human input and external information (Li et al., 2017). While several methods exist for developing topic models, one of the most common is latent Dirichlet allocation (LDA).

Topic modeling for ETA is frequently a two-fold task. First, a user may seek to find the high-frequency words to understand which important terms appear throughout the corpus. Second, a user may look for exclusive terms – words that are rare but may hold key information for understanding the corpus (Bischof and Airoldi, 2012). However, the LDA process determines which terms constitute a topic by minimizing *perplexity*, a metric that will be discussed in greater detail in subsequent sections. Unfortunately, lower perplexity does not equate to improved readability or comprehension for a user. This research seeks to improve user comprehension and knowledge discovery with regard to topic models through the use of association rule mining.

However, topic models are subject to several limitations. For example, users

often face difficulty with readability and interpretation of large topic models for knowledge discovery tasks (Sievert and Shirley, 2014). Subject matter experts find a topic model with a large number of topics difficult to navigate and comprehend, reducing the effectiveness of the model (Mimno et al., 2011). Even within individual topics, individuals may find it difficult to identify a single unifying theme (Röder et al., 2015). This effect is amplified when using a topic model to explore a new corpus or subject area.

1.1.2 Association Rule Mining

Association rule mining (ARM) is a machine-learning technique used to identify *interesting* relationships between itemsets (Agrawal et al., 1993). ARM is typically framed as a market-basket problem, in which one analyzes *items* contained within *transactions*. Consider the archetypal example of a supermarket, in which receipts represent transactions and products represent items. The manager of the supermarket may perform ARM to identify interesting patterns and boost sales.

While there is no consensus definition of *interestingness*, several criteria can be used to determine if a pattern is interesting, including: conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility, and actionability (Geng and Hamilton, 2006).

Returning to the example, the supermarket manager desires to organize products in her store to maximize profits. Through her analysis she identifies $\{\text{Diapers}\} \rightarrow \{\text{Beer}\}$ as a rule, suggesting that a customer who buys diapers is more likely than expected to also purchase beer (Leskovec et al., 2014). This rule is interesting as it is concise (two items), surprising (beer and diapers are not usually associated),

and actionable (the manager can alter behavior based on the information).

As with topic modeling, ARM faces several constraints. ARM is a computationally expensive task and is not well suited for direct analysis of large corpora. Such an attempt would require tens of thousands of individual items (the dictionary), broken into thousands of baskets (each individual text in a corpus), with each basket containing several thousand words. This problem rapidly becomes intractable - using known complexity metrics as well as exploratory analysis on a personal computer, it is clear that the algorithms used in ARM are not powerful enough to provide information to a user in a reasonable amount of time. Furthermore, ARM suffers a risk of producing false-positive rules, especially as the size of the dataset increases. (Liu et al., 2011)

1.2 LDaRM

This thesis proposes Latent Dirichlet Allocation-Association Rule Mining (LDaRM) as a methodology for understanding the results of a topic model by performing ARM on a topic model's output. LDaRM will leverage the strengths of its component models to assist a user in knowledge discovery while sidestepping many of the weaknesses of either individual approach. The analysis will demonstrate LDaRM on a corpus of articles related to COVID-19, providing a step-by-step walk-through of the methodology as well as a discussion of the results.

The results of this discussion regarding LDaRM will show improved knowledge discovery from topic models by:

- Highlighting interesting association rules and relationships between topics.

Many interestingness measures are already commonly used in ARM and will be discussed in greater detail. LDaRM also leverages visualizations and selection criteria to bring interesting rules to the user's attention

- Contextualizing and describing named entities, as well as providing assistance with term disambiguation. While LDaRM does not seek to be used for named entity recognition, in the service of knowledge discovery, LDaRM provides important context to words with multiple definitions or currently unknown named entities.

- Identifying stop words (that is, non-information bearing words) that confound topic models. Confounding, in this case, refers to an term that an observer would say "does not belong" in a topic. Many of the most interesting rules, in fact, contain stop words. This thesis will discuss the qualitative advantages of stop word identification, as well as quantitative improvements to established metrics (topic coherence) using LDaRM.

1.3 Thesis Outline

Chapter 2 of this thesis provides the background necessary to understand LDaRM, its underlying components, data requirements, and an overview of performance metrics. This includes an overview of relevant work and descriptions of state-of-the-art procedures for ETA. Further, Chapter 2 provides an in-depth history and descriptions of ETA, topic modeling and LDA, ARM, and the performance metrics associated with each. Chapter 3 outlines the LDaRM process from start to finish. Chapter 4 will set the stage for a case study, describing the data set

used for analysis - a selection of academic research articles regarding COVID-19 and SARS-CoV-2. This will be followed in Chapter 5 with an analysis of the data set using LDaRM. The analysis will highlight improvements to topic coherence by using LDaRM as well as a description of how an individual would use LDaRM for knowledge discovery. Chapter 6 will further the discussion of LDaRM, including the limitations of the approach and future research opportunities. Finally, Chapter 7 will present a summary of the work.

2 Background

This chapter will discuss the background information required to understand the LDaRM methodology. The discussion begins with an overview of text analytics and knowledge discovery as a whole. Following the overview, this chapter will describe topic modeling in greater detail including the mechanisms of LDA and associated metrics such as topic coherence. This will lead into an overview of ARM, the algorithms used to collect rules, and measures of *interestingness*. Finally, this section will present some of the research that highlights the usefulness of LDaRM for knowledge discovery.

2.1 Text Analytics and Knowledge Discovery

Text analytics, broadly, refers to the use of statistical learning to automatically collect information about a corpus of text. Classically, there are two pillars of text analytics: knowledge discovery and information retrieval (Hotho et al., 2005). Information retrieval for text may include techniques such as document classification (Yang et al., 2016), named entity recognition (Lample et al., 2016), or web search (Azad and Deepak, 2019). Information retrieval tasks for text mining typically employ supervised machine learning techniques with the objective of uncovering specific facts.

Knowledge discovery, in contrast, is the process of identifying understandable and potentially useful patterns within a corpus (Fayyad et al., 1996). Knowledge discovery tasks may or may not have a specific end goal and typically employ

unsupervised learning techniques, otherwise known as exploratory text analytics (ETA) (Murphy, 2012). Applications of ETA include document summarization (Nenkova and McKeown, 2012), concept mining (Kolekar et al., 2009), and topic modeling (Blei et al., 2003).

Regardless of the analytical method used, the high-level objective of text analytics is the same: to derive human-understandable meaning from a set of texts. Writing (and recorded speech) fixes social, cultural, and factual information about the author and her subject through a process called *entextualization* (Park and Bucholtz, 2009). Text is a sort of fossil – it is an artifact that reflects the ideas, beliefs, and culture of the writer entextualized at the time of publication (Pieniążek, 2015). It is the job of ETA to help individuals investigate these fossils and derive meaning from what they find. Inferring meaning from a set of texts is an iterative process dubbed the Hermeneutic circle, pictured in Figure 2.1 (Alvarado, 2020a). The Hermeneutic circle is a classical framework for understanding texts.

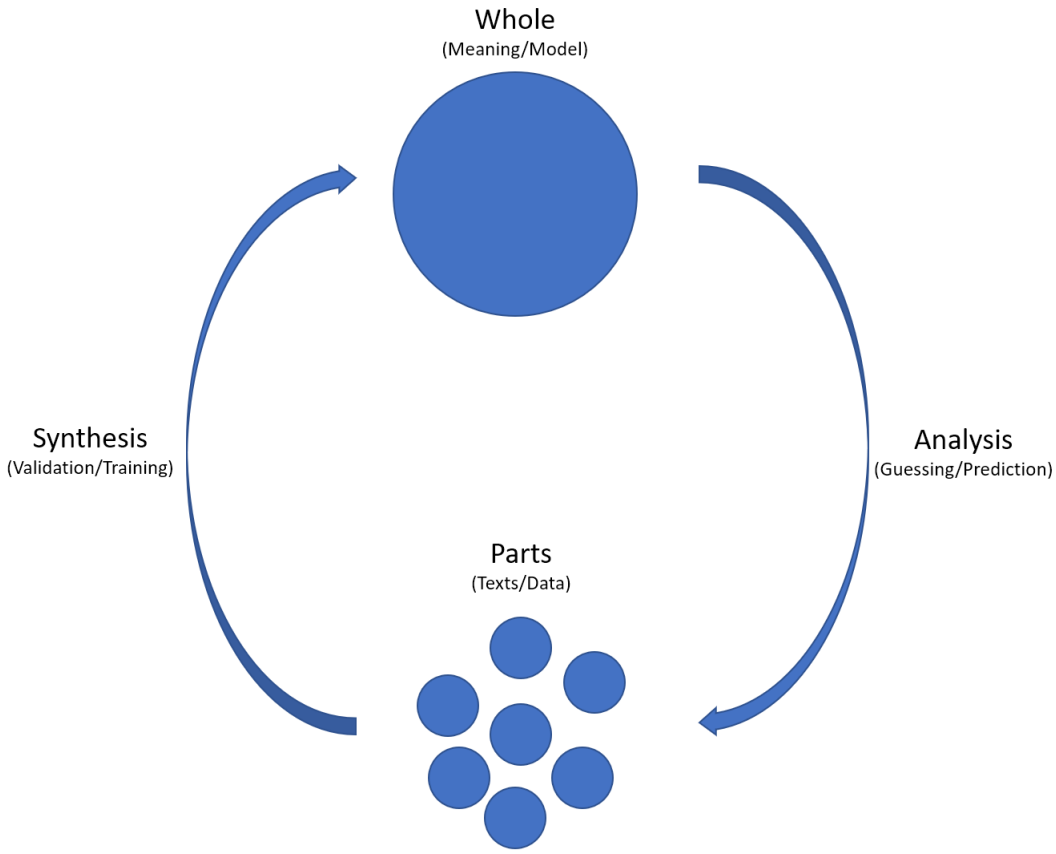

Figure 2.1: The Hermeneutic Circle, adapted from (Boell and Cecez-Kecmanovic, 2010) and (Alvarado, 2020a)

The Hermeneutic circle is based on the premise that a corpus cannot be completely understood as a whole or as individual parts, one needs both to derive meaning from the texts. *Meaning* in this context refers to the relationship between the reader, the texts, and societal context that allow the reader to understand latent cultural and historical information (Klausen, 2017). For example, the American television series *M*A*S*H* is both a situation comedy and a commentary on America's involvement in the Vietnam War. In parts, each episode is intended to make the viewer laugh. When taken as a whole, the series is a commentary on America's role in global affairs. Neither of these points can exist on their own, hence the circle.

Uncovering the meaning of a set of documents is iterative; it requires constant re-visitation. An individual begins at the whole, with limited understanding of the meaning of the corpus. One then performs an analysis, or some form of mapping of prior expectations to the individual parts. An individual then observes the parts, which may confirm or contradict the prior beliefs. One then synthesizes their findings from observing individual parts, forming a new understanding of the corpus as a whole. This is repeated until the individual has satisfactorily derived meaning from the text.

This process can be mapped to ETA and knowledge discovery. One begins with a cursory knowledge of the corpus – in a statistical modeling sense, this represents one's prior beliefs. One then applies their priors to the data. The data is then synthesized and used for training, yielding a statistical model of the text. The new model is used to update prior beliefs, and the cycle continues. In theory one could navigate the circle indefinitely, but the objective is to uncover "valid, novel, useful, and understandable patterns" which allow one to derive meaning from a corpus (Fayyad et al., 1996). With this in mind, it is critical to understand the text and language models used in ETA and learn the form useful patterns may take.

2.1.1 Text Models and Language Models

2.1.1.1 Text Models

As previously stated, text is the entextualization of latent social, cultural, historical, and factual information. ETA is a tool used to assist with decoding text to uncover such information. ETA accomplishes this by mining one or more of the three main elements of text: **structure, sequence, and symbol** (Alvarado, 2020a).

Broadly, text can be described as a structured sequence of symbols (Pierce, 2019; Alvarado, 2020a). Structure refers to the framework or *shape* of a text, usually represented by a hierarchy. For example, in a novel the structure could be book → chapter → paragraph → sentence → word. One could mine the structure of a corpus to establish authorship, as some have done to determine which parts of Shakespeare's plays were penned by Shakespeare himself or by other writers (Aljumily, 2015).

Symbol refers to the smallest information-bearing elements of text. Symbols include words and punctuation. One could mine symbols to uncover trends in a language or to estimate public sentiment regarding a topic or event (Roth, 2016). The Google Ngram Viewer, used to identify word use over time, relies heavily on symbol (Google, 2012).

Sequence is the order of symbols in a text and provides much of the grammatical structure of a document. Sequence helps make language predictable and understandable. Sequence is a major factor in differentiating genres of writing - it is what makes poetry stand out from prose (Rudy, 2010). In ETA, one may mine sequence for part-of-speech identification (Wang et al., 2015).

Each of these three elements of text need not be analyzed alone – there are many ETA techniques that mine two or all three elements at once to derive meaning from a corpus. Named entity recognition – the automated process by which proper nouns are identified in a text – leverages both sequence and symbol through use of neural networks (Lample et al., 2016). By mining structure and symbol, researchers have developed robust document classifiers (Clement and Sharp, 2003). Using all three

elements of text, individuals can automatically generate syntactic annotations (Lin et al., 2012) or summaries of entire documents (Nenkova and McKeown, 2012). Many of these techniques rely on a special class of text models called *language models*

2.1.1.2 Language Models

Language models are statistical distributions of sequences of symbols (Ponte and Croft, 1998). Language models are derived from C. E. Shannon's Theory of Communication in that words in a message are not chosen at random, but rather intentionally selected in a way that is measurable and predictable (Shannon, 2001). If an author is writing about a particular subject, there is an increased likelihood that they will choose certain words related to the subject. The likelihood of a specific sequence of n words can be seen in Equation 2.1 (de Kok and Harm, 2010).

$$P(w_1, w_2, w_3, ..., w_n) = \prod_i P(w_i | w_1, w_2, w_3, ..., w_{i-1}) \tag{2.1}$$

That is, the likelihood of word w_i appearing in a sequence is dependent on which words precede it. Many modern applications such as text editors, search engines, and messaging services rely on language models. Language models are the basis of text prediction (Wandmacher and Antoine, 2008), auto-complete (Fowler et al., 2015), text-to-speech (Chelba et al., 2008), and machine translation (Tang et al., 2018).

Building useful language models is critical for effective ETA, especially as one moves from the statistical model to deriving meaning. Language models allow a

user to identify unique patterns in data – perhaps a particular word or phrase appears with more regularity in a particular document compared to the corpus as a whole. The objective of ETA, then, is to tune language models to identify *significant words*, as shown in Figure 2.2 (Luhn, 1958).

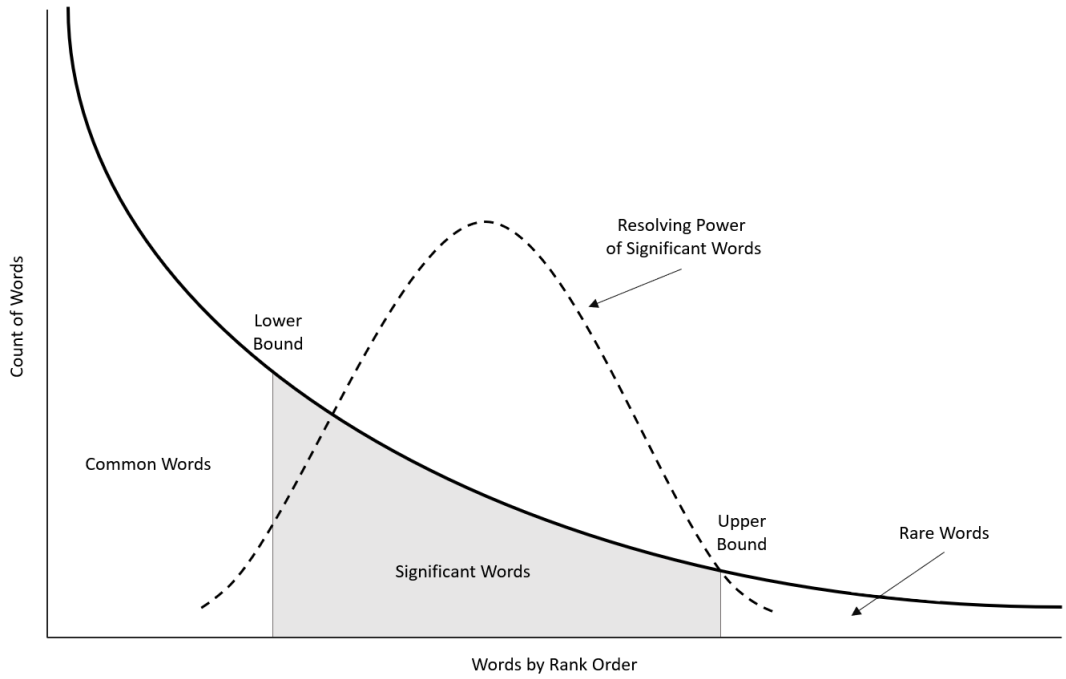

Figure 2.2: Luhn's Model of Significant Words (Luhn, 1958)

Effective ETA seeks to use language models to identify significant words. Common words such as articles (the, a), prepositions (to, in, at), and pronouns (he, she, it) have grammatical importance but do little to deliver a message – that is, they do not shed much light on the meaning of text. Luhn declared such terms *stop words*, which will be discussed in greater detail in subsequent sections (Luhn, 1960). Rare words also contribute little to the to meaning of a corpus. Words that appear once or twice out of hundreds of thousands of terms in a corpus are unlikely to be part of any latent themes.

Significant words provide *resolving power* – a loan term from optics that describes

the degree of discrimination of a model. According to Luhn, deriving accurate lower and upper bounds for significant terms is a matter of experience and trial-and-error, much in the vein of the Hermeneutical circle. Domain knowledge of a corpus would allow an individual to determine if an ambiguous term belongs in the language model – for example, the word *cell* in a corpus of newspapers is likely more significant than a corpus of biology textbooks. Developing the model may surface other terms that are too frequent or infrequent and can be disregarded in future iterations of the model.

2.2 Topic Modeling

Topic models are a category of ETA techniques used to mine structure and symbol to assign labels to words in a corpus (Wallach, 2006). At a higher level, topic modeling can be considered a type of clustering. The intuition behind topic models is fairly straightforward – if a document focuses on a particular topic, one would expect certain terms to appear in the document. For example, a legal brief will contain the words *court* and *defendant*, while a math textbook will contain the words *calculus* and *algebra*. Topic models mine the entire corpus to identify latent themes within the texts.

The first topic modeling techniques were developed to address the *synonymy* and *polysemy* problems in information retrieval Papadimitriou et al. (2000). Synonymy is the exclusion of synonyms from search results (missing results that include *poodle* when searching *dog*). Polysemy is the inclusion of synonyms in search results when they should not be included (including results for *iPhone* when searching for content on *apples*). Rather than searching for specific terms, topic modeling

allowed individuals to search by topic. As more advanced topic modeling techniques were developed, many researchers began to use topic models to uncover latent information from large text corpora, such as identifying trends in news cycles (Hu et al., 2014). Topic modeling has been used for targeted ETA, such as identifying experts on a specific subject (Momtazi and Naumann, 2013), or in very open-ended contexts such as investigating social media posts (Hong and Davison, 2010).

Topic models represent a relationship between documents, words, and topics. Consider a hypothetical example: an individual collects articles from several magazines and uses them as the corpus to train a topic model. One may see the topic breakdown as outlined in Figure 2.3.

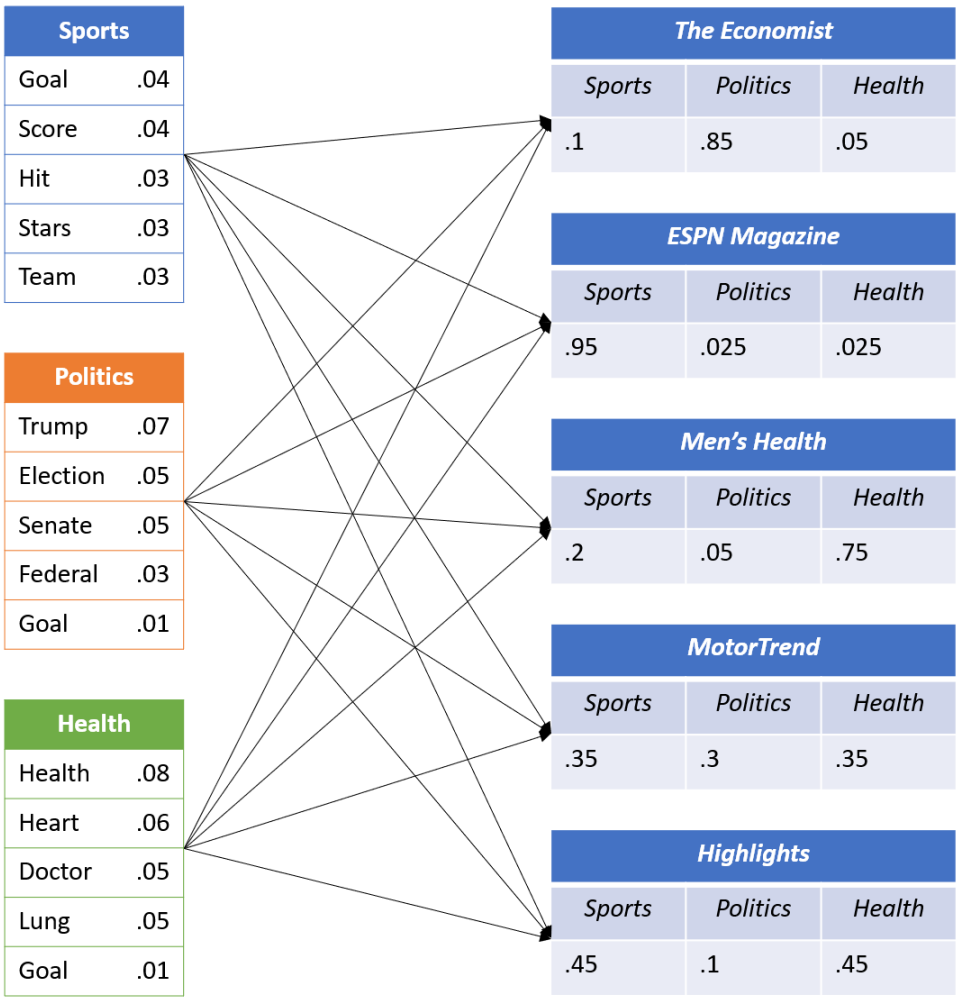

Figure 2.3: Sample Topic Model

Each topic is a mixture of words, with each word possessing a likelihood of belonging to that topic. The higher the number, the stronger the relationship between the word and the topic. While each topic highlights certain terms, words can belong to more than one topic at a time. Note that *goal* appears in each topic – while it is strongly associated with the *sports* topic, it is still weakly associated with *politics* and *health*.

Further, each magazine is a mixture of topics. Some magazines such as *ESPN Magazine* or *The Economist* are highly focused in only one topic, while others like *Men's Health* are more diverse. Notice that *MotorTrend* is roughly equal across all three topics - this is due to automobiles not fitting neatly into any of the three topics.

As a probability distribution over a set of words, each topic is technically a language model. This is what makes topic models a powerful tool for ETA - topics are able to bring significant words to a user's attention directly. Further, a topic highlights terms that are closely related to one another, opening new avenues for knowledge discovery.

Although topic models are powerful exploratory tools, they are not without their constraints. Topics are often dominated by uninteresting noise terms or stop words (Wallach et al., 2009), require significant tuning (Choo et al., 2013), or are otherwise difficult to distinguish from a random assortment of words (Chang et al., 2009). Generally, while individuals prefer topic models with a large number of topics, domain experts find it difficult to identify actionable patterns in the data as the number of topics grows (Mimno et al., 2011). Furthermore, LDA, a specific

type of topic model, generates topics to minimize perplexity – a measure of the *surprise* in the model (both LDA and perplexity will be discussed in greater detail in the following section). However, perplexity is a poor proxy for human judgement and does not correlate to the quality of a topic Chang et al. (2009). For this reason, researchers have developed a metric called *topic coherence* to determine how interpretable a topic is (Newman et al., 2010). Many techniques, including LDaRM, are used to improve the quality of topic models.

2.2.1 Data Preprocessing for Topic Models

The process of developing topic models (and many other models used for ETA) follows a pipeline, as pictured in Figure 2.4 (Hahn et al., 2007).

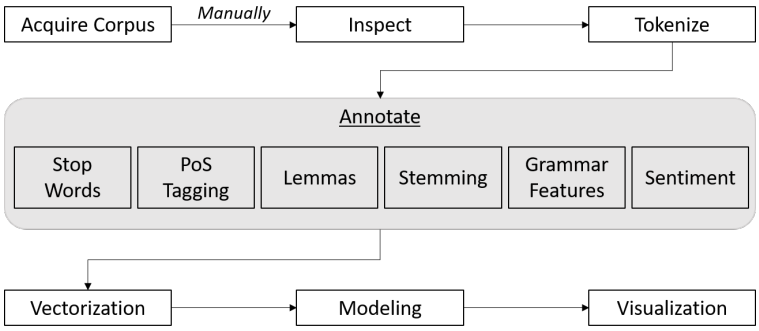

Figure 2.4: High Level Text Analytics Pipeline

Acquiring a corpus requires collecting digital editions of texts. Inspection is a manual process through which an analyst observes features of a text. This may include noting the document hierarchies (book → chapter → paragraph, etc.) and developing early opinions about the corpus.

Tokenization is the process of dividing text into individual symbols for analysis. Typically this is done at the level of a single word (unigrams), but tokens can consist of two or more contiguous words as well (bigrams and n-grams respectively). Tokens

are usually separated by white space and punctuation, but there are circumstances in which this heuristic will fail (it is likely that *Mr. Smith* should be considered a single token rather than both *Mr* and *Smith*). As such, much research has focused on developing effective tokenizers to accommodate such exceptions (Manning et al., 2009).

Once a corpus is tokenized, it is annotated. Annotations are token-level features that add context to a word. Part-of-speech tagging, grammatical features, and sentiment values are some common annotations. However, not all annotations are necessary for all types of ETA, and in many circumstances it is worth excluding some of the labels (Manning et al., 2009). The annotations critical for LDaRM include lemmatization, stemming, and labeling stop words.

2.2.1.1 Lemmas

Lemmatization is the process by which words with multiple inflected forms are grouped together. Lemmatized words such as *differently, differentiated, differing* and other forms would be converted to the base form *different*. The base form of a word is called its *lemma*. Lemmatizers are a suite of tools trained on massive text data sets and maintain an awareness of token sequence to attempt to automatically group words with their appropriate lemma (Lezius et al., 1998).

2.2.1.2 Stemming

Stemming is similar to lemmatization in that it seeks to reduce the number of forms of a word, but rather than grouping terms into a single lemma a stemmer will reduce a token to the root-form of the word. For the *different* example above,

the stem would be *differ*. Stemmers are less selective than lemmatizers as they disregard context - a stemmer may reduce *cars, carton*, and *carousel* into the form *car*. This is due to the fact that stemming algorithms rely on heuristics. However, stemming is extremely effective for dimensionality reduction without being a substantial hindrance to model performance (Porter, 2001).

2.2.1.3 Stop Words

Stop words are terms that are not information bearing with regard to text comprehension. Function words such as *the, of, it, be, as, and, how*, and *in* are usually considered stop words. Stop words tend to confound models, especially in ETA. Recall Luhn's model of significant words in Figure 2.2 - stop words fall in the *common words* portion of the graph. As the objective of ETA is to identify significant words, most models exclude stop words all together (Wilbur and Sirotkin, 1992).

However, there is not a single fixed set of stop words. While many terms (such as those mentioned above) are almost always stop words, some stop words are corpus-specific. In a corpus of legal briefs, the term *prosecutor* may become a stop word - a term that appears so often that it does not effectively convey any special meaning (Schofield et al., 2017).

Selecting stop words is highly subjective and time consuming process. However, stop word removal is an important aspect of the LDaRM process. Research has shown that effectively removing stop words improves the quality of topics as measured by topic coherence (O'callaghan et al., 2015). Further, it is suggested that improved topic coherence enables individuals to more easily interpret topic

models (Mimno et al., 2011).

2.2.1.4 Vectorization

Most topic models, including LDA, are bag-of-words models (Alghamdi and Alfalqi, 2015). A bag-of-words disregards word sequence and treats each document simply as a collection of its component terms. A bag-of-words model is also computationally convenient as it is a vector space representation of a text in the form of a *document-term matrix* (DTM), with rows representing each document, and columns representing the frequency of each word in the document (Daniilidis et al., 2010). Consider two sentences "I am eating lunch" and "I am eating dinner". These sentences can be represented in a DTM as in Table 2.1:

Table 2.1: Example of a DTM

I	am	eating	lunch	dinner
1	1	1	1	0
1	1	1	0	1

Bag-of-words models are extremely effective for developing co-occurrence networks, and co-occurrence is a foundational property that enables one to move from frequency to meaning (Turney and Pantel, 2010). However, as noted, frequency alone is inadequate for determining a word's significance and meaning. Stop words appear frequently, but as they are not information bearing and are rarely useful for analysis. As such, it is common to weight terms by comparing their frequency in a document to their frequency in the corpus as a whole. These weights are called *term frequency - inverse document frequency*, or *TF-IDF* (Jones, 1972).

There are dozens of formulas for calculating the adjusted weights of a term using

TF-IDF, but all take the same form: the weight is a product of local term frequency and global frequency as seen in Equation 2.2.

$$weight_{word} = document\_frequency_{word} * corpus\_frequency_{word} \tag{2.2}$$

LDaRM will utilize the raw frequency of a term in a document, and the global frequency log-normalized for document length, as seen in Equation 2.3. To calculate the weight of word i in document j given a corpus of D documents, the TF-IDF equation is:

$$weight_{i,j} = frequency_{i,j} * log_2 \frac{D}{docs\_with\_i} \tag{2.3}$$

TF-IDF is a critical tool for identifying significant words - words that appear frequently across all documents are penalized by the global term, while words that appear infrequently in a small number of documents are penalized by the local term. While one could use a DTM as the input for a topic model, most users will use a TF-IDF. This is true for latent Dirichlet allocation as well (Blei et al., 2003).

2.2.2 Latent Dirichlet Allocation

There are a variety of implementations of topic modeling, including latent semantic analysis (LSA) (Kintsch et al., 2000), probabilistic LSA (Xue et al., 2008), correlated topic models (CTM) (Blei and Lafferty, 2006), and latent Dirichlet allocation (LDA) (Alghamdi and Alfalqi, 2015). Each implementation uses a different set of assumptions - for example, LSA uses singular value decomposition to reduce the

dimensionality of the DTM and derive the pairwise similarity of documents in a corpus. PLSA, conversely, uses the latent class model to reduce the dimensionality of the DTM. LDA assumes that the topics follow Dirichlet prior distributions, and estimates the associated parameters.

There are several versions of LDA: some versions may utilize faster algorithms, new sampling techniques, different priors, or include information external to the corpus of study (Jelodar et al., 2019). Correlated topic models, for example, include a covariance matrix with the DTM and samples from both spaces. This research will focus on LDA as described by Blei et al. (Blei et al., 2003) who introduced a generative process for topic models, and as expanded by Griffiths and Steyvers (Griffiths and Steyvers, 2004) who included collapsed Gibbs sampling to infer parameters.

The intuition of LDA is fairly straightforward: first, it is assumed that there are a set of topics available to write about in any given language. These topics are language models that represent latent structure of a text. Second, a corpus contains only a subset of all available topics, and this subset is recognizable based on the words chosen by the author. Third, it is assumed that there is a dictionary of terms, and these terms exist as a universal language model containing the expected frequency of terms as they appear globally. A topic, then stands out as the relative frequency of terms differs for this topic when compared to the global language model. Finally, it is assumed that there exists a generative process in which one can randomly select a topic, randomly choose a word from the chosen topic, and repeat this process for the entirety of a corpus. This process will ultimately reveal which terms are most strongly associated with a given topic (Alvarado, 2020b).

This process is traditionally expressed in plate notation, as picture in Figure 2.5:

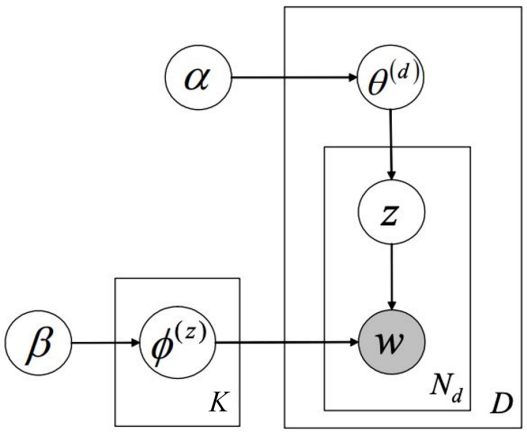

Figure 2.5: Plate Notation of the Latent Dirichlet Allocation Process

Where K is the set of topics, D is the set of documents, and N_d is the set of words in document $d \in D$. α is a vector of prior topic weights (usually defaulting to equal likelihoods for each topic) and β is the vector of prior weights for a word in a topic (which also defaults to equal weights for each word). Further, w is a particular word token in document d, while z is the assigned topic of this word. It is assumed that these assignments are selected from a Dirichlet distribution, which gives LDA its name. The parameters of interest, however, are θ and ϕ - the distribution of topics within documents, and the distribution of words within topics respectively. The generative process can be expressed as follows:

1. For $k = 1...K$:
   (a) $\phi^{(k)} \sim Dirichlet(\beta)$
2. For each document $d \in D$:
   (a) $\theta_d \sim Dirichlet(\alpha)$
   (b) For each word $w_i \in d$:

i. $z_i \sim Discrete(\theta_d)$

ii. $w_i \sim Discrete(\phi^{(z_i)})$

The above generative process yields the following probability model:

$$P(w, z, \theta, \phi, |\alpha, \beta) = P(\phi|\beta)P(\theta|\alpha)P(z|\theta)P(w|\phi_z) \tag{2.4}$$

By Bayesian inference, we can adjust the above probability model to isolate our parameters of interest:

$$P(\theta, \phi, z|w, \alpha, \beta) = \frac{P(\theta, \phi, z, w|\alpha, \beta)}{P(w|\alpha, \beta)} \tag{2.5}$$

Equation 2.5 represents the posterior distribution that we would like to solve to create our topic mixtures (Blei et al., 2003).

With a sufficiently small corpus, Equation 2.5 is solvable. However, as corpora grow, the denominator of the right hand size of the equation – called the Bayesian normalizing constant – quickly becomes computationally prohibitive to solve. LDA in its original form utilized variational interference to estimate the normalizing constant. However, a more intuitive and effective method utilizing Gibbs sampling was later introduced, and is the method employed for LDaRM (Griffiths and Steyvers, 2004).

2.2.3 Gibbs Sampling

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method for obtaining data from a specific distribution. In the case of LDA with Gibbs sampling, the

desire is to estimate $P(z|w)$, the probability of assigning a particular topic given a word. The expansion of this probability can be found in Equation 2.6

$$P(z|w) = \frac{P(w,z)}{\sum_Z P(w,z)} \tag{2.6}$$

One may note that the denominator is intractable - it involves K^N terms, where K is the number of topics and N is the total number of words in the corpus (Porteous et al., 2008).

The intuition of Gibbs sampling can be seen in Equation 2.7, as outlined by Griffiths and Steyvers (Griffiths and Steyvers, 2004). The objective is to calculate the product $P(z_i = j|z_{-i}, w_i, d_i)$, the estimate for the best topic given a word in a document and all other topic assignments.

$$P(z_i = j|z_{-i}, w_i, d_i) \propto \frac{C^{WT}_{w_i j} + \beta}{\sum_{w=1}^{W} C^{WT}_{w,j} + W\beta} \times \frac{C^{DT}_{d_i j} + \alpha}{\sum_{t=1}^{T} C^{WT}_{d_i,t} + T\alpha} \tag{2.7}$$

For clarity, C is a matrix that represents a count of words or documents that does not include the current assignment of z_i. The left hand factor represents how much a topic likes a word, while the right hand factor represents how much a document likes a topic. One may recall that these are the desired outputs of LDA, ϕ and θ respectively. The objective of Gibbs sampling, at a high level, is to randomly assign each word in each document to one of the topics. The process is (Griffiths and Steyvers, 2004):

1. For each document d
    (a) For each word w in d

i. Exclude the current topic assignment from ϕ and θ

ii. Calculate $P(t|d)$, the proportion of words in d that are assigned to topic t

iii. Calculate $P(w|t)$, the proportion of assignments to t that are derived from w

iv. Develop the distribution $P(T) = P(t|d) \times P(w|t)$, the multinomial distribution of all topics

v. Sample from T, weighted by $P(T)$, and increment C matrix

This algorithm needs to be performed many times in order for the ϕ and θ tables to converge on appropriate distributions. Research suggests between 20 and 200 iterations of the algorithm are adequate, depending on the corpus and number of topics (George and Doss, 2017).

2.2.4 Topic Coherence

It should be noted that outside of hyper-parameter selection (the number of topics in a model and the number of iterations of the Gibbs sampler), there is no human intervention in the development of topics. Generally, the quality of a topic model is measured by *perplexity*.

Perplexity, in a statistical sense, is a measure of how well a model predicts a sample. Specifically for language modeling, perplexity is a measure of how well the model predicts the following terms in a sentence. The formula for calculating perplexity can be seen in Equation 2.8 below:

$$2^{H(p)} = 2^{-\sum_x P(x) \log_2 P(x)} \tag{2.8}$$

H(p), in this case, represents entropy, the number of binary elements of information required to deliver a message. $P(x)$ is the probability of a word occurring, and $\log_2 P(x)$ is the factor by which uncertainty is decreased given the knowledge of $P(x)$. Perplexity is often described as a measure of surprise in a model.

Perplexity is expressed as a number, but to an observer there is little intuition in claiming that one's topic model has a perplexity of 500. However, one could interpret perplexity through an example: if a model has a perplexity of 500 and an observer were to guess the next word in a sequence, they are as surprised on average as they would be if they had 500 equally likely words to choose from. If all words in a language model occurred with equal probability, the model would be high-perplexity - it would be nearly impossible to guess the next word of a sentence. If there were only a single word in a language, the language would be low perplexity, it would be simple to guess the next word in a sentence. (Manning et al., 2009).

Perplexity as a measure of topic quality is effective for checking model performance, but it has been found to be a poor predictor of expert opinions on topic quality (Röder et al., 2015). In an experiment in which experts were asked to rate the quality of topics, they found that the topics rated as "poor" had one of four problems (Mimno et al., 2011):

- Words were chained through pairwise connections, linking semantically unrelated terms. For example, *computer, apple*, and *fuji* would appear

together. *Computer* and *fuji* are unrelated, but connected through *Apple*.

- An otherwise good topic had an "intruder" word or words
- The topic appeared random, with very few clear connections between words
- Several of the top words in a topic are connected, but the topic quickly becomes nonsensical

To combat these issues, topic coherence measures were developed. Topic coherence identifies the degree of similarity between words in a topic to rate topic quality. Topic quality, in this case, indicates that the topic is not marred by the problems listed above. *UMass* and *UCI* are two of the most common coherence measures and are highly related. That is, if one takes an action to improve one coherence score (such as increasing the number of Gibbs samples), the other coherence score is likely to improve as well (Röder et al., 2015). Of particular interest is the UMass topic coherence measure as it has been found to correlate highly with expert classification of topic quality and leverages word order within a topic, which is a critical component of ARM.

Each topic receives a UMass score, but the average UMass of all topics is often used to judge model quality. The formula for calculating UMass can be see in Equation 2.9 (Mimno et al., 2011).

$$UMass_T = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{P(w_m, w_l) + \frac{1}{D}}{P(w_l)} \tag{2.9}$$

Where T is the ordered set of the top M words in a topic. $P(w_m, w_l)$ is the frequency of at which w_m and w_l are found together out of all documents D, while

$P(w_l)$ is the frequency of occurrence of a term in document l (Rosner et al., 2014).

While topic coherence measures represent an improvement to topic model interpretability, users still face certain challenges. For example, it is known that topic coherence increases as the number of topics in a model increases - this increase in topics, however, decreases the usability of the model. It becomes difficult for users to understand the corpus as a whole, comparing and contrasting dozens or hundreds of topics at once (Ramage et al., 2009). As such, the LDaRM methodology introduces another machine learning technique to assist users when navigating large numbers of topics.

2.3 Association Rule Mining

Association rule mining (ARM) is used to discover relationships in a large data set that might be non-intuitive or otherwise difficult to discern. As previously stated, ARM is often called a "market-basket problem" in that the input data resembles the sales ledger of a supermarket, consisting of *transactions*, each containing *items*. This framing continues through to the output, as ARM could highlight interesting buying patterns in a supermarket such as the canonical example {Diapers} $\rightarrow$ {Beer}, indicating that customers who buy diapers are more likely than average to buy beer. For LDaRM, the transactions will be the topics generated by LDA, while the items are the top words identified by the model. Such a process can help one find useful and actionable patterns in text data.

To perform ARM, the data is divided into N transactions containing items $q \in Q$, the set of all items. I is any itemset constructed from Q. We define *support S(I)* –

the rate at which I appears in transactions – as:

$$S(I) = \frac{\text{count of transactions with I}}{N} \tag{2.10}$$

An itemset is *frequent* if it has support greater than or equal to the threshold of support s, a parameter set by a user (Agrawal et al., 1993). Now consider an *association rule* $I \rightarrow J$ where I and J are non-intersecting sets of items. Itemset I is called the *antecedent*, while J is called the *consequent* of the association rule. This says that if a transaction contains I, there is a relationship to the presence of J (Geng and Hamilton, 2006). It follows that the support of an association rule is:

$$S(I \rightarrow J) = \frac{\text{count of transactions with I and J}}{N} \tag{2.11}$$

Using these definitions, we derive the *confidence* of a rule as:

$$C(I \rightarrow J) = \frac{S(I \rightarrow J)}{S(I)} \tag{2.12}$$

Confidence is a measure of the probability of J being in a transaction given that it already contains I. A rule is *high-confidence* if it is greater than or equal to the confidence threshold c, also set by the user.

Support and confidence are simple concepts to understand, but in a database with dozens of transactions and hundreds of items, it is often difficult to calculate quickly. However, once calculated, having these values allow one to calculate more complex and useful interestingness measures.

2.3.1 FP-Growth

Given a large database of transactions, it is critical to develop algorithms for mining association rules efficiently. There are two common techniques for identifying rules that exceed support and confidence thresholds: the Apriori algorithm and Frequent Pattern Growth (FP-Growth). The Apriori algorithm works in two phases, first identifying candidate itemsets with at least the minimum support, then finding rules with at least the minimum confidence. Apriori is able to reduce the field of candidate itemsets due to the property that any subset of a frequent itemset is itself frequent (Agrawal et al., 1993). However, even with this fact, it is understood that Apriori is not effective on large data sets (Pei et al., 2004).

A faster algorithm for mining frequent itemsets is FP-Growth. FP-Growth also relies on two-phases: the construction of a search tree data structure, and the mining of the structure (Han et al., 2000). The first phase, building the tree, is performed with the following algorithm:

1. Scan all transactions once and count single items, ranking the items in descending order by support (this sorted list is named L)
2. Create an FP-Tree. This begins by creating a *null* node, then:
   (a) For each transaction in the data set:
      i. Sort the items in the transaction according to the order of L
      ii. Note the first item in the new sorted list – this item is called the *prefix*
   (b) Identify transactions that share the same prefix – if transactions share

the same prefix, they are assigned to to the same stem node, branching from the null node

(c) Note the count of items remaining in each transaction and repeat the above proces for each child node

Once the tree is constructed, it is mined for rules. This process occurs as follows:

1. Identify nodes in the tree with the lowest support (these are called suffixes) – if the node has support lower than the minimum threshold, it is cut and the algorithm moves to the next lowest support items
2. Once candidate low-support items are found, follow the suffix node up the tree to the prefix node: this is considered a new sub-tree
3. Remove any suffix nodes or branches from the sub tree that do not meet the minimum confidence threshold
4. Each remaining path is a qualifying frequent itemset
5. Repeat for each prefix branch

FP-Growth is an improvement over Apriori in a few important respects. First, FP-Growth only scans the data set twice: once to identify qualifying singleton items, and again to organize the items in each transaction. This saves on processing time as processing time increases linearly with the number of transactions in FP-Growth, but exponentially in Apriori. Second, FP-Growth uses less memory, as candidate itemsets are stored in the FP-tree structure as contrasted with Aprioiri where each candidate set is stored in memory (Han et al., 2000).

Once all qualifying itemsets have been found, it is time to turn to interesting

measures to identify useful trends in the data.

2.3.2 Interestingness Measures

Thus far, this section has discussed in detail how to collect support and confidence of association rules, but deciding which rules are important is a different matter. Given any rule generated from a data set, how would one decide if that rule was more or less useful than others? Unfortunately, there is no single, agreed upon definition for what makes a rule *interesting*. However, there are several dimensions of *interestingness* that can alert an analyst to the quality of a rule. Among these are *conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility,* and *actionability*.

A concise rule is one that contains relatively few items. {Beer} $\rightarrow$ {Diapers} is an example of a concise rule. Such rules are easier to understand and remember, and are more simply added to a user's knowledge of the data (Padmanabhan and Tuzhilin, 2000).

Coverage refers to the universality of a rule. It is based in the notion that if a rule appears frequently (that is, with exceptionally high support), there is likely a hidden driving factor that is causing such a trend. Rules that contain stop words that are found using LDaRM are interesting due to their coverage – the rules appear frequently because stop words appear frequently topic models (Agrawal et al., 1993).

Reliable rules are rules in which the relationship is found with high frequency. Rules with high confidence would be reliable rules. A rule that is interesting due

to its reliability could be {Peanut Butter} → {Jelly} (Tan et al., 2004).

Peculiar rules are patterns that are generated from outlier data points. If an item rarely appears overall but when it does it frequently co-occurs within a particular itemset, it may be of interest to a user (Zhang et al., 2004). Finding peculiar rules is a driving factor for setting a lower support threshold – though this may increase computation time, it allows peculiar rules to rise to the surface. A canonical example of a peculiar rule is {caviar} → {vodka}.

A rule is diverse if the consequent and antecedent differ greatly in context. If, for some reason, {motorcycle} → {moon} were found to occur frequently in a data set, a user may be interested in why such disparate terms were highly associated (Hilderman and Hamilton, 2013).

Novelty is an interestingness measure of particular importance to ETA. A rule is considered novel if it presents a user with information of which they were previously unaware. Specifically with LDaRM, novelty is a critical measure for the contextualization of named entities, a process which will be discussed in greater detail in later chapters. It is impossible for a model to prove the absence of knowledge in its user – however, case studies have shown that novelty in rule mining can assist one in knowledge discovery tasks (Sahar, 1999).

Surprising rules contradict the existing knowledge of a user. A surprising pattern is interesting as it highlights gaps in one's understanding of the underlying data. Surprising patterns provide opportunities for a user to investigate this gap more completely (Liu et al., 1999).

The utility of a rule is a measure of how well it helps a user reach a certain goal.

If one were to perform ARM on real estate data to identify common features of expensive homes with the objective of finding under-priced units, they would be mining rules for utility (Yao and Hamilton, 2006).

Finally, a rule is actionable if it affects the user's decision making in a related domain. If a store owner placed beer and diapers next to each other in their store due to finding the {Beer} → {Diapers} rule, the rule would be considered actionable. In the context of LDaRM, rules can be actionable if they affect how a user performs a deeper analysis – for example, a user may be alerted to a novel name or location, which serves as the basis for the next steps of research.

The above concepts are broad categories of what may make a rule interesting. The next section will select several specific measures used to bring interesting rules to the surface.

2.3.3 Association Rules

Below are a set of calculations for how to derive values for various interestingness measures. The selected measures were chosen for LDaRM due to their ability to surface novel, surprising, reliable, concise, and ultimately actionable rules.

Lift is based on statistical independence, and is defined as:

$$L(I \to J) = \frac{C(I \to J)}{S(J)} \tag{2.13}$$

A lift of greater than 1 indicates that I and J have a positive association. Conversely, a lift less than 1 indicates that I and J inhibit each other (Leskovec et al., 2014).

This lift measure is distinct from Lift as a topic quality metric (Taddy, 2012). Lift is a powerful indicator of interesting rules as highly related itemsets will achieve large values. Terms that are statistically independent have a value of 1, which may be interesting as well. Negatively related terms are bounded as (0,1). Further, lift is a symmetric measure. That is, $L(I \rightarrow J) = L(J \rightarrow I)$

Conviction is an enhancement of the lift measure that considers the absence of J, defined as:

$$Con(I \rightarrow J) = \frac{1 - S(J)}{1 - C(I \rightarrow J)} \tag{2.14}$$

Conviction, unlike lift, is directional, highlighting the uniqueness of individual itemsets (such as I indicating a positive association with J, but not the opposite) (Brin et al., 1997). Conviction has a tendency to present misleading or false-positive results as it does not account for the support of both the antecedent and consequent. However this is not always a negative, as such a measure has a tendency to highlight novel rules.

Leverage describes the difference in I and J appearing together compared to what would be expected if I and J were independent, defined as:

$$Lev(I \rightarrow J) = S(I \rightarrow J) - S(I)S(J) \tag{2.15}$$

Leverage is similar to lift in that it expresses the confidence that itemsets are independent. However, where lift emphasizes low-support itemsets, leverage will identify items with high support (Piatetsky-Shapiro, 1991). That is, high support

itemsets may have low confidence and do not achieve high lift scores (representing items that are appear frequently overall but rarely with other items), but would be highlighted using leverage.

Correlation is analogous to Pearson's correlation coefficient, defined as:

$$\phi(I \to J) = \frac{Lev(I \to J)}{\sqrt{S(I)S(J)S(\overline{I})S(\overline{J})}} \tag{2.16}$$

Correlation is on a scale from [-1,1]. A correlation near 1 indicates that the appearance of I implies the appearance of J. A correlation near -1 indicates the presence of I implies the absence of J (Tan et al., 2004). Correlation is an excellent interestingness measure – it is a normalized form of leverage so it identifies high-support, low-confidence rules, but is scaled for support. Further, correlation is widely understood outside of ARM and is easy to explain to practitioners from outside disciplines.

Rule Power Factor can be interpreted as the weighted confidence of a rule, and is defined as:

$$rpf(I \to J) = S(I \to J)\frac{S(I \to J)}{S(I)} \tag{2.17}$$

RPF is used to identify rules with similar confidence but varied support (Ochin et al., 2016). It is designed as an improvement to lift, scaling for the number of transactions.

2.4 Related Works

LDaRM is far from the first attempt at improving the interpretability of topic models for ETA tasks (Sievert and Shirley, 2014). Typically, users prefer topic models with large numbers of topics. However, there exists a strong relationship between the number of topics and the likelihood that a topic will be deemed nonsensical by experts (Mimno et al., 2011). Many methods have been proposed to avoid such situations, including improved sampling methods (Li et al., 2017), outside information encoding (Krasnashchok and Jouili, 2018), and use of topic coherence measures (Lau et al., 2014). The approach with LDaRM leverages term prevalence and topic relationships to help a user navigate topics and in turn understand the underlying corpus.

Of particular interest is a topic modeling visualization platform, LDAvis (Sievert and Shirley, 2014). Topics are presented to users as lists of words – a topic model with hundreds of topics is difficult to navigate as raw text. LDAvis takes a topic model as an input and graphs the topics based on principal components (derived from the original TF-IDF matrix). A user is then able to interact with the model, clicking on each topic to see the top words. LDAvis is a major improvement for interacting with topic models over simply reading lists of words. However, LDAvis suffers from a few issues: first, principal components are abstract concepts and difficult to understand at a glance. Further, as represented through a software tool, one can only view two principal components at once. For a diverse corpus it is highly likely that there are more than three significant principal components.

Previous work has combined ARM and topic modeling through the Frequent

Pattern – Latent Dirichlet Allocation (FP-LDA) method (Dave et al., 2014). In the FP-LDA technique, the closest technique to LDaRM, a user performs ARM and applies topic modeling on the output, resulting in a set of topics. FP-LDA satisfied the authors' objective of identifying relevant code files for completing maintenance tasks. The approach in LDaRM does the reverse, first applying topic modeling to a corpus followed by ARM, resulting in a set of association rules. FP-LDA delivers a fixed set of documents from a corpus for a user to review, circumventing the need to interpret topics directly. LDaRM provides a set of related terms that one can use to identify critical similarities and differences between topics, serving as a springboard to deeper qualitative analysis.

3 Methodology

Chapter 3 will outline the LDaRM process. This will include an overview of data preprocessing, the steps required for performing LDA, the ARM techniques and interestingness measures chosen, and visualization techniques and rule selection metrics. All steps of LDaRM are presented at a high-level and will describe the parameters and hyper-parameters available for tuning models. This methodology will be employed for a case study in subsequent chapters.

3.1 LDaRM Pipeline

LDaRM (latent Dirichlet allocation – association rule mining) is a combination of two machine learning techniques with a human-in-the-loop component. While each of the two methods is a powerful tool, it is crucial to add an element of human judgement. Learning never happens in a single step – it is always iterative, a process of constantly revisiting what one believes and adjusting views when presented with new evidence. LDaRM encourages such learning through .

The first part of the method employs LDA to develop a set of topics. Following this, using the topics as transactions and the words as items, LDaRM uses association rule mining to uncover interesting rules. One can then explore these rules for knowledge discovery and as the basis for improving topic coherence through stop word identification. A high level overview of LDaRM can be seen below in Figure 3.1.

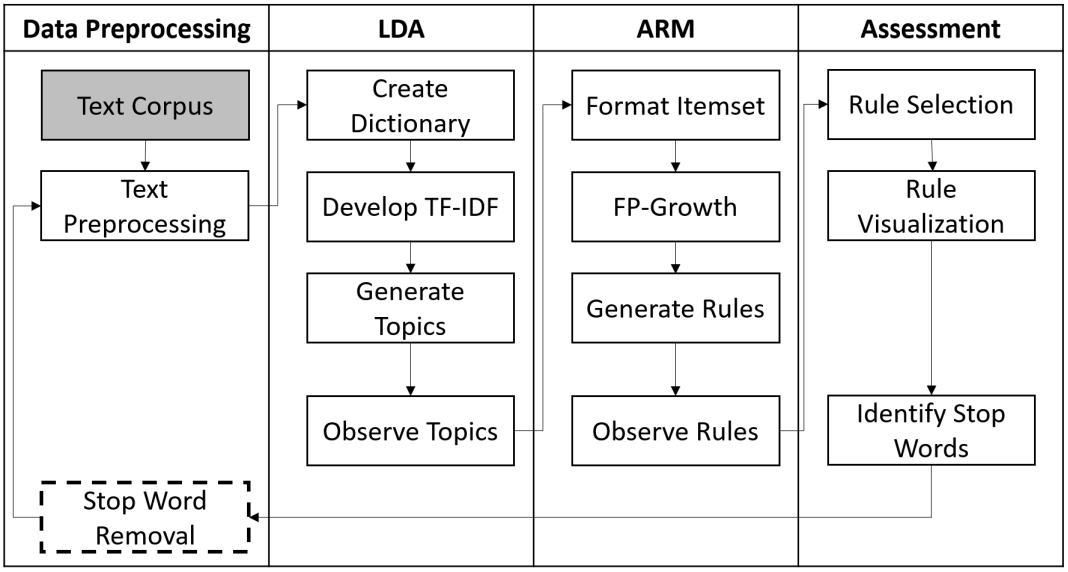

Figure 3.1: High Level Text Analytics Pipeline

Like most ETA tasks, LDaRM begins with a text preprocessing phase in which text features are extracted from a corpus and dimensionality is reduced. Following this, LDaRM develops a topic model using LDA, weighted by a TF-IDF. The resulting topics are analyzed using association rule mining, employing the FP-Growth algorithm to generate rules. Finally, the rules are presented both as lists and as interactive visualizations, from which one can engage in knowledge discovery.

It is important to note that topic modeling and association rule mining are both complex machine learning techniques with a variety of implementations. The version of LDaRM presented here uses LDA with Gibbs sampling, but any implementation of topic modeling is viable. Additionally, there are dozens of interestingness measures used in ARM – LDaRM employs a small subset of the total number of measures. Further, this methodology assumes that a user of LDaRM is performing the analysis using a high-level programming language such as Python, C#, or Java. The implementation of LDaRM for this analysis leverages Python and several associated libraries.

3.2 Data Preprocessing

3.2.1 Text Corpus

Data preprocessing begins with collecting digital editions of a text corpus. These can be in any digital format (.docx, .html, .xml, .pdf. .txt, etc.), so long as an analyst is able to convert documents into bags of words. While many applications and programming tools exist and can receive a variety of inputs, it is most common to convert the texts in a corpus into a sequence of unicode texts (Honnibal and Montani, 2017).

3.2.2 Text Preprocessing

Once the corpus is collected and converted to unicode texts, it should be tokenized and annotated. As LDA mines only sequence and structure, the annotations required for analysis are fairly limited. The process for annotating a corpus for LDA can be seen in Figure 3.2.

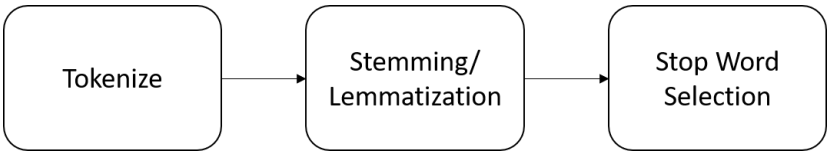

Figure 3.2: Text Preprocessing for LDA

The first step is tokenization – the process of slicing documents into individual n-grams. For topic models it is most common to select unigrams – n-grams that contain a single words. One can tokenize manually through use of regular expressions or with the assistance of more advanced libraries. For English, it is common to tokenize by white space characters such as spaces and new lines, as

well as by punctuation such as periods and underscores. More advanced tools implement heuristics for tokenization – for example, they would keep *Mr. Smith* as a single token rather than separating it into *mr* and *smith*. Tokens are then regularized – a process that converts all words to lowercase so tokens are not under-counted due to casing. This implementation of LDaRM uses the spaCy API to tokenize documents (Honnibal and Montani, 2017)

Once all documents have been tokenized, the tokens are stemmed, lemmatized, or both. Recall that stemming is typically done heuristically, while lemmatization is based on trained models. An individual analyst could implement their own stemmer or lemmatizer, but existing tools are more than adequate for LDaRM. When tokens are stemmed and lemmatized, the total number of unique tokens are reduced, reducing the computational time for running topic models. This implementation of LDaRM does not use a stemmer, but uses the Natural Language Toolkit's (NLTK) WordNet lemmatizer (Loper and Bird, 2002).

The remaining tokens are then labeled as stop words, if needed. There are a few common methods for removing stop words, such as removing words that are 2-characters or shorter or removing words that are on a preset stop word list. It is also common to remove words that appear in only one or two documents throughout the corpus. Recall that stop word selection is dependent on the corpus, and it is the responsibility of a user to identify corpus-specific stop words (LDaRM assists with stop word selection, a usage that will be discussed in greater detail in Chapter 5). As a baseline, this implementation of LDaRM removes all 2-character tokens and the standard NLTK stop word list (Loper and Bird, 2002). Following data preprocessing, the corpus is ready for LDA.

3.3 LDA

While one could program their own implementation of LDA, many APIs exist to automatically create topics. This implementation of LDaRM uses the Gensim API (Řehřek and Sojka, 2011). The overview of the steps required for LDA using Gensim can be seen in Figure 3.3.

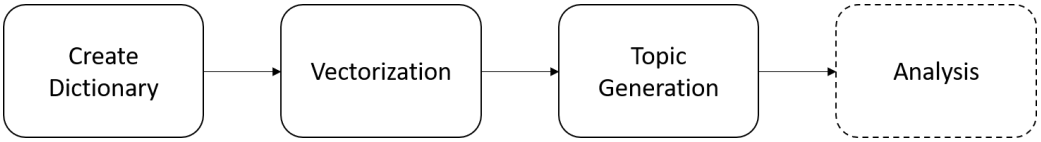

Figure 3.3: Steps of the LDA Process Using the Gensim API

The first step of any programmatic implementation of LDA is the creation of a dictionary. The dictionary forms the rows of the document-term matrix and consists of at least the tokens in the corpus. Some dictionaries can include terms that are not in the corpus of interest, but that may appear in future texts of interest. However, for this analysis, the dictionary consists only of the preprocessed tokens.

Once the dictionary is established, the corpus needs to be vectorized. Topic modeling as a whole relies on the bag-of-words model and the document-term matrix. As previously discussed, it may be beneficial to use TF-IDF to weight terms based on their relative appearance in a document compared to their appearance in the corpus as a whole. One can convert a document-term matrix into a TF-IDF with manual computation, but Gensim can perform this operation automatically.

The vectorized model can then be used to develop topics. This requires the user to select the desired number of topics and the number of iterations of Gibbs sampling.

The number of iterations should be sufficiently high for topics to converge – current research recommends a value greater than 20 passes but fewer than 200. This number is corpus dependant and one can tune the value to minimize the perplexity of the model (George and Doss, 2017). This implementation of LDaRM uses 50 iterations of the Gibbs sampler. Next, users should select the desired number of topics. This parameter is a matter of preference. Too few topics (two to 20 or so) and a user may not find enough interesting patterns. Too many (over 200), and the topics become too similar and difficult to navigate. A common number for a typical analysis ranges from 50 to 150 topics (Stevens et al., 2012). The subsequent analysis will use 75 topics

Once the topic model is complete and the topics have been generated, one should search for patterns. Each conclusion should serve as an avenue for further analysis. However, as previously stated, it can be difficult to find conclusions using a topic model with a large number of topics. Using ARM allows a user to more quickly identify interesting patterns.

3.4 ARM

As with LDA, ARM can be performed manually by programming the algorithms explicitly, but many libraries and applications exist that will perform the operations automatically. The steps needed to perform ARM in the LDaRM process are outlined in Figure 3.4.

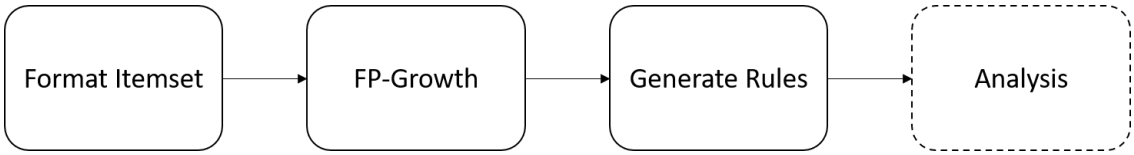

Figure 3.4: Steps for Association Rule Mining in LDaRM

ARM begins with the creation of a transaction list. In LDaRM, the topics generated by LDA act as the transactions, while the individual words act as items. As they are language models, a topic technically contains every word in the corpus, with some words having a negligible probability of appearing in the topic. As such, one must first determine a cut-off point for which words will be included in a topic. A good heuristic for ETA is choosing the 15 to 25 highest probability terms in a topic for analysis (Mabey, 2018). This implementation of LDaRM will use the first 20 terms from each topic. Therefore, the transaction list will consist of 75 transactions (each of the 75 topics), each with 20 items.

Following this, one must find high-support itemsets. There are many available algorithms, but LDaRM uses the FP-Growth algorithm – specifically the mlxtend implementation of the algorithm (Raschka et al., 2016). Recall that support is a measure of the frequency at which an itemset appears in all transactions – it can attain any value [0,1], with 0 meaning an itemset never appears, and 1 indicating every transaction contains the itemset. Therefore, one must select a minimum support value of interest. Choosing a lower value may be over-inclusive (every itemset will be included) but choosing too high of a value may exclude interesting rules. Given the relatively small number of transactions, this implementation of LDaRM will set a minimum support threshold of 0.02 – that is, every itemset that appears in at least two transactions will be included.

Once support is calculated, one will calculate confidence – an indication of how often itemsets co-occur. Confidence is a measure of conditional probability, measuring the probability of finding one a particular itemset given the presence of another. Again, one will set a minimum threshold – for this implementation, LDaRM sets this value at 0.1. With both the support and confidence of every rule, LDaRM is able to derive other interestingness measures, including lift, leverage, conviction, rule power factor, and Pearson correlation.

When all interestingness measures are calculated, one may be left with thousands of rules to sort through. To aid a user in the analysis of such rules, LDaRM presents several interactive visualizations to assist one with ETA tasks.

3.5 Assessment

With the association rules established, one must use the rules to identify interesting patterns. This process is made substantially easier through interactive visualization. While there is no fixed methodology for ETA, the high-level steps used in LDaRM are outlined in Figure 3.5 below:

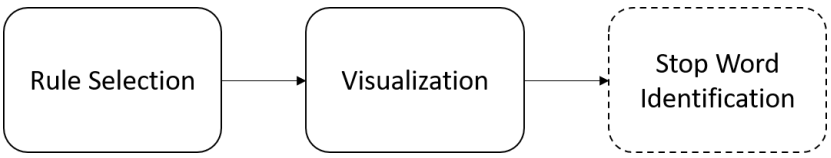

Figure 3.5: Steps for Association Rule Mining in LDaRM

A useful first step when using LDaRM is to observe the highest and lowest scoring rules for each interestingness measure. This will highlight major themes of the topic model – for example, a high lift rule would indicate that certain itemsets were dependent on each other. That is, a high lift rule indicates that certain words

occur together more than expected. This opens up new avenues of inquiry for *why* the itemsets occur together more than expected. A common conclusion from this analysis is the contextualization of named entities. LDaRM allows one to identify the words most strongly associated with certain entities. Furthermore, if there is ambiguity in the definition of certain terms, one can disambiguate by observing the rules. Analyses such as these will be expanded on in Chapter 5. While observing the top rules can be useful, it quickly becomes cumbersome. Using data visualization, in this case scatterplots, one can observe multiple rules at once, helping a user find interesting relationships with ease. By graphing two measures at once, one can more quickly observe rules. For example, a high-leverage, low-conviction rule would indicate a set of terms that do not appear together at an above-expected rate.

One of the strengths of LDaRM is its ability to identify stop words. Stop words, by definition, are likely to occur in a document over expectation, and many interestingness measures seek out such terms. One will notice stop words dominating many rules, and they will often stand out in visualizations. It is important to note suspected stop words, as it is useful to perform LDaRM multiple times using different sets of stop words. This process will improve topic coherence and allow one to perform more effective ETA. Examples of this process will be outlined in Chapter 5.

3.6 Repeat

LDaRM, as an ETA methodology, is inherently iterative. Only through a cycle of constantly updating prior assumptions, tuning the model accordingly, and

performing another round of analysis can one find interesting and useful patterns in the data. Performing LDaRM multiple times and removing stop words that confound the model for each iteration not only provides a slightly different result set, but also improves topic coherence.

Stop words, by nature, are high-interestingness. That is, they will appear in topics at a rate higher than expected (per Luhn's model of significant words). As such, stop words are likely to appear in rules with high lift or leverage. By looking through high-lift rules, one may note a term or several terms that appear frequently, but do not appear to provide much context. Examples of this selection process and its effect of LDaRM performance will be discussed further in Chapter 5.

It is important to note that the set of stop words is not fixed. If one were to remove one or more words in an early iteration but has evidence to suspect that a word is information bearing, it is reasonable to add the word or words back to the model in subsequent iterations. One can simply remove the terms from the stop word list, or roll back to the iteration before the terms were removed.

4 Data

Chapter 4 will discuss the data used for the subsequent case study. The data is a collection of research articles regarding COVID-19 and SARS-CoV-2. This chapter presents a brief description of COVID-19 and the related pandemic, as well as a description of the data used in the analysis. COVID-19 is a subject area of particular interest from a societal and research standpoint. COVID-19 has had a global impact as individuals have drastically altered their day-to-day behavior to prevent the spread of the disease. Research into treatments, prevention, and transmission is critical for helping society return to pre-COVID-19 routines. From a technical perspective, the COVID-19 data set is very useful for testing the effectiveness of LDaRM. The corpus is sufficiently large in that an individual could not possibly read all documents, and each document in the corpus is long enough to avoid short-text bias in LDA. Further, the corpus is focused enough in subject-matter to limit the range of possible conclusions (all conclusions will relate to COVID-19 or SARS-CoV-2), but diverse enough that one can identify unique insights (identifying risk-factors, vaccine research, the mechanisms of transmission, etc.).

4.1 COVID-19

The coronavirus disease 2019 (COVID-19) is an illness caused by the by a strain of coronavirus, severe acute respiratory syndrome coronavirus 2 (SARS-Cov-2). COVID-19 causes several symptoms, including fever, cough, fatigue, and loss of smell and taste (Organization et al., 2020). The disease carries a mortality rate

of roughly 3.7%, though this figure is still a subject of research (Mehta et al., 2020). The leading cause of death for individuals infected with COVID-19 is acute respiratory distress syndrome – a failure of the respiratory system due to inflammation.

On March 11, 2020, the World Health Organization declared COVID-19 a pandemic. On March 13, 2020, the United States declared a national emergency due to the outbreak (for Disease Control and Prevention, 2020). In response to this, the White House and several research groups assembled the COVID-19 Open Research Data Set (CORD-19).

4.2 COVID-19 Open Research Data Set

CORD-19 is a collection of over 167,000 scholarly articles related to COVID-19 and SARS-CoV-2. These articles were all converted to the .json format for analysis. The articles are sourced from multiple journals from many countries, written in several languages.

The purpose of CORD-19 is to crowd source insights into the large corpus. A few of the explicit tasks include identification of COVID-19 risk factors, knowledge of transmission and incubation periods, and tracking progress on a vaccine. A primary focus of the group is information distillation – creating useful summaries of the of data. (AI, 2020)

For the case study in chapter 5, only 5000 articles were used. The articles were randomly selected and are assumed to be a representation of the corpus as a whole.

5 Analysis

Chapter 5 presents a case study using LDaRM on the CORD-19 data set. This case study seeks to illustrate the strengths of the LDaRM methodology, including its value as a tool for ETA. The analysis will take the form of a walk through of the results of using LDaRM on the COVID-19 data outlined in Chapter 4. In particular this case study will emphasize LDaRM's ability to contextualize proper nouns and named entities. Further, this chapter will show how to use LDaRM for identifying stop words and demonstrate the subsequent improvements to topic coherence.

5.1 Overview of COVID-19 Analysis using LDaRM

Recall that for this analysis, each topic model contains 75 topics derived over 50 sampling iteration. For the association analysis, each of the 75 topics were treated as transactions, and the top 20 terms in each topic were considered the items of the transaction. Each topic model was then analyzed, emphasizing potential ETA conclusions. A key aspect of this analysis is the identification of stop words. The process of stop words identification is highly subjective, but this analysis will provide evidence for effective removal to improve coherence. The act of modeling, exploring the data, removing stop words, and modeling again reflects the ETA process through the Hermeneutic circle.

5.2 Exploratory Text Analytics

5.2.1 Iteration 0

In the zeroth iteration, the topic model was developed without removing any stop words. This leaves non-word tokens in the corpus (chiefly punctuation). It is rare that one would begin a topic model-based analysis without removing basic stop words as it is generally agreed that stop words confound model interpretability and inhibit model performance. In this situation, the analysis was performed to emphasize the stop word removal capabilities of LDaRM.

For the purpose of demonstration, however, iteration 0 will be briefly explored. As a brief overview, it is useful to look at the highest and lowest scores across many interestingness measures. Observe Table 5.1 below. Organizing by the Lift measure, one can see that that the rule {cell, patient, infection} $\rightarrow$ {data, et, protein, case, al, virus} indicates that the presence of the antecedent is highly predictive of the presence of the consequent terms. This point is further emphasized by the correlation coefficient of 1. Such a rule provides high interestingness values, but is likely not useful – one with a passing knowledge of SARS-CoV-2 would know that it is a virus that infects patients, and all terms are likely to appear in related research.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	RPF	corr_1
cell, patient, infection	data, et, protein, case, al, virus	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000
infection, et, case, [, cell, virus	protein, patient,], al	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000
infection, et, [, cell, al, virus	case, protein, patient,]	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000

Table 5.1: High Lift Rules from LDaRM Iteration 0

With over 300,000 rules in this iteration, it is helpful to turn to visualization to identify interesting rules. Consider Figure 5.1 below, which explores the relationship between correlation and leverage as well as correlation with Rule Power Factor. The highlighted rule in the top image, {]} → {[} stands apart from the larger cluster of rules. This indicates that the presence of a closing bracket is highly indicative of the presence of an open bracket. Again, this rule is not very useful as it is probably obvious to any analyst.

Finally, there are many rules that are negatively correlated. In the bottom image of Figure 5.1, there is a rule {et, al} → {[,]}. This rule is slightly more interesting – it notes a difference in in-text citation style. That is, some of the papers used a named citation style (e.g., Smith et al.), while others use numbered citation (e.g., [1]). Such a rule may cause an analyst to question why there are multiple citation styles. This line of questioning could lead the analyst to conclude that the corpus is a set of articles from diverse journals, which may quell an assertion that the articles are in some way biased due to single-sourcing from one set of documents. While such an assertion is not *proven* by the presence of this rule, it does provide an analyst with a useful starting point.

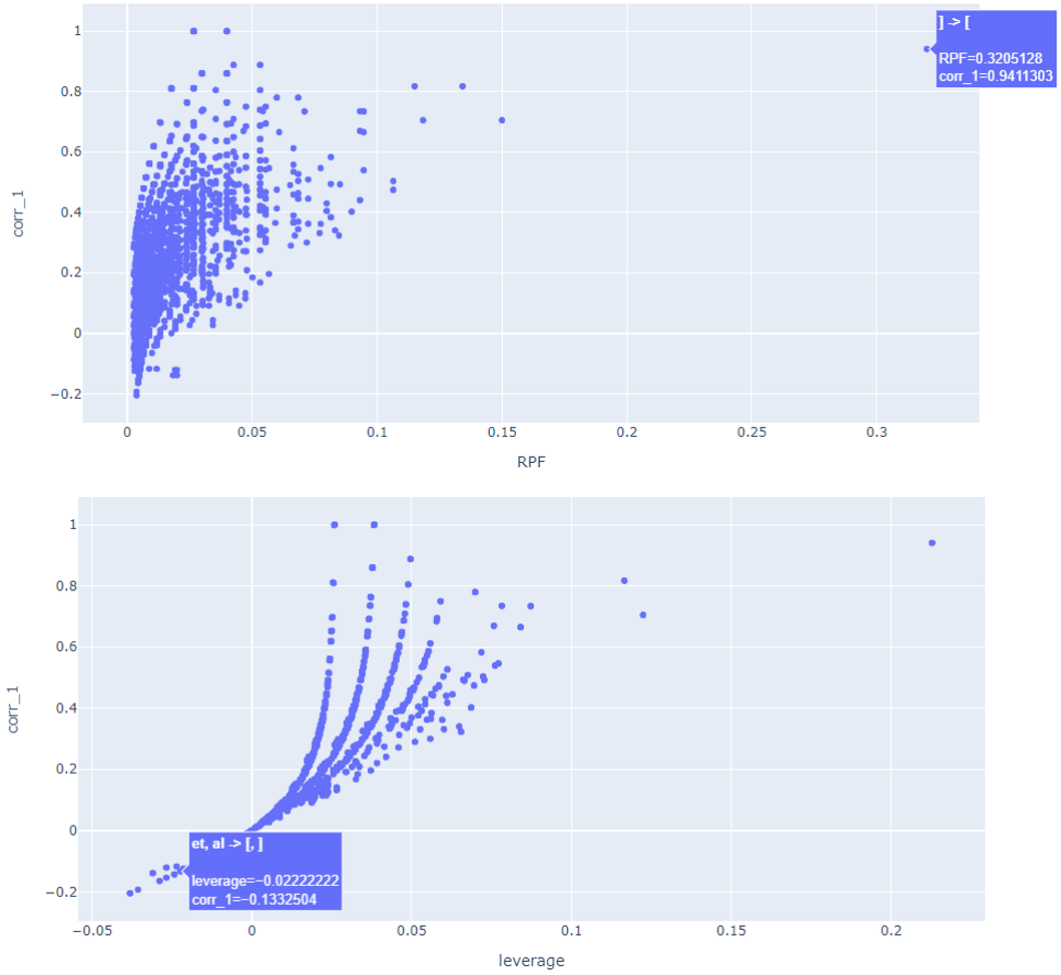

Figure 5.1: Selected Visualizations for Iteration 0

One may note in Figure 5.1, and in subsequent graphs, that the observations appear stratified – that is, the observations appear as lines, curves, or in fixed intervals. These patterns arise for two reasons. First, all interestinginess measures are functions of support and confidence – each axis has a definable relationship to the other. Second, there are fewer support and confidence (and therefore, other interestingness measures) than one may expect. Using the 75 topics from these examples, the highest support item is roughly 25%, appearing in 19 topics. Given this, the maximum number of discrete confidence values cannot exceed 210, and is actually far fewer. These factor combined lead to the patterns in graphs.

5.2.2 Iteration 1

Iteration one is the first analysis to include a base set of stop words. This version uses the heuristic of removing all terms with two or fewer characters, as well as removing the stop words from the NLTK English stop word set. This means, upon running the model, one will notice that punctuation is no longer present and all tokens are words.

Beginning the analysis, again, it is useful to look at the highest and lowest scoring rules first. Observe Table 5.2 and note the inhibiting effect of the rules {model} $\rightarrow$ {patient} as well as {virus} $\rightarrow$ {case}. These rules have negative correlation values, highlighting that the presence of the antecedents ("model" and "virus") indicates the absence of the consequent terms ("patient" and "case"). An analyst may find these rules interesting and actionable – one now has reason to believe there are two frameworks of study regarding COVID-19: a model based approach and a case-study approach that examines infected patients. This, as with all ETA, requires more evidence to prove, but may provide an interesting feature to include in subsequent analysis.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	RPF	corr_1
model	patient	0.160000	0.253333	0.026667	0.166667	0.657895	-0.013867	0.896000	0.004444	-0.086969
patient	model	0.253333	0.160000	0.026667	0.105263	0.657895	-0.013867	0.938824	0.002807	-0.086969
activity	infection	0.120000	0.293333	0.026667	0.222222	0.757576	-0.008533	0.908571	0.005926	-0.057676
virus	case	0.280000	0.226667	0.053333	0.190476	0.840336	-0.010133	0.955294	0.010159	-0.053905
case	virus	0.226667	0.280000	0.053333	0.235294	0.840336	-0.010133	0.941538	0.012549	-0.053905
viral	patient	0.133333	0.253333	0.026667	0.200000	0.789474	-0.007111	0.933333	0.005333	-0.048099

Table 5.2: Negative Correlation Rules from LDaRM Iteration 1

The analysis then turns to stop word identification. As previously described, stop

words are likely to appear in multiple topics, confounding interpretability. From an association mining perspective, stop words are likely to have high lift and correlation values. Observing such rules, we see the results as in Table 5.3.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	RPF	corr_1
group, protein, patient, virus	level, model, response, data	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000
cell, viral, data	sars, infection, protein, patient, infected, h...	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000
level, patient, model	data, infection, human, protein, case, group, ...	0.026667	0.026667	0.026667	1.000000	37.500000	0.025956	inf	0.026667	1.000000

Table 5.3: Likely Stop Words as Identified in LDaRM Iteration 1

As in iteration zero, one can see words like “virus”, “viral”, “protein”, “group”, and “patient” occur together frequently. Observe the relatively large sets of words in both the antecedent and consequent itemsets. One may note these longer itemsets and assume that there are two or more fairly homogeneous topics – that is, topics with a high-number of the same terms. Furthermore, while these words are typically information bearing, in a corpus as focused as the COVID-19 data set the terms are actually confounding factors. The entire set of stop words selected are available in Appendix A1.1.

5.2.3 Iteration 2

Iteration two is the first of the topic models that has targeted, domain-specific stop word removal. Removing stop words allows the topic model to converge on significant terms, and therefore provide more latent semantic content.

For example, in Figure 5.2, in the top graph comparing RPF to correlation, we

see a high-correlation, high-RPF rule {cytokine, increased} → {inflammatory, expression}. These rules appear to be interesting, useful, and actionable and open a vast area for ETA. Based on this rule, one may ask the questions "what is cytokine, what does it mean for cytokine to increase, and what is cytokine's relationship to inflammation?"

Through only a fraction of the corpus discusses cytokine, this rule directs an analyst and guides their in-depth analysis. The papers that discuss cytokine reveal the mechanism by which COVID-19 causes distress, via events called "cytokine storms", or more formally "Cytokine release syndrome". Cytokine storms cause rapid inflammation and are the main life-threatening events spurred on by COVID-19 (Coperchini et al., 2020).

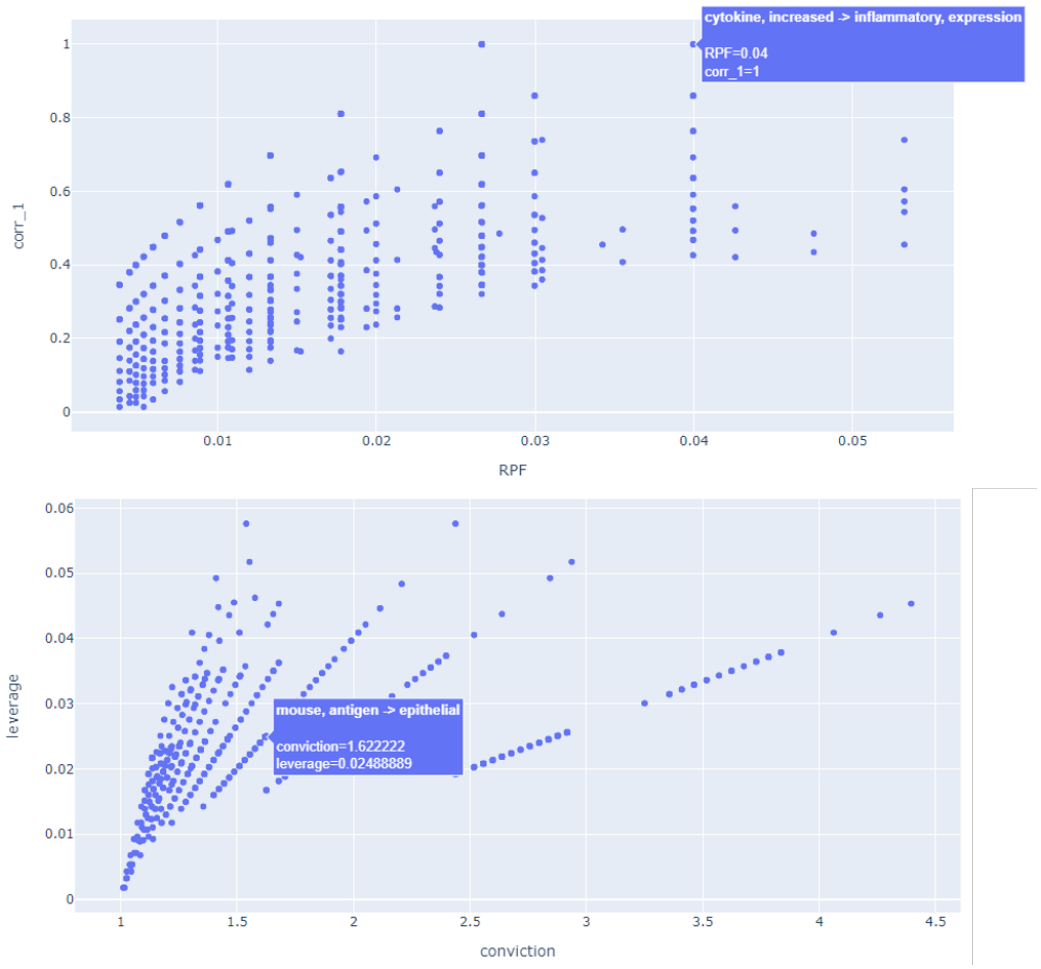

Figure 5.2: Selected Visualizations for Iteration 1

In the lower graph in Figure 5.2, one sees the rule {mouse, antigen} → {epithelial}. This rule contextualizes a segment of the COVID-19 research, which uses the cells of mice to identify antigens. Upon further analysis, one would find that such research is investigating the effectiveness of COVID-19 antibody tests (Long et al., 2020).

While these rules are interesting and provide an excellent starting point for analysis, one may note several stop words still confound the model. The set removed for iteration three can be found in Appendix A1.1.

5.2.4 Iterations 3 - 7

Iterations three through seven are much of the same as iteration two: perform LDA using the new set of target stop words, generate interestingness measure, observe measures searching for actionable content, and note any terms that are not information bearing for potential removal as stop words. Further interesting examples of possible insights include the Figure 5.3, which reveal the research into animal-to-human transmission of the COVID-19, and that vaccine testing has begun animal trials.

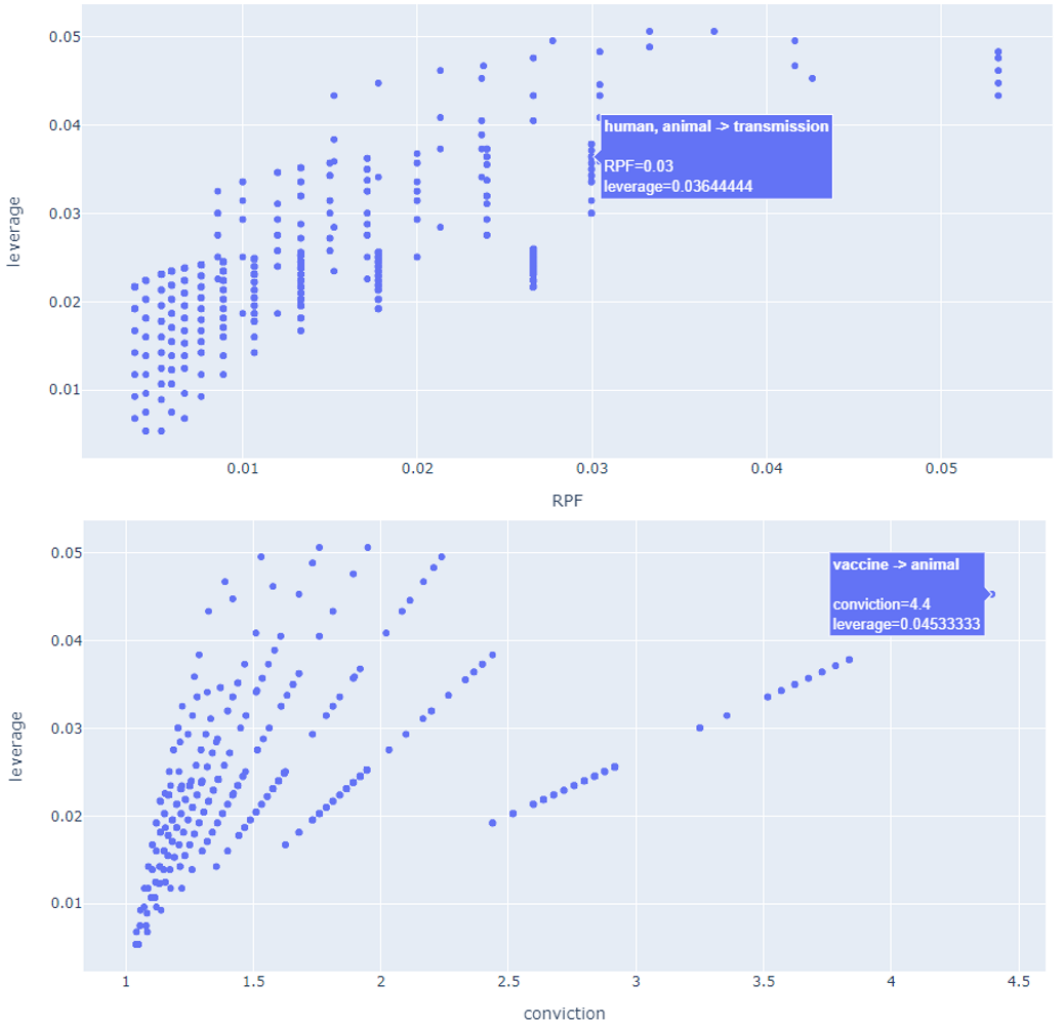

Figure 5.3: Selected Visualizations for Iteration 3

In Figure 5.4, we see a high-coverage rule – a rule that one would expect to occur given prior knowledge, or one that otherwise confirms a known pattern. A user may already know or expect this rule, but it illustrates an interesting point. {sars, coronavirus} → {respiratory} is an example of a rule that provides descriptors for a named entity. LDaRM's ability to contextualize named entities at a glance is a particular strength of the method. LDA alone isolates named entities by default, as language models have a bias for proper nouns. ARM is excellent for finding interesting relationships between items in a set. However, topic modeling alone often provides muddy or difficult to interpret topics regarding named entities (as measured by topic coherence). ARM alone is very computationally intensive, especially when calculating support and confidence for thousands of transactions (documents), each with thousands of items (words). LDaRM leverages the strengths of each technique to bring critical contextual information about named entities to the surface.

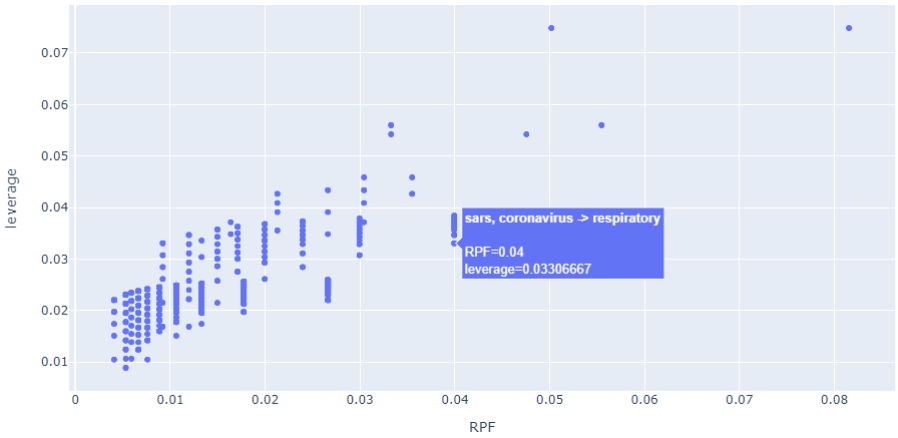

Figure 5.4: Selected Visualizations for Iteration 5

5.2.5 Contextualization of Named Entities

Named entity recognition is a text modeling technique used to identify named entities – words in a corpus that refer to a specific proper noun. Named entities are a difficult problem in text analytics – named entities may not be general knowledge or may be named after a regular noun (Apple Inc., usually referred to by the moniker Apple, is one such example). This makes collecting training data a resource intensive task. State-of-the-art machine learning algorithms mine symbol and sequence through use of recurrent neural networks and long-short term memory networks. Such algorithms consider the context in which a word appears (for example, in English, capitalized words in the middle of a sentence are more likely to be proper nouns) (Lample et al., 2016). LDaRM does not seek to identify named entities – rather, it provides additional context for named entities. This feature is explored in greater detail in the following Discussion section.

5.2.6 Discussion

Consider the rule in Table 5.4. Here, one sees an association between the words ace2, receptor, expression, and activity. Imagine an analyst who has a passing knowledge of the COVID-19 data set, but is not an expert. ACE2 is certainly not a commonly known word, at least to the extent that a general-readership magazine published an article breaking down the concept for a broad audience (Molteni, 2020). This presents an opportunity to use LDaRM to contextualize a named entity.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	RPF	corr_1
ace2, receptor, expression	activity	0.026667	0.173333	0.026667	1.000000	5.769231	0.022044	inf	0.026667	0.361475

Table 5.4: Important Features of ACE2

Using the information from LDaRM alone, it is known that ACE2, receptor, expression, and activity are highly related terms based on the relatively large lift score (indicating that the terms have a positive association). Without any external information, one can make a fairly educated guess about ACE2 based on the context provided – it is likely related to proteins (receptor and expression, which appeared with cytokine in Figure 5.2), that is behaving differently under COVID-19 (given the data set and the term "activity"). One can then dig into the corpus for additional information. One way to accomplish the deeper dive is by searching the corpus using keyword search (using terms of interest as the keywords). However, unless the corpus is indexed, this search can be a lengthy process. It may be helpful to search only document titles, abstracts, or other levels of the document hierarchy for the keyword to identify relevant texts. Another way to find documents related to terms of interest is by keyword searching the topics in the model, then using the document-topic matrix to find which texts are comprised primarily of the identified topics.

Upon further investigation, ACE2 is a surface protein on many cells in the human body ("ACE2" rather than "protein" is what qualifies the enzyme as a named entity). In fact, ACE2 is the surface protein to which SARS-CoV-2 binds. Individuals who suffer severe symptoms due to COVID-19 tend to have increased ACE2 activity, allowing a greater viral load to bond to cells in the respiratory tract (South et al.,

2020).

Another example of named entity contextualization can be seen in Figure 5.5. This analysis again turn to cytokine in the rule {cytokine} → {inflammatory, il-6, blood}. Recall that cytokines are a class of proteins that cause distress and inflammation. It is a safe assumption that il-6 has a similar relationship – that is, it is a protein or enzyme related to immune response. Upon reviewing the corpus, one will find that IL-6 is a specific type of cytokine (and therefore a named entity). IL-6 is of particular interest to researchers who are investigating treatments for COVID-19, as current technology exists to block IL-6 receptors, and may serve to reduce symptom expression.

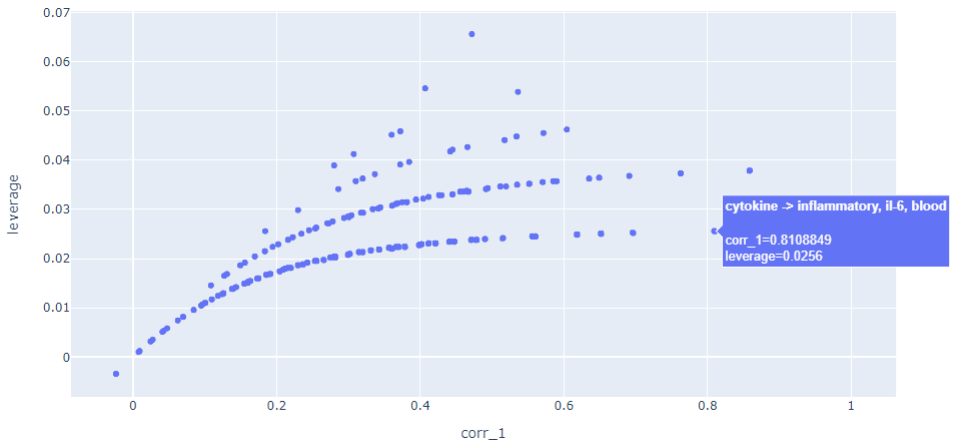

Figure 5.5: Association Rule to Contextualize IL-6

5.2.7 Stop Word Selection and Topic Coherence Improvement

As previously mentioned, LDaRM is an excellent tool for removing stop words. One can see, qualitatively, that stop word removal over several iterations led to interesting knowledge discovery. However, this notion can also be shown

quantitatively. This assessment returns to topic coherence, in particular the coherence measure *UMass* – an intrinsic measure of how "good" a domain expert would deem a topic (Mimno et al., 2011). UMass was chosen over other coherence measures such as UCI due to its reliance on word-order within each topic. Because LDaRM selects only the top words from a topic, the stop words found by using this method rely on term-order. It is appropriate, then, to use a measure which considers term-rank when measuring coherence (Röder et al., 2015).

While topic coherence is not a perfect measure, it is generally agreed that coherence correlates with human understanding of topics. For the implementation of UMass used in this analysis, results are presented as negative numbers, with larger magnitudes representing better topics (the more negative, the better). To reduce confusion, all UMass values have been adjusted using absolute value. Further, the values presented are the average coherence of the entire model (Stevens et al., 2012).

While there is no unified model for what constitutes "good" and "bad" topics, there are heuristic thresholds (Stevens et al., 2012). An average UMass of 1.0-1.2 represents a very poor model, perhaps incomprehensible. A topic model would likely only average around this score if there were too few topics in the model or if the number of sampling cycles were too low. A UMass of around 1.2-1.5 would represent a low-quality topic model. A UMass of 1.5-1.8 represents a moderate-quality model, while models from 1.8 and above are high-quality models. It is uncommon for the average UMass of a topic model using LDA to exceed 2.1, though scores as high as 2.5 are achievable.

Using LDaRM, this case study showed that it is possible to take a topic model with low to moderate-quality topics, remove confounding terms, and convert the model into one with high-quality topics. Figure 5.6 shows the progress of topic coherence for each iteration. Over the eight iterations in the case study, the average UMass coherence score was recorded. In the zeroth iteration, with no stop words removed, the model would be rated as a low-quality topic model. Removing the base set of over 100 stop words, one can see a solid increase in topic quality, though the model is still fairly poor. Once stop words are selected using LDaRM, one can see rapid improvement. By removing around a dozen terms identified using LDaRM, there is a jump from a low-quality model to a high-quality model in just six iterations. The maximum UMass achieved, 2.059, would be rated as a high-quality model.

While there is no set rule for when to end an analysis, a few heuristics can assist a user. First, one should continue iterating with LDaRM, removing new stop words with each pass, until the model achieves a "good" topic coherence score. From this point, if one does not see major improvements to topic coherence (more than a few hundredths of a point) for three consecutive iterations, it is safe to stop. Additionally, if there is a decrease in topic coherence after an iteration, one should stop the analysis or re-run the model with a different set of stop words.

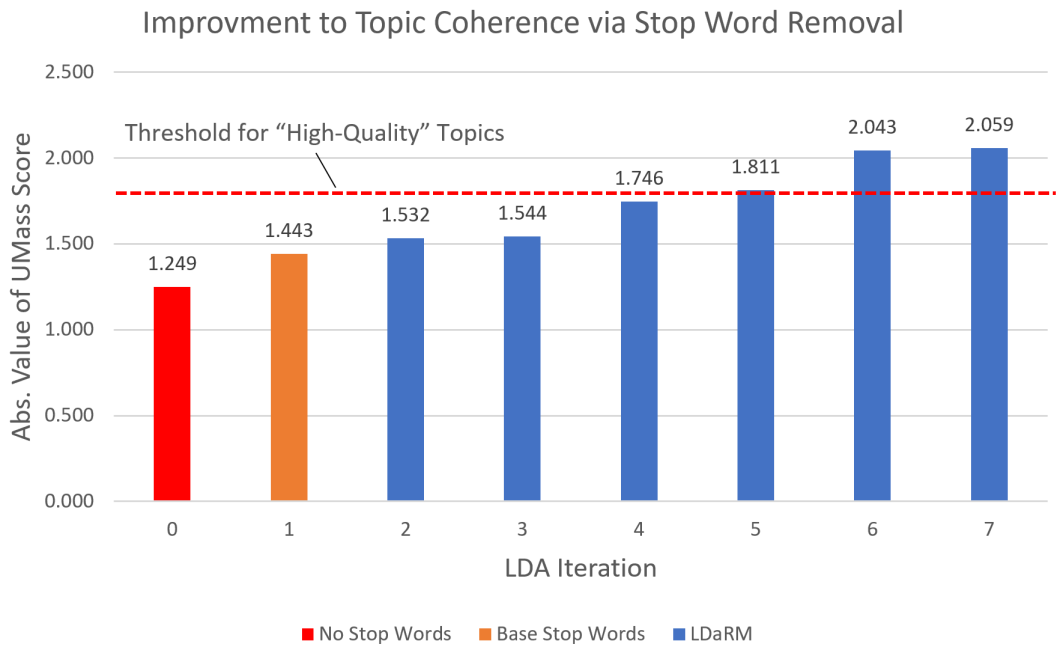

Figure 5.6: Improvements to Average Coherence Through the Removal of Stop Words

6 Limitations and Future Work

6.1 Limitations

LDaRM is an excellent tool for identifying interesting rules that provide insight into a corpus, especially concise, novel, surprising, reliable, and actionable insights. However, LDaRM is still subject to some of the constraints of its component processes. For example, LDA is fairly computationally expensive, especially as the number of documents in a corpus and the number of topics grows. A common training data set for testing topic models is the PUBMED set, which contains 8.2 million documents. The memory required to generate 10,000 topics and store all LDA parameters exceeds 36 TB. On a personal computer with a single processor, the time required to complete this process would be on the scale of weeks (Wang et al., 2014). While it would be extremely difficult to assess such a model for ETA (navigating 10,000 topics is no easy task), LDaRM does not avoid the time and memory constraint of LDA.

It is also difficult to reproduce results when using Gibbs sampling, as it is a generative process. To get the same results in two runs of LDA, one must set the random seed (this is, in fact, a parameter that is included in most LDA software packages). Due to this, it is said that topic modeling results are *unreliable*, making it difficult to assign value to findings with a high degree of certainty (Rieger et al., 2020). It is possible that some iterations of LDA will not converge on major latent themes in a corpus, and a user of the model would have no way of knowing for sure if this were the case. LDaRM, as a tool for gathering insights from a topic

model, has the same issue.

Association rule mining suffers several shortcomings as well. One major issue of note is the lack of testing to confirm if a rule is a true-positive. That is, if association rule mining identifies 10,000 rules, the odds are high that some number of these rules occurred by chance and do not represent an interesting relationship. Currently there is no effective method for assessing the statistical significance of a rule.

Additionally, ARM algorithms do not incorporate weighted inputs. All transactions and items are treated with equal importance. For example, ARM does not accommodate transactions that contain multiple copies of an object (say, a customer buys 5 oranges – *oranges* are treated as a single item). For weighted transactions, perhaps an organization deems customers from their target demographic as more important than non-target customers (Tao et al., 2003). If one were to perform LDaRM on customer reviews, topics that include certain products by name would almost certainly be of greater importance than topics that that do not include product names or are simply complaints.

Further, ARM can generate a high number of rules, perhaps in the tens of thousands. Even with interestingness measures it can quickly become difficult to navigate all results and isolate useful and actionable results. One can raise the support and confidence thresholds, but it has been shown that rules can be interesting *because* of their low support or confidence values. LDaRM does seek to improve user experience through use of visualizations and several diverse interestingness measures. However, identifying techniques to cull large rule sets even further would

improve the usability of LDaRM

6.2 Future Work

A potential improvement to LDaRM would include a "memory" mechanism that allows a user to return to a past iteration using different sets of stop words. If one identifies a stop word in an early iteration but notes several iterations later that the term should have been included, the individual can add the term back to the bag-of-words. A future version of LDaRM could use topic coherence to alert users to potential terms that should be added back to the model. By comparing topic coherence in the base topic model to topic coherence if the stop word were added to the set (by simply adding the term to topics), LDaRM could identify terms that may be worth returning to the bag-of-words in subsequent iterations.

A promising area of future research related to LDaRM is the use of association rule mining for stop word selection. This research has shown that using LDaRM to identify stop words can improve topic coherence. This aspect of LDaRM could be greatly expanded. The objective of this research would be to identify a stronger heuristic for selecting stop words and showing that the method improves topic coherence more than removing random terms. This method would have to be compared to other stop word selection techniques using a standard corpus such as the PUBMED database (Amarasinghe et al., 2015). This research would also need to ensure the selection process is robust against the number of topics in a model and the number of iterations of the sampler. It would also be important to explore how LDaRM affects other topic coherence measures beyond UMass.

Additionally, LDaRM presents an interesting opportunity to explore weighted itemsets for association rule mining. As language models, the words in each topic have an associated probability of occurrence - these could act as natural weights for the items. The weights are already normalized for the length of documents and total number of occurrences. Using weighted items would require a new mining method, as Apriori and FP-Growth are based on raw counts. However, if this issue could be resolved, weighted items would provide an additional dimension for identifying interesting rules.

7 Conclusion

7.1 Summary

Exploratory text analytics is a sub-field of text mining that focuses on uncovering latent semantic content from a set of documents using statistical modeling. Much text mining typically employs supervised learning to identify specific conclusions such as classification or prediction. ETA is less rigid, typically employing unsupervised techniques and alerting a user to interesting patterns that can spur a deeper analysis.

This research introduces LDaRM – an exploratory text analytic technique which combines two data mining techniques, topic modeling via latent Dirichlet allocation and association rule mining. Each technique taken alone is a powerful tool for data mining, in particular text mining. Topic modeling is a powerful tool for uncovering latent content, but it is often difficult for users to navigate models with a large number of topics. Association rule mining provides concise rules, ranked by a variety of interestingness measures. However, the algorithms used to derive these rules are not powerful enough to mine thousands of documents with tens of thousands of words. LDaRM leverages the strengths each technique to surface interesting, useful, and actionable insights. Broadly, LDaRM develops a topic model using the corpus of interest and performs association rule mining on the results of the topic model. The technique uncovers interesting relationships between and within topics – specifically, it can contextualize proper nouns by presenting the term of interest alongside many highly associated terms. Additionally, LDaRM is a

powerful tool for identifying stop words which confound topic models and provide little insight into the meaning of a corpus.

This research used a case study of a set of medical journal articles related to COVID-19 to highlight the impacts of LDaRM. Across several iterations of LDaRM, the study uncovered interesting patterns in the data and identified many of the major mechanisms through which COVID-19 causes illness. Further, the study identified named entities in the form of specific proteins. Along with these proteins, LDaRM identified key features of these proteins to quickly provide context to an analyst.

Finally, LDaRM is particularly useful for identifying stop words that can confound topic models – that is, stop words decrease the quality of topics. While there is no unified set of stop words, it is known that removing stop words can improve topic coherence measures. Outside of a set of about 100 terms, stop word selection is very ad hoc and up to the discretion of the modeler. However, as previously stated, as the number of topics grows it becomes more difficult for one to navigate each topic to identify confound terms. LDaRM, through its interestingness measures, highlights likely stop words for a user at a glance. The case study showed an improvement to topic coherence over several iterations, ultimately taking the model from “low-quality” to “high-quality” topics.

References

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *Acm Sigmod Record*, 22(2):207–216.

AI, A. I. F. (2020). Covid-19 open research dataset challenge (cord-19).

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

Aljumily, R. (2015). Hierarchical and non-hierarchical linear and non-linear clustering methods to "shakespeare authorship question". *Social Sciences*, 4(3):758–799.

Alvarado, R. (2020a). Text models.

Alvarado, R. (2020b). Topic models.

Amarasinghe, K., Manic, M., and Hruska, R. (2015). Optimal stop word selection for text mining in critical infrastructure domain. In *2015 Resilience Week (RWS)*, pages 1–6. IEEE.

Azad, H. K. and Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Bischof, J. and Airoldi, E. M. (2012). Summarizing topical content with word frequency and exclusivity. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 201–208.

Blei, D. and Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems*, 18:147.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Boell, S. K. and Cecez-Kecmanovic, D. (2010). Literature reviews and the hermeneutic circle. *Australian Academic & Research Libraries*, 41(2):129–144.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *ACM Sigmod Record*, 26(2):255–264.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 288–296.

Chelba, C., Acero, A., and Mahajan, M. (2008). Discriminative training of language models for text and speech classification. US Patent 7,379,867.

Choo, J., Lee, C., Reddy, C. K., and Park, H. (2013). Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992–2001.

Clement, R. and Sharp, D. (2003). Ngram and bayesian classification of documents for topic and authorship. *Literary and linguistic computing*, 18(4):423–447.

Coperchini, F., Chiovato, L., Croce, L., Magri, F., and Rotondi, M. (2020). The cytokine storm in covid-19: an overview of the involvement of the chemokine/chemokine-receptor system. *Cytokine & Growth Factor Reviews*.

Daniilidis, K., Maragos, P., and Paragios, N. (2010). *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*, volume 6315. Springer.

Dave, N., Potts, K., Dinh, V., and Asuncion, H. U. (2014). Combining association mining with topic modeling to discover more file relationships. *International Journal on Advances in Software*, 7(3&4).

de Kok, D. and Harm, B. (2010). Natural language processing for the working programmer.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88.

for Disease Control, C. and Prevention (2020). New icd-10-cm code for the 2019 novel coronavirus (covid-19).

Fowler, A., Partridge, K., Chelba, C., Bi, X., Ouyang, T., and Zhai, S. (2015). Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 649–658.

Gao, J., Jockers, M. L., Laudun, J., and Tangherlini, T. (2016). A multiscale theory for the dynamical evolution of sentiment in novels. In *2016 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC)*, pages 1–4. IEEE.

Geng, L. and Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9.

George, C. P. and Doss, H. (2017). Principled selection of hyperparameters in the latent dirichlet allocation model. *Journal of Machine Learning Research*, 18:162–200.

Google (2012). Google ngram viewer. http://books.google.com/ngrams/datasets.

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

Hahn, U., Buyko, E., Tomanek, K., Piao, S. S., McNaught, J., Tsuruoka, Y., and Ananiadou, S. (2007). An annotation type system for a data-driven nlp pipeline. In *Proceedings of the Linguistic Annotation Workshop*, pages 33–40.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *ACM Sigmod Record*, volume 29, pages 1–12. ACM.

Hannigan, T. R., Haans, R. F., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., Kaplan, S., and Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, 13(2):586–632.

Hilderman, R. J. and Hamilton, H. J. (2013). *Knowledge discovery and measures of interest*, volume 638. Springer Science & Business Media.

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

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hotho, A., Nürnberger, A., and Paaß, G. (2005). A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62. Citeseer.

Hu, X. and Liu, H. (2012). Text analytics in social media. In *Mining text data*, pages 385–414. Springer.

Hu, Y., Boyd-Graber, J., Satinoff, B., and Smith, A. (2014). Interactive topic modeling. *Machine learning*, 95(3):423–469.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., and Zhao, L. (2019). Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kaur, M. and Sapra, R. (2013). Classification of patents by using the text mining approach based on pca and logistics. *International Journal of Engineering and Advanced Technology*, 2(4):711–714.

Kintsch, E., Steinhart, D., Stahl, G., LSA Research Group, L. R. G., Matthews, C., and Lamb, R. (2000). Developing summarization skills through the use of lsa-based feedback. *Interactive learning environments*, 8(2):87–109.

Klausen, S. H. (2017). Levels of literary meaning. *Philosophy and Literature*, 41(1):70–90.

Kolekar, M. H., Palaniappan, K., Sengupta, S., and Seetharaman, G. (2009). Semantic concept mining based on hierarchical event detection for soccer video indexing. *Journal of multimedia*, 4(5).

Krasnashchok, K. and Jouili, S. (2018). Improving topic quality by promoting named entities in topic modeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 247–253.

Kumar, V., Smith-Renner, A., Findlater, L., Seppi, K., and Boyd-Graber, J. (2019). Why didn't you listen to me? comparing user control of human-in-the-loop topic models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 6323–6330, Florence, Italy. The Association for Computational Linguistics.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lau, J. H., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Leskovec, J., Rajaraman, A., and Ullman, J. D. (2014). *Mining of Massive Datasets*. Cambridge University press.

Lezius, W., Rapp, R., and Wettler, M. (1998). A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for german. *arXiv preprint cs/9809050*.

Li, C., Duan, Y., Wang, H., Zhang, Z., Sun, A., and Ma, Z. (2017). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems (TOIS)*, 36(2):11–42.

Lin, Y., Michel, J.-B., Aiden, E. L., Orwant, J., Brockman, W., and Petrov, S. (2012). Syntactic annotations for the google books ngram corpus.

Liu, B., Hsu, W., Mun, L.-F., and Lee, H.-Y. (1999). Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):817–832.

Liu, G., Zhang, H., and Wong, L. (2011). Controlling false positives in association rule mining. *arXiv preprint arXiv:1110.6652*.

Long, Q.-X., Liu, B.-Z., Deng, H.-J., Wu, G.-C., Deng, K., Chen, Y.-K., Liao, P., Qiu, J.-F., Lin, Y., Cai, X.-F., et al. (2020). Antibody responses to sars-cov-2 in patients with covid-19. *Nature medicine*, pages 1–4.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Luhn, H. P. (1960). Key word-in-context index for technical literature (kwic index). *American Documentation*, 11(4):288–295.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Mabey, B. (2018). pyldavis: Python library for interactive topic model visualization. *Port of the R LDAvis package*.

Manning, C. D., Raghavan, P., and Schutze, H. (2009). Introduction to information retrieval.

Mehta, P., McAuley, D. F., Brown, M., Sanchez, E., Tattersall, R. S., Manson, J. J., Collaboration, H. A. S., et al. (2020). Covid-19: consider cytokine storm syndromes and immunosuppression. *Lancet (London, England)*, 395(10229):1033.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of*

the Conference on Empirical Methods in Natural Language Processing, pages 262–272. Association for Computational Linguistics.

Molteni, M. (2020). Meet ace2, the enzyme at the center of the covid-19 mystery.

Momtazi, S. and Naumann, F. (2013). Topic modeling for expert finding using latent dirichlet allocation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(5):346–353.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nenkova, A. and McKeown, K. (2011). *Automatic summarization*. Now Publishers Inc.

Nenkova, A. and McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

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

Ochin, Kumar, S., and Joshi, N. (2016). Rule power factor: a new interest measure in associative classification. *Procedia Computer Science*, 93:12–18.

Organization, W. H. et al. (2020). Coronavirus disease 2019 (covid-19): situation report, 72.

O'callaghan, D., Greene, D., Carthy, J., and Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657.

Padmanabhan, B. and Tuzhilin, A. (2000). Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 54–63.

Papadimitriou, C. H., Raghavan, P., Tamaki, H., and Vempala, S. (2000). Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235.

Park, J. S.-Y. and Bucholtz, M. (2009). Introduction. public transcripts: Entextualization and linguistic representation in institutional contexts. *Text & Talk*, 29(5):485–502.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering*, 16(11):1424–1440.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases*, pages 229–238.

Pieniążek, M. (2015). The application of paul ricoeur's theory in interpretation of legal texts and legally relevant human action. *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique*, 28(3):627–646.

Pierce, J. E. (2019). *Languages and linguistics: an introduction*, volume 4. Walter de Gruyter GmbH & Co KG.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577.

Porter, M. F. (2001). Snowball: A language for stemming algorithms.

Ramage, D., Rosen, E., Chuang, J., Manning, C. D., and McFarland, D. A. (2009). Topic modeling for the social sciences. In *NIPS 2009 workshop on applications for topic models: text and beyond*, volume 5, page 27.

Raschka, S., Fernandez, P., Bahnsen, A. C., Abramowitz, M., hsperr, and Kale, A. (2016). mlxtend: v0.4.1.

Řehřek, R. and Sojka, P. (2011). Gensim—statistical semantics in python. *Retrieved from genism. org*.

Rieger, J., Koppers, L., Jentsch, C., and Rahnenführer, J. (2020). Improving reliability of latent dirichlet allocation by assessing its stability using clustering techniques on replicated runs. *arXiv preprint arXiv:2003.04980*.

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

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rosner, F., Hinneburg, A., Röder, M., Nettling, M., and Both, A. (2014). Evaluating topic coherence measures. *arXiv preprint arXiv:1403.6397*.

Roth, S. (2016). Fashionable functions: A google ngram view of trends in functional differentiation (1800-2000). In *Politics and Social Activism: Concepts, Methodologies, Tools, and Applications*, pages 177–203. IGI Global.

Rudy, S. (2010). *Poetry of grammar and grammar of poetry*, volume 3. Walter de Gruyter.

Sahar, S. (1999). Interestingness via what is not interesting. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 332–336.

Sahoo, N., Callan, J., Krishnan, R., Duncan, G., and Padman, R. (2006). Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 357–366.

Schofield, A., Magnusson, M., and Mimno, D. (2017). Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 432–436.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Sievert, C. and Shirley, K. (2014). Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70.

South, A. M., Diz, D. I., and Chappell, M. C. (2020). Covid-19, ace2, and the cardiovascular consequences. *American Journal of Physiology-Heart and Circulatory Physiology*.

Stevens, K., Kegelmeyer, P., Andrzejewski, D., and Buttler, D. (2012). 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*, pages 952–961.

Taddy, M. (2012). On estimation and selection for topic models. In *Artificial Intelligence and Statistics*, pages 1184–1193.

Tan, P.-N., Kumar, V., and Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313.

Tang, G., Müller, M., Rios, A., and Sennrich, R. (2018). Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*.

Tao, F., Murtagh, F., and Farid, M. (2003). Weighted association rule mining using weighted support and significance framework. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–666.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

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

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009). Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981.

Wandmacher, T. and Antoine, J.-Y. (2008). Methods to integrate a language model with semantic information for a word prediction component. *arXiv preprint arXiv:0801.4716*.

Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015). Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*.

Wang, Y., Zhao, X., Sun, Z., Yan, H., Wang, L., Jin, Z., Wang, L., Gao, Y., Zeng, J., Yang, Q., et al. (2014). Towards topic modeling for big data. *arXiv preprint arXiv:1405.4402*.

Wilbur, W. J. and Sirotkin, K. (1992). The automatic identification of stop words. *Journal of information science*, 18(1):45–55.

Xue, G.-R., Dai, W., Yang, Q., and Yu, Y. (2008). Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Yao, H. and Hamilton, H. J. (2006). Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3):603–626.

Zhang, H., Padmanabhan, B., and Tuzhilin, A. (2004). On the discovery of significant statistical quantitative rules. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383.

Appendix

A1 Stop Word Sets

Table A1.1: Stop Word Sets

Stop Word Set	Words
Set 1	virus, viral, cell, protein, patient, treatment, immune, response, infection, infected, group, control, test, clinical
Set 2	level, analysis, sequence, specific, function, associated, based, solution, model, significant, significantly
Set 3	case, hospital, care, risk, day, state, event, admission, rate, effect, type, year, found, reported, number, anti
Set 4	well, process, technique, could, system, finding, technique, performed, sample, gene, genome, animal, human
Set 5	week, different, hospital, public, health, compared, shown, need, i.e., infectious, left
Set 6	figure, child, mouse, antibody, cause, experiment, figure, assay, table, site, high