Performance of Latent Dirichlet Allocation with Different Topic and Document Structures

Haotian Feng
Clemson University, regalia.feng@gmail.com

Performance of Latent Dirichlet Allocation with different topic and document structures

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Statistics

by
Haotian Feng
August 2019

Accepted by:
Dr. William Bridges, Committee Chair
Dr. Alexander Herzog
Dr. Jun Luo
Dr. Christopher McMahan
Dr. Ilya Safro

Abstract

Topic modeling has been used widely to extract the structures (topics) in a collection (corpus) of documents. One popular methods is Latent Dirichlet Allocation (LDA). LDA assumes a Bayesian generative model with multinomial distributions of topics and vocabularies within the topics. The LDA model result (i.e., the number and types of topics in the corpus) depends on tuning parameters. Several methods, ad hoc or heuristic, have been proposed and analyzed for selecting these parameters. But all these methods have been developed using one or more real corpora. Unfortunately, with real corpora, the true number and types of topics are unknown and it is difficult to assess how well the data follow the assumptions of LDA. To address this issue, we developed a factorial simulation design to create corpora with known structure that varied on the following four factors: 1) number of topics, 2) proportions of topics in documents, 3) size of the vocabulary in topics, and 4) proportion of vocabulary that is contained in documents. Results suggest that the quality of LDA fitting depends on the document-topic distribution and the fitting performs the best when the document lengths are at least four times the vocabulary size. We have also proposed a pre-processing method that may be used to increase quality of the LDA result in some of the worst-case scenarios from the factorial simulation study.

Dedication

To my mother and father, Yan Ma and Jianping Feng. To my wife, Zhuhua Tang. To my dear little son, Huaiyuan Feng.

Acknowledgments

First and foremost, I would like to thank my advisor Dr. William Bridges for his constant support, patience and guidance throughout my doctoral studies. Billy, thank you for being an invaluable source of wisdom and ideas and more importantly for being a role model of a researcher and a teacher. This is the only section that you can not edit and please be calm when you find any bad sentences.

I am thankful to my committee members Dr. Alexander Herzog, Dr. Jun Luo, Dr. Christopher McMahan, and Dr. Ilya Safro for their helpful and insightful comments and opinions that made this dissertation better.

I would like to mention those who have made the journey through my doctoral studies an exciting and enjoyable experience. I would like to start by thanking Dr. Scott R Templeton and Dr. Christopher Cox for their support in helping me switching from Economics major to Math major. Many thanks for the instruction and guidance from Dr. Kevin James and Dr. Taufiquar Khan, who has provided a wonderful environment in the mathematics department.

Lastly, I would like to thank my family. Jianping Feng and Yan Ma, it is your endless love and encouragement that makes me be myself now. Zhuhua Tang, my dear wife, flew thousands miles away from home to accompany me in the other side of the planet. Huaiyuan V. Feng, my son little Varian, took all my spare time and are planning to do the same in the future. I can't imagine being able to complete my doctoral studies without the unconditional support from my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In linguistics, the concept of "topic" was originally described as the item that the sentence is about. When extended to the whole document, the concept of "topic" evolved towards "abstract", which usually consists of a few sentences. In an abstract, information is concentrated and communicated while details are omitted. This allow readers to quickly decided if they have interest in the document. Readers save valuable time and can focus on articles of interest. As the number of documents has increased, keywords/labelling systems begin to be important because even reading abstracts became too time consuming.

As far back as the of year 1800, around the time that Joseph Marie Jacquard invented the Jacquard loom, human beings started to convert natural language to something that a machine could record. In 1945, the earliest electronic general-purpose computer called Electronic Numerical Integrator And Computer (ENIAC) was completed. It didn't take too long before people realized that computers were more efficient than human beings for recording and summarizing documents. In 1950, Calvin Moores formally defined the term "Information Retrieval"[23] as the discovery and location of stored-away information so that it can be used. He specifically pointed out the difference between Information Retrieval and Information Warehousing. The later is more like the database systems we are using today for cataloging and storage of information. The simplest method to do information retrieval is to carefully describe the information wanted in terms of a natural language description and then search the whole database to find documents that match the description. There

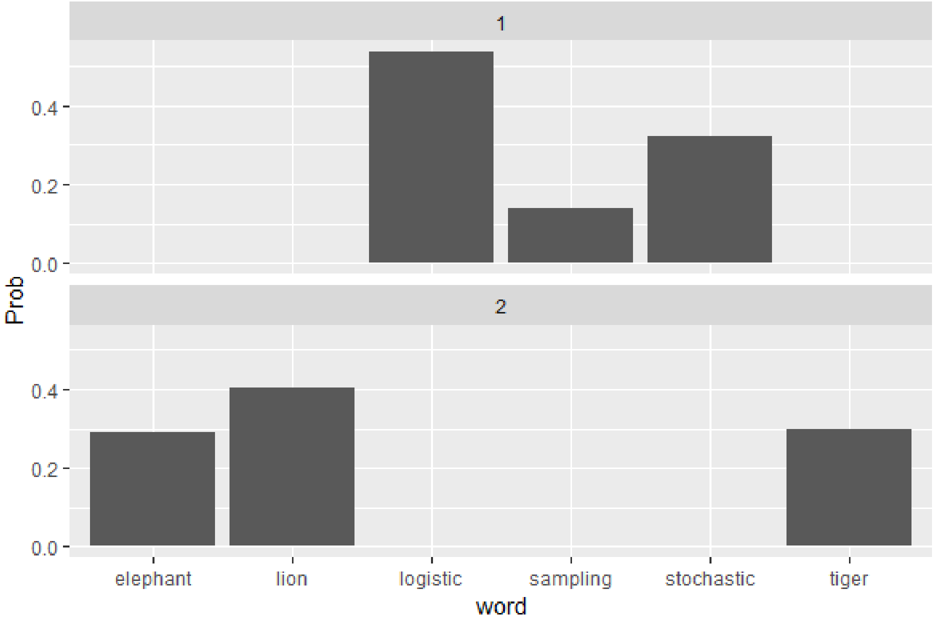

Figure 1.1: Topic illustration

are two main problems with this simple method of information retrieval:

- Human beings often can not accurately describe everything they want in terms of natural language.
- The complexity of linguistics allow several possible descriptions that are essentially equivalent to each other.

To solve these problems, researchers started to use sets of keywords instead of a natural language description to conduct information retrieval. Hence, for each information retrieval inquiry, we can just return the set of articles that contain the same set of keywords or a related set of keywords. In 1955, James W. Perry introduced “precision” and “recall” as concepts to evaluate the quality of an information retrieval task[24]. This was the start of systematic and objective evaluations of information retrieval tasks.

Topic modeling is a specific method of information retrieval that uses statistical tools to discover the ”Topics” that occur in a collection of articles. “Topic” is defined as a multinomial distribution over the words in a vocabulary. Consider a simulated document with a vocabulary consisting of six words: elephant, lion, tiger, logistic, sampling, stochastic. Human beings would most likely recognize two potential sets within the vocabulary and we would probably name them “animals” and “statistics”. Instead of giving a name to each set of words, topic modeling treats these two sets as two probability distributions. Figure 1.1 shows an illustration of the two topic distributions in a simulated document. Two interesting observations from Figure 1.1 are:

1. The probabilities for words like elephant are very small within topic 1, but are not zero. This can be interpreted as it is very rare but possible that someone has document about a statistical experiment related to elephants (or some combination like this)

2. The sets of words that have relatively high probabilities represent the original concept of ”Topic” as we discussed above. We will use topic-distribution or topic-word distribution for clarity in the remainder of this dissertation.

With the help of computers, searching and discovering relevant pieces of information has evolved from supervised methodologies to unsupervised methodologies. By saying unsupervised, it means that the methods will produce results without the need of assistance from humans, but only after the data has been properly processed and the parameter estimates have been selected. Topic models are considered some unsupervised but do require participation of human beings during the processing. Statistical topic modeling is a useful method for finding topics in large unstructured document collections. Typical applications of topic modeling includes document clustering[31], exploratory data analysis and visualization[30], retrieving document translation pairs[21], and automatic labelling[20].

One of the most widely used topic model methodologies is Latent Dirichlet Allocation (LDA) [4]. It allows multiple topics which are represented by multinomial topic-word distributions for one document and documents may possess different topic structures which are represented by the document-topic distribution. A Bayesian estimation approach allows the LDA model the flexibility of extension and makes the estimation steps easier. Documents may possess different topic structures and topics may evolve as new data are observed. More details of LDA are going to be discussed in Chapter 2.

While the LDA methodology is widely used to produce an estimated topic-word distributions, there is no guarantee that the resulting set of topics is correct. In fact, the concept of the correct set of topics is not well understood in topic modeling. Some previous work has been done on evaluating topic modeling results, but they do not directly address the issue of the correct set of topics.

document	word	n
Deathly Hallows_33	snape	113
Goblet Fire_36	dumbledore	95
Deathly Hallows_10	kreacher	87
Goblet Fire_21	dobby	77
Goblet Fire_30	dumbledore	76
Chamber Secrets_2	dobby	65
Deathly Hallows_35	dumbledore	63
Goblet Fire_24	hagrid	63
Deathly Hallows_23	greyback	61
Deathly Hallows_24	wand	61
Goblet Fire_27	sirius	61

Table 1.1: Sample Data from the Collection of the Selected Books

Developing techniques to determine if the topics produced by topic modeling are correct is the primary objective of this research. In the next section the current methods of evaluating topic modeling results will be discussed; but first an example illustrating the concept of correct topics will be discussed. Suppose we are trying to find the topics for a collection of three randomly selected *Harry Potter* books : *Chamber of Secrets*, *Deathly Hallows*, and *Goblet of Fire*. Since there are three books, we set the number of topics to be detected as three and use the standard default R package **topicmodels**. After removing the default stop-words (words that are considered meaningless, i.e. “a”, “the”) in **topicmodels** and applying the proper pre-processing steps, the three books are then converted to a data set that suitable for LDA analysis. Table 1.1 shows part of the data set.

The resulting of the three fitted topic-word distributions are shown in Figure 1.2. This is clearly not the correct set of topics based on simple observation of the plots without the need of any formal statistical analysis. Some of the issues that indicate that this particular LDA solution is not the correct set of topics include:

- ”harry” is the highest frequency word within each topic distribution.
- Names in general appear too often as high frequency words in the distributions.
- Topic 1 and Topic 3 of the LDA solution have the same five words with the highest distributions.

A solution to finding the correct topic distributions in this specific example is straightforward. One can simply eliminated all the formal names from the data set to remove them in the LDA solution. But to figure out what names should be removed, one needs to read the text thoroughly and that is out of the scope of an un-supervised method like LDA and requires even more assistance from human

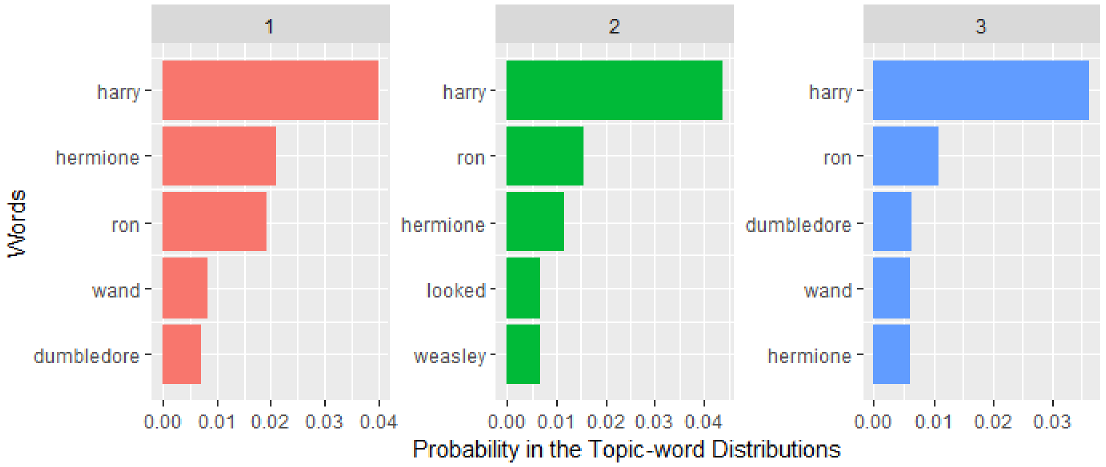

Figure 1.2: Topic-word distribution generated by Latent Dirichlet Allocation

beings. In this dissertation, we will present a method that systematically evaluates the performance of LDA model under different scenarios.

It is worth noting that topic models are applied on data sets from unstructured texts. However, topic model methods usually explicitly or implicitly assume a structure for the texts from which the data sets are developed. In Chapter 3 we will discuss more about the structures implicitly and explicitly proposed by LDA methodology.

1.2 Previous Research on Topic Model Evaluation

The evaluation methods can be divided into four categories. The categories are combinations of two classification factors. The first classification factor is intrinsic vs extrinsic and the second classification factor is efficient vs accurate.

1.2.1 Intrinsic vs. Extrinsic

The intrinsic vs extrinsic classification is based on whether or not extra tasks are required to perform the evaluation. Intrinsic evaluations only consider the fitted result and the original data set. On the other hand, extrinsic evaluations utilize extra tasks which take the topic model outputs (i.e. topic-word distributions) and evaluate the quality of the topic model with based on the extra tasks.

1.2.2 Efficiency vs Accuracy

The efficient vs accurate classification is based on whether or not the metric used in the evaluation depends on the situation. Efficiency measures the time and resources required for the method to produce results and independent to the method itself. Accurate measures how well the model performs based on the selected scenario. Efficiency has a clear definition and usually consists of time complexity and memory usage which are discussed in detail below. Accuracy, on the other hand, is still struggling in finding a good measure. Table 1.4 illustrates the categories for evaluation methods. It is worth noting that the two types of evaluation methods are related to each other and may be used as trade-off pairs.

In general, efficiency is easy to measure and accuracy is much more difficult to measure.

1.2.2.1 Efficiency

There are usually two methods of measuring efficiency. The first method is based on the computational complexity to process the model, and this method is part of the intrinsic evaluation. The second method is based on the time or resource consumed to process the model in real applications and this method is part of extrinsic evaluation.

The computational complexity not only depends on the structure of the model itself, but also depends on the procedures used for estimating the model parameters. For LDA, the original paper [4] derived an estimation algorithm that uses variational inference but in [12], a bayesian approach usingthe Gibbs Sampling procedure is developed for estimation. So there are usually multiple estimation methods that can be selected for a given model. In addition to the multiple estimation techniques, many topic models are designed to enhance the performance under specific scenarios and use extra information besides the dataset. For these reasons (multiple estimation techniques and specific scenarios), computational complexity is rarely used as a measure of efficiency. However, if a standard data set and standard estimation algorithm were chosen for testing topic models, this might change in the future.

Extrinsic measurement of efficiency is based on time and resource used to completely process the model. For this method of efficiency, there are some widely used standard data sets. These data sets are considered as the ”standard” data sets, and the time and resource consumed to perform the same task may be used to compare topic model approaches. The sizes of data sets have increased rapidly due to internet searching and new tools had been developed to reduce the time and resources required for these large data sets. For example, Gropp et al.[13] derived Clustered Latent Dirichlet Allocation and showed that the model does well in terms of the wall-time when the number of processors is increased. The quality of the result in terms of perplexity (to be discussed later) is not degenerated. In the same paper, they also proved that the CLDA model conserves the ability of expansion from simple LDA model.

Since standard data sets do exists, extrinsic efficiency evaluations are straight forward to check, easy to compare, and have little ambiguity. But the measure of efficiency is partly determined by the physical configuration of the platform which researcher is using. The same algorithm may have different running time on different computers. Hence, comparing the efficiency measurements across papers is hard, if not impossible. As a result, there does not exist a preferred system of efficiency evaluation.

1.2.2.2 Accuracy

The accuracy evaluation approach is based on trying to determine how accurately the true underlying topics are discovered by the topic model. The term "Accuracy" is commonly used within the area of statistics and has several definitions. Table 1.2 lists the definitions of Accuracy and several related measures based on a confusion matrix. These metrics are all designed to determine (in slightly different ways) if the model is performing "accurately" for the given task. The problem with using theses "accuracy" measures from a confusion matrix is the requirement of a "True Condition" which is rarely known in topic modeling. Also, topics models require both the topic-word distribution and the document-topic distribution to be estimated, so there are actually two levels of "True Condition". Hence, the definitions of "accuracy" from Table 1.2 can't be directly applied.

1.2.2.3 Perplexity and Other Intrinsic Measurements

Perplexity is one of the most widely used intrinsic methods to evaluate topic models. It was originally introduced in information theory as a measure of how well a probability distribution predicts a sample. It is based on the log-likelihood of the language model, which tries to predict the next word given the current word. For example, suppose there is a document D that is written under topic T. Under topic model assumptions, the document D is equivalent with a list of frequencies of the words which showed up, and the topic T is a probability distribution over the whole vocabulary.

		True Condition	
	Total population	Condition positive	Condition negative
Predicted condition	condition positive	True positive	False positive
	condition negative	False negative	True negative
		$\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$, True positive rate (TPR), Recall, Sensitivity, probability of detection	$\frac{\sum \text{False positiv}}{\sum \text{Condition negative}}$,False positive rate (FPR), Fall-out
		$\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$,False negative rate (FNR), Miss rate	$\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$Specificity (SPC), Selectivity, True negative rate (TNR)
		$\frac{\sum \text{Condition positive}}{\sum \text{Total populatio}}$,Prevalence	$\frac{\sum \text{True positive}+\sum \text{True negative}}{\sum \text{Total population}}$, Accuracy (ACC)
		$\frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}$,Positive predictive value (PPV), Precision	$\frac{\sum \text{False positive}}{\sum \text{Predicted condition positive}}$,False discovery rate (FDR)
		$\frac{\sum \text{False negative}}{\sum \text{Predicted condition negative}}$,False omission rate (FOR)	$\frac{\sum \text{True negative}}{\sum \text{Predicted condition negative}}$, Negative predictive value (NPV)
		$\frac{TPR}{FPR}$,Positive likelihood ratio (LR+)	$\frac{LR+}{LR-}$, Diagnostic odds ratio (DOR)
		$\frac{FNR}{TNR}$,Negative likelihood ratio (LR)	

Table 1.2: Contingency table and Evaluation metrics generated from it

Similarity Metric	Algebraic Expression	Min. Value	Max. Value
Kullback-Liebler (KL)	$\sum_{i=1}^{n} p(x_i) log \frac{p(x_i)}{q(x_i)}$	0	∞
Jensen-Shannon (JS)	$\frac{1}{2}KL(p, \frac{p+q}{2}) + \frac{1}{2}KL(q, \frac{p+q}{2})$	0	1
Hellinger (He)	$\sum_{i=1}^{n}(\sqrt{p(x_i)} - \sqrt{q(x_i)})^2$	0	2

p and q are two discrete probability distributions

Table 1.3: Common similarity measurements used in measuring distance between probability distributions

	Efficient	Accurate
Intrinsic	Run time for one specific model	Perplexity
Extrinsic	Memory consumption for Parallelized computing	Accuracy in Retrieve documents under a specific topic

Table 1.4: Categories of Evaluation

The likelihood is the probability that the document D is presented as it is now under the probability distribution T. The log-likelihood is the log transformation of the likelihood to avoid the extremely small value.

Similarly, the ideas of information entropy (the expectation of the negative log likelihood) is trying to quantify the uncertainty of texts, the Kullback-Leibler divergence (Table 1.3) measures the distance between two probability distributions that is not symmetric, the Jensen-Shannon divergence and Hellinger divergence measure the symmetric distance between two distributions. They are some of the classical metrics inherited from information theory. Manning [5] showed how perplexity is generated from entropy. Hanna Wallach in [28] showed in detail on how to estimate perplexity.

1.2.2.4 Human Interpretability

Topic models have the potential to provide a better understanding of large document collections by discovering interpretable topics (or small sets of words). This process may be viewed as a type of dimension reduction. One way to measure the quality of the model results is to ask human experts to review the topics that been produced and comment the accuracy of how the model topics reflect the true underlying topics in the documents. Unfortunately, it is often the case that the data set based on the collection of documents is too large or too diverse for any human to accomplish this task.

1.2.2.5 Other Ad-hoc Measurements

For a specific task, often an ad-hoc measurement of accuracy is created. A specific example comes from speech recognition tasks. Word-error rate and M-ref [8] has been shown to be better than perplexity in measuring accuracy.

1.3 Problems with the Current Evaluation Methods

1.3.1 Lack of truth or "true condition"

The lack of knowledge of the actual true underlying topics is the major problem for current evaluation methods. As an example, the two metrics precision and recall defined in Table 1.2 are widely used but they both require the knowledge about "True Positive" which is not available in general. In topic modeling, we may never know the true condition. Two experts may agree on the top few words which reflect the topic for a given article, however, when extended to 50 words, they most likely will diverge. It is also an unknown as to how many words from the word distribution should be included to find the best representation of a topic.

1.3.2 Issues with Perplexity

Perplexity is designed to measure the log-likelihood of a held-out test set. This is usually done by splitting the data set into two parts, one for training and the other for testing (the held-out part). A training set is used to estimate the document-topic distribution and the topic-word distribution. The held-out part, the test set, is then used to compute the perplexity value.

Suppose LDA model generates the document-topic distribution Θ and the topic-word distribution Φ. The log-likelihood of the test set is

$$\mathbf{L}(D_{test}) = \log P(D_{test}|\Theta, \Phi) = \sum_i \log P(D_i|\Theta, \Phi)$$

The perplexity is then defined as

$$perplexity(D_{test}) = \exp\{-\frac{\mathbf{L}(D_{test})}{N}\}$$

where N is the total number of words in the test set and D_i is the i-th document in the test set of documents.

Perplexity provides an intrinsic evaluation measure, but it has several drawbacks. The first drawback was shown by Chen[8]. The issue was that, surprisingly, predictive likelihood (perplexity) and human judgement are often not correlated, and even sometimes slightly negatively correlated. They ran a large scale experiment on the Amazon Mechanical Turk platform. For each topic, they took the top five words of that topic and added a random sixth word (de). Then, they presented these lists of six words to participants asking them to identify the intruder word.

If every participant could identify the intruder, then we could conclude that the topic is well defined and easy to recognize. If on the other hand, many people identified one of the top five words from the topic as the intruder, it means that they could not easily identify the topic, and we can conclude the topic was not well defined. The result suggests that, given a topic, the five words that have the largest frequency within their topic are usually not enough to clearly describe a topic.

A second issue with perplexity occurs when using modern Bayesian topic models with a more complicated structures of topics. These models often lead to an intractable posterior likelihood. For a held-out collection of test set D_{test} documents and a corpus wide vocabulary of V words, when evaluating LDA model, perplexity is computed using the following formula:

$$perplexity(D_{test}) = 2^{\{-\frac{\sum_{d=1}^{D_{test}} \log(\sum_{n=1}^{N_d} \sum_{t=1}^{T} (\theta_{dt}\phi_{tn}))}{\sum_{d=1}^{D_{test}} N_d}\}}$$

where ϕ_{tn} is the inferred probability of word n in topic t and θ_{dt} is the probability of assigned topic t to the held-out document d while N_d is the total number of words in that document. The multiplication of $\theta_{dt}\phi_{tn}$ is the part that is intractable. Since word n might be presented in multiple topics, one needs to figure out which topic it belongs to at every time the word n is presented, which is impossible.

Hence the perplexity has to be estimated through some sampling methods. In this situation, the estimated perplexity is not a deterministic measurement, but rather a stochastic measurement. This means that each time perplexity is estimated for the same corpus, the perplexity scores will be different.

The final draw-back for perplexity is shown in another word intrusion experiment conduct by Chang et al. [7]. In this paper, Chang et al. showed that the words in topics generated through the lowest perplexity criteria often do not have natural semantic relationship with each other.

1.3.3 Model Assumptions

One big problem for topic models is that the model assumptions are not able to be validated. Some of the assumptions, like "Topics evolve in time so they are correlated to each other", might be validated through a thorough investigation of the corpus. For a given corpus, one may find human experts to summarise the topics for each time period and compare the result. Methods to assess other assumptions, like "Topics are multinomial distribution over the vocabulary", have not been developed.

1.3.4 Parameter Tuning

Simple models like the n-grams topic model which will be introduced in Chapter 2 do not have a parameter to be tuned. More complicated models such as Bayesian methods require selection of proper tuning parameters. Even the commonly used LDA model contains three parameters that need to be tuned. The standard tuning parameter selection method is to run the topic modeling approach several times over different setups of the parameter, and select the set of parameter values that produces the lowest perplexity score. When no parameter values produce a topic modeling result with a reasonable perplexity value, the topic modeling result does not make semantic sense, or the perplexity score varies widely based on the parameter values, there is no recommended alternative plan for parameter value selection.

The importance of the parameter tuning can not be over stated. In the LDA modeling approach, there is a parameter α that defines the convergence rate and it plays an important role in the LDA results as illustrated in Figure 1.3. This figure shows how the associated document-topic multinomial distribution is going to change when the different value of the parameter α is used. A small α value leads to a multinomial distribution that contains few high-probability topics and a large α value leads to a more "flat" multinomial distribution. The number of topics to be fitted is another important tuning parameter to be considered. Unlike α that can be interpreted as a prior belief, and this prior might be overridden in Gibbs Sampling process, the number of topics is the primary factor in evaluating the quality of modeling result.

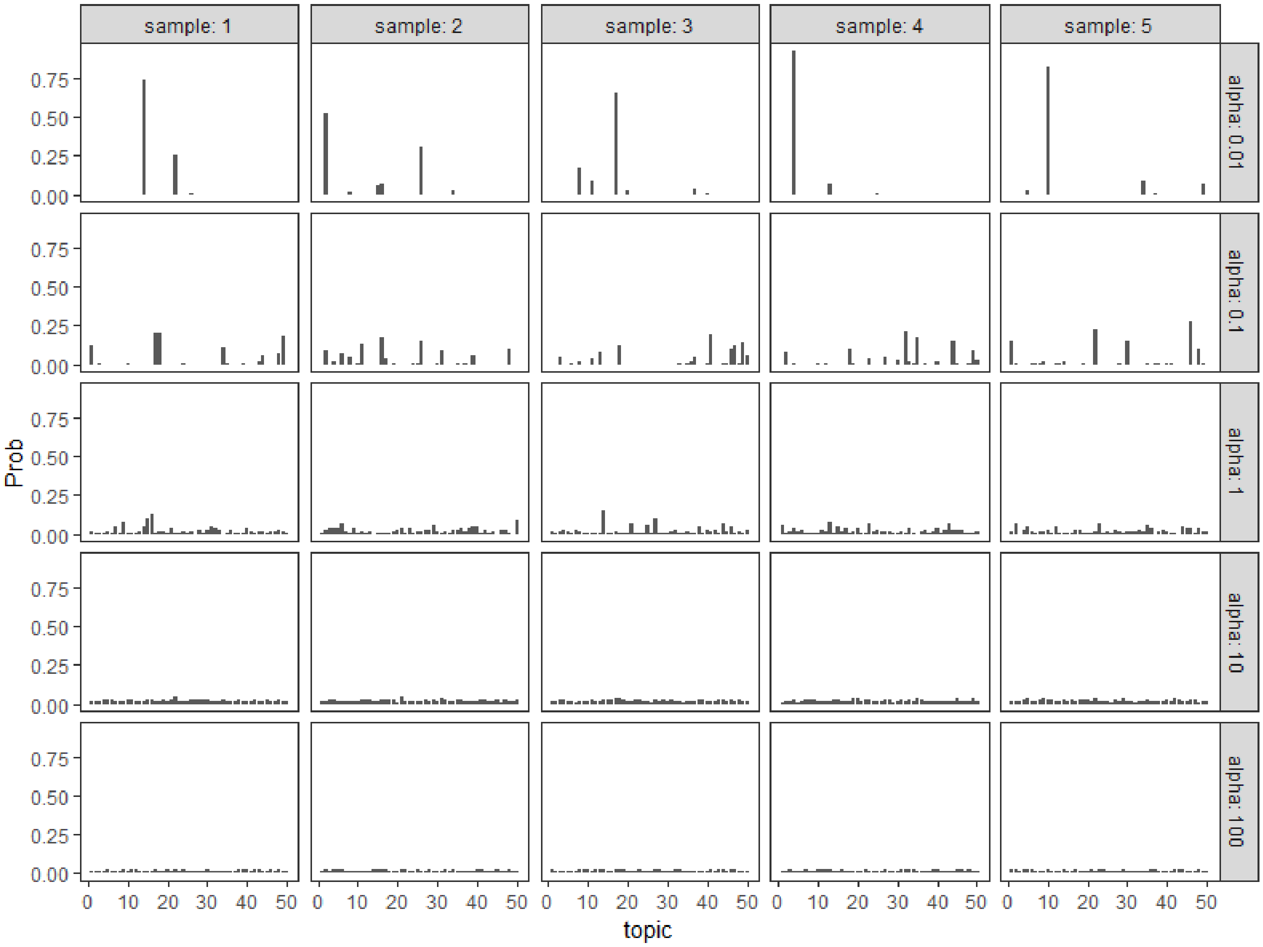

Figure 1.3: Example of multinomial distribution samples drawn from Dirichlet distributions with different parameter α

1.3.5 Dependency of evaluation on a Particular Corpus

Although there are many articles produced each year, only a few are used for evaluation purpose. TREC (Text REtriveal Conference) publishes about 10 datasets a year, and approximately 1 human-annotated dataset for every 3 years. While these can be used as standards, the quality of the annotation is actually unknown and the annotated data may not be similar to other data sets used by researchers. The other reason that the evaluation method is corpus dependent is that some models require more information about the dataset than others. Rosen-Zvi et al. in [26] published a well-designed author-topic model for authors and documents. Without the certain type of information, the model should be at most has a similar quality as LDA.

1.3.6 Pre-processing Steps Impact on Evaluations

This is best illustrated in Figure 1.2. It is hard for anyone to get enough information about the topics from the graphs. There are all the names from the novel that take the higher probability. Should we remove the names if we know it is not helping? It is still not clear. The books are definitely talking about Harry Potter and his mates. Remove those names result in other “not-as-important” names showing up. For this specific example, maybe more topics or a longer list of words will help. But the pre-processing steps like stop-words recognition and tokenization do have influence on the topic model fitting result, but this has rarely been evaluated.

1.4 Build the Underline Truth

The generative process used in Bayesian Topic modeling provides a useful tool for correctly evaluating the model results. The main problem of not knowing the true condition is solved. Based on a carefully designed simulation, one can construct a corpus with known topic-word distribution as well as the document-topic distribution. In this case, We may also modify the assumptions from the model to check the performance of the model under different conditions. Knowing the true distributions also allow us to use the statistic measurements of agreement like correlation. It is also an attempt to avoid corpus dependency and issues associated with pre-processing steps. Parameter tuning might be studied in detail as well. We will talk about these in detail in Chapter 4.

In Chapter 1, we have discussed that proper evaluation of topic modeling results is unknown.

We propose in this study to develop method to perform a proper evaluation.

In Chapter 2, we will develop the technical details in some classic topic models and evaluation methods. In Chapter 3, we will decompose the structure of LDA model and detect important characteristics. In Chapter 4, we will go through the detail of simulation study that can be used to evaluate topic model results. In Chapter 5, we will discuss the findings and future work.

The objective of this study is to learn the impacts of different topic and document structures to the performance of LDA model. The topic and document structures can be summarised with certain characteristics that will be discussed in Chapter 3. We would like to identify the impacts of different characteristics on topic modeling results.

Chapter 2

Technical Background

2.1 Introduction

In chapter 1 we briefly discussed Latent Dirichlet Allocation. In general, topic models are trying to extract article features, and use the features to represent articles for applications like answering a query. For these applications, the human interpretability is not always the largest concern. The techniques for extracting features are usually conducted within single articles first, and then extended to multiple documents. This extension to multiple articles is not an easy or simple extension and will be discussed later.

The simplest topic modeling approach is the unigram model. Word counts are used to summarize the information in the articles. The idea is simple: if a word w occurs more times than other words, then w should be able to capture more information compared with other words. A *Topic*, also known as the *Topic-word distribution*, is a multinomial distribution that assigns probability to each word in the vocabulary. Usually, a topic is illustrated by a few high-probability words. It is a natural extension of the idea of word counts. A collection of articles may contain multiple topics, denoted by K. Since a multinomial distribution can be represented by its parameter $\vec{p}$, it is sufficient to use the associated parameters to represent topics. Under this idea, multiple topics can be represented in a matrix: let $\varphi_k = (p_1, ..., p_V), k = 1, ..., K$ be one set of parameters among K topics, then $\mathit{\Phi} = \{\varphi_k$ as row k, $k = 1, ..., K\}$ is the desired topic matrix.

The ultimate target of any Topic model is to find the proper $\mathit{\Phi}$, topic-word distribution, for any given collection of documents. We will discuss some basic definitions and terms first, then some important probability background, and finish the chapter with formal details about topic models and evaluation methods.

2.1.1 Definitions and Terminology

In this section we will state clear definitions and terms for the discussion that follow. Some of the definitions and terms are quite self-explanatory, but we will try to provide a strict concepts instead of general ideas.

- **Corpus**: Corpus is a collection of documents or articles. It usually contains articles from the same language. The documents or articles contained in the corpus may contain extra information other than the text itself. i.e. title, author names, tags, keywords, date of publishing, where it is published, table of contents, etc. This extra information may help in increasing the quality of the model estimates. Given that there are M documents altogether within the corpus, We have:

$$\text{Corpus} = \begin{pmatrix} D_1 & D_2 & \dots & D_M \end{pmatrix}$$

  Where each D_i indicates a single document.

- **Tokens**: The basic building blocks of the texts. A token is considered as the smallest element to express a single meaning to the reader. Word-token is the most commonly used token in topic modeling. More complicated token concepts might consist of semantic phrases or even sentences. For a single document D_i that has N_i tokens, document D_i is denoted by:

$$D_i = \begin{pmatrix} w_{i1} & w_{i2} & \dots & w_{iN_i} \end{pmatrix}$$

- **Vocabulary**: An ordered set of tokens. It should contain all possible tokens based on the method of tokenization. In real-world applications, this is usually impossible to construct. There are new words and phrases that are invented everyday. Typically, when there is a relatively large corpus, people will collect the tokens that have shown up in the corpus to build the vocabulary. The size of the ordered set of vocabulary is denoted by V. Each token can be expressed by a V dimensional vector with 1 at the associated entry and 0 elsewhere.

$$\text{Vocabulary} = \begin{pmatrix} w_1 & w_2 & \dots & w_V \end{pmatrix}$$

Note that the subscript only has one index, instead of two indexes, in the notation of tokens in documents.

- **Topic**: A topic is a distribution of words over the whole vocabulary. It is also referred as "topic-word distribution". Usually a multinomial distribution is used. The dimensionality of the distribution is the same with the size of vocabulary, V. When there are multiple topics associated with a corpus, we use a matrix called topic-word distribution matrix such that each row represent one topic and each column is for a token. Hence, if we have K topics altogether, the topic-word distribution matrix will be a $K \times V$ dimensional matrix. This is usually denoted by φ:

$$\varphi = \begin{array}{c} \\ topic_1 \\ topic_2 \\ \vdots \\ topic_K \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & w_2 & w_3 & \dots & w_V \end{array} \\ \left[\begin{array}{ccccc} \varphi_{11} & \varphi_{12} & \varphi_{13} & \cdots & \varphi_{1V} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \cdots & \varphi_{2V} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_{K1} & \varphi_{K2} & \varphi_{K3} & \cdots & \varphi_{KV} \end{array} \right] \end{array}$$

- **Document-topic distribution**: The uni-gram model only allows one topic for each document. If we relax the assumption and allow more than one topic within the same document (which is a natural relaxation), then each topic might contribute to some proportion of the documents. Of course, a single sentence, or maybe a single word, is possible to be contained within more than one topic. A latent assumption for topic models is that *each word is coming from only one topic*. Hence, suppose we know exactly where each word is coming from, then the proportion of each topic's contribution may be computed. Naturally, by definition, these proportions build a multinomial distribution. Figure 2.1 illustrates the distribution of two documents and five topics. Note that it is possible that the probability of some topics is zero for some specific documents. Hence, the total number of topics associated with the corpus might not be the same as the number of topics associated with each document. Similar to the topic-word distribution matrix, we use the document-topic distribution matrix to record these information. Suppose we have M documents within our corpus and K topics associated with the corpus, the document-topic distribution matrix will be a $K \times M$ dimensional matrix such that each column indicates a document and each row indicates a topic. We usually use Θ to denote this matrix.

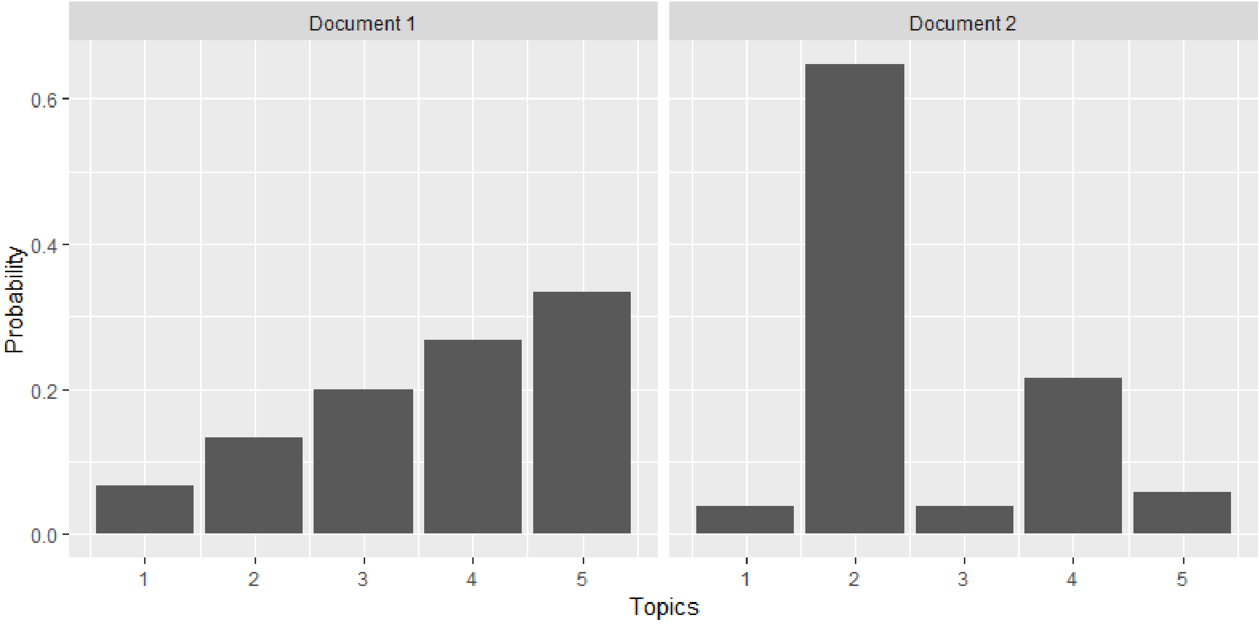

Figure 2.1: Example of Document-Topic distribution for 2 documents and 5 Topics

$$\Theta = \begin{array}{c} \\ topic_1 \\ topic_2 \\ \vdots \\ topic_K \end{array} \begin{array}{c} \begin{array}{ccccc} Document_1 & Document_2 & Document_3 & \dots & Documet_M \end{array} \\ \left[\begin{array}{ccccc} \theta_{11} & \theta_{12} & \theta_{13} & \dots & \theta_{1M} \\ \theta_{21} & \theta_{22} & \theta_{23} & \dots & \theta_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{K1} & \theta_{K2} & \theta_{K3} & \dots & \theta_{KM} \end{array} \right] \end{array}$$

2.1.2 Bayes' Theorem and Beyes' Estimation

Bayes' Theorem is one of the most important probability rules used in language models and topic models. Suppose we have two events A and B, then the event that both A and B happens at the same time is denoted by $A \cap B$. The probability associated with those events are $P(A)$, $P(B)$, and $P(A \cap B)$, respectively. We may use a Venn Diagram, Figure 2.2, to illustrate the situation:

The conditional probability is defined based on the the probability that both A and B happened. The probability that event A occurs known event B already occurred is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.1}$$

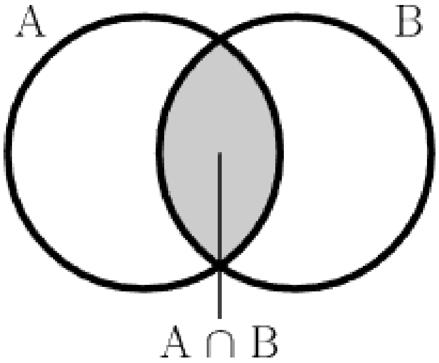

Figure 2.2: A Venn Diagram illustrating the probability that both event happen. Areas represent probabilities

Similarly, the probability that even B occurs known that even A already occurred is:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{2.2}$$

From equation 2.1 and 2.2, We have:

$$\begin{aligned} P(A \cap B) &= P(A|B)P(B) \\ P(A \cap B) &= P(B|A)P(A) \\ \Rightarrow P(A|B)P(B) &= P(B|A)P(A) \\ \Rightarrow P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \end{aligned}$$

The Bayes' Theorem relates the probability before getting the evidence $P(A)$ to the probability after getting the evidence $P(A|B)$. For this reason, $P(A)$ is called the prior probability and $P(A|B)$ is called the posterior probability. The fraction $\frac{P(B|A)}{P(B)}$ is called the likelihood ratio. Using these terms, Bayes' theorem can be rephrased as "the posterior probability equals the prior probability times the likelihood ratio".

For topic modeling, Bayes' Theorem is used as follows. Given two discrete random variables $X == x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_m$, Bayes' Theorem is:

$$P(X = x_i|Y = y_j) = \frac{P(Y = y_j|X = x_i)P(X = x_i)}{P(Y = y_j)} \tag{2.3}$$

Based on the rule of total probability, $P(Y = y_j) = \sum_{i=1}^{n} P(Y = y_j, X = x_i) = \sum_{i=1}^{n} P(Y = y_j|X = x_i)P(X = x_i)$, equation 2.3 may be expressed as:

$$P(X = x_i|Y = y_j) = \frac{P(Y = y_j|X = x_i)P(X = x_i)}{\sum_{i=1}^{n} P(Y = y_j|X = x_i)P(X = x_i)} \tag{2.4}$$

The idea of Bayesian generative process is constructed using Bayes' Theorem as expressed in Equation 2.4. In general, one assumes no prior information about an article in the beginning. Hence, a distribution is selected as the standard distribution, and it is utilized to describe the topic model. The multinomial distribution is the natural selection to describe the topics and we will discuss it in the next section. After observing the article, one will update the standard distribution with the information gathered from the article.

The process of Bayesian estimation relies upon the structure of the article. Or more precisely, the way that we believe the article is constructed. Those beliefs are summarized into assumptions and different models are very likely to have different assumptions. Are the words independent to each other? Are there multiple topics within each article? How should one choose the standard distribution (or prior)? How should one choose the likelihood function? Is there any other information, i.e. author name and affiliation, that might help in generating the posterior estimate?

One obvious conclusion is that there is no universal set of assumptions that fits for all documents. Some of those assumptions are able to be tested, but most of them can not. Moreover, similar with regression analysis, we need to understand that the model is almost always wrong (i.e. not completely consistent with the actual data generating process). But some of the models might produce a better estimate of the underlying truth than others. The only way to find it out is to evaluate the result, especially when the assumptions are unable to be checked.

2.1.2.1 Important Assumptions for Bayesian Estimation in Topic Model

There are still some baseline assumptions that everyone uses. One is that the conceptual population of all texts that might be constructed from the current language systems. One question we are interested in is that what is the basic building block of these texts? Linguists have dived deep into this topic. Chomsky [9] stated that the grammatical differences between human languages can be explained on the basis of a small number of hierarchically organized discrete principles and parameters. Mark C. Baker tried to construct principles and parameters theory in his book [1]. We

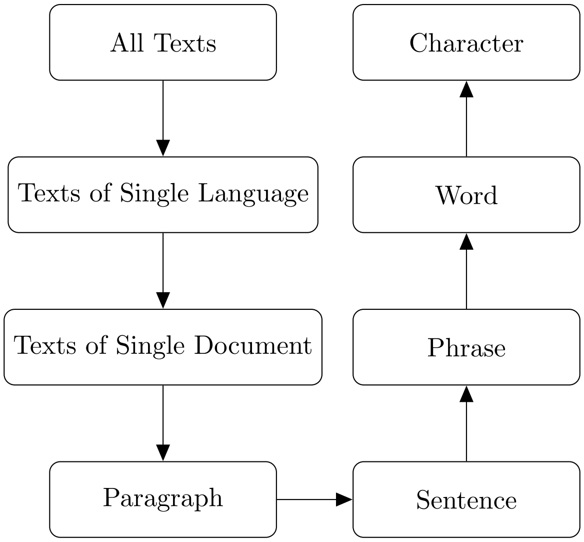

Figure 2.3: Layer of Texts

would like to point out that languages are discrete and texts are countable. This property allows us to select a proper discrete distribution while building our model. Figure 2.3 shows that all layers of texts are discrete. This is a prior belief in all topic models that we should know.

2.1.3 Multinomial Distribution and Dirichlet Distribution

The multinomial distribution is a natural choice in topic modeling. Suppose there is an random experiment that might generate a finite number of results, and the probability of those results that have been generated is a known fixed number. We may use a vector of zeros and ones to indicate which result is generated by the experiment. This vector is a random variable and we define the distribution of this random variable to be multinomial. If multiple independent experiments have been conducted, the multinomial random variable is defined with a vector that summarizes the number of each result that is generated.

Formally, suppose there are n trials of an experiment. For each trial, there is a set of k possible out comes $\{x_1, x_2, \ldots, x_k\}$ from the random experiment with a set of probability values $\{p_1, p_2, \ldots, p_k\}$ such that the $\sum_{i=1}^{k} p_i = 1$ and $p_i \geq 0$, the probability of x_i happens is defined as $P(X = x_i) = p_i$. Define $\vec{c} = (c_1, c_2, \ldots, c_k)$ such that

$$c_i := \{\text{The number of instances of the out come } x_i \text{ being observed in the } n \text{ trials}\}$$

then $\vec{c}$ is said to have a multinomial distribution with parameters $\vec{p} = (p_1, p_2, \dots, p_k)$ and n.

The probability mass function of a multinomial distribution is:

$$P(C = (c_1, c_2, \dots, c_k)) = \frac{n!}{c_1! \dots c_k!} p_1^{c_1} \dots p_k^{c_k} = \frac{(\sum_{i=1}^k c_i)!}{\prod_{i=1}^k c_i!} \prod_{i=1}^k p_i^{c_i}$$

In the above equation, the fraction $\frac{(\sum_{i=1}^k c_i)!}{\prod_{i=1}^k c_i!}$ is called the multinomial coefficient which quantifies the number of ways that we could divide the set of observations n into subsets of size from c_1 to c_k. We may also use the Gamma function to represent this coefficient:

$$\frac{(\sum_{i=1}^k c_i)!}{\prod_{i=1}^k c_i!} = \frac{\Gamma(\sum_{i=1}^k c_i + 1)}{\prod_{i=1}^k \Gamma(c_i + 1)}$$

The topic-word distribution matrix φ and the document-topic distribution Θ contains the parameter of probabilities $\vec{p}$ for each multinomial distribution. Each element in vector $\vec{p}$ takes value from zero to one and is restricted by $\sum_{i=1}^k p_i = 1$. In a uni-gram model, maximum likelihood estimates of these probabilities are found using a frequentist approach. The Bayesian estimation method would assume that these probabilities follow some continuous distribution and try to update this prior belief using the observed data. The Dirichlet distribution and the multinomial distribution are two of the most commonly used prior distributions in the Bayesian approach.

The Dirichlet distribution, often denoted by $Dir(\alpha)$, is a family of continuous multivariate probability distributions that take on positive real number values based on parameter α. The support of the Dirichlet distribution is the vector $\vec{x} = (x_1, x_2, \dots, x_k)$ where $x_i \in (0, 1)$ and $\sum_{i=1}^k x_i = 1$. α is often referred as the concentration parameter because it determines the spread of the realization from the distribution. Figure 1.3 in chapter 1 illustrated the effect of α. Note that the support of the Dirichlet distribution is exactly the restriction of the parameters of the multinomial distribution. Hence, the α is also called the hyperparameter since it could be the parameter of the probability distribution of the probability parameter of multinomial distribution.

The probability density function of the k dimensional Dirichlet distribution is:

$$f(\vec{p} = (p_1, \dots, p_k) | \vec{\alpha} = (\alpha_1, \dots, \alpha_k)) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k p_i^{\alpha_i - 1}$$

If α_i are the same for all i, we call the Dirichlet distribution **symmetric** and often write the vector $\vec{\alpha}$ as a product of a scalar and a k-dimensional vector so that all entries are 1s: $\vec{\alpha} = \alpha\vec{1}$. Sometimes the $\vec{1}$ is ignored and the parameter of a Dirichlet distribution is simply stated as α, which indicates a symmetric distribution. α is also known as the concentration parameter of the Dirichlet distribution. In general, one may use a base vector $\vec{u}$ such that $\vec{\alpha} = \alpha\vec{u}$. If $\vec{u}$ is not a vector of ones, then the Dirichlet distribution is called asymmetric. Figure 1.3 shows five multinomial distributions drawn from five different Dirichlet distributions with different symmetric hyperparameters $\alpha \in [0.01, 100]$. As the parameter value increases, the random variable is more evenly spread over the possible outcomes and eventually almost uniformly distributed when the parameter value is very high.

As we mentioned in Chapter 1, the values of the hyperparameters are typically set using certain heuristics which are based on the document collection. Griffiths et al in [12] stated that for Latent Dirichlet allocation, $\alpha = \frac{50}{T}$ where T is the number of topics, and $\beta = 0.01$ for the document-topic and the topic-word distributions often generate reasonable results. Hence, these values are been considered the default for fitting Latent Dirichlet Allocation. The other way to estimate the concentration parameter is discussed in Minka [22], which generated maximum likelihood estimates of the parameters.

Asymmetric Dirichlet distributions also discussed by Wallach et al. [28]. She states that the asymmetric Dirichlet priors for document-topic distributions offer modeling advantages over symmetric priors in terms of evaluation based on perplexity. To find the best asymmetric base measures vector estimate, she applied a hierarchical structure of Dirichlet priors in which another symmetric Dirichlet distribution is considered as the prior of the base measures vector.

Based on equation 2.3, if we take X as the vector of probabilities from the underlying multinomial distribution, that follows a Dirichlet distribution before we observed the data set, and Y is the observation of the multinomial distribution, then the posterior distribution $P(X|Y)$ after we observe the data Y is also a Dirichlet distribution. In general, if the posterior distribution is from the same family of distributions as the prior these distributions are called a conjugate pair. Across different generative models it is more convenient to use conjugate pairs as they simplify the description of the generative process and provide mathematical convenience when deriving the posterior estimates.

There are many well known conjugate pairs such as Gamma-Poisson, Beta-Binomial, Gamma-Exponential, Normal-Normal, etc. The initial distribution is the prior and the latter one is the likelihood distribution. We have discussed the conjugate relationship above using the Dirichlet-Multinomial conjugate pair which is also used in the Latent Dirichlet Allocation. Next, we use a simple example to show how the concept of conjugate paires is utilized for topic modeling.

Suppose we have a topic $t = (p_1, p_2, \ldots, p_5)$ that is built upon a vocabulary which has five different tokens $Vocabulary = \{w_1, w_2, w_3, w_4, w_5\}$ and a document D_1 that contains 100 words which is built solely upon this topic. Hence, $D_1 = (w_{1,1}, w_{1,2}, \ldots, w_{1,100})$ and each $w_{1,j}$ is selected from the vocabulary based on probabilities within t. By definition, the number of times each token is observed $\vec{c} = (c_1, c_2, \ldots, c_5)$ should follows a multinomial distribution with parameter $\vec{p} = t$ and $n = 100$. Hence, we have:

$$\vec{c} \sim Multinomial(t, 100) \tag{2.5}$$

Furthermore, we assume that t is a random variable that follows a symmetric Dirichlet distribution with parameter α.

$$t \sim Dir(\alpha) \tag{2.6}$$

This is called the prior distribution and α is the hyperparameter that can be interpreted as our prior belief before observing data. We would like to estimate the true distribution of $\vec{c}$ which is equivalent with estimating the parameter t. Since t is a random variable, we would like to find its posterior distribution $f(t|\vec{c})$. Based on Bayes' rule:

$$f(t|\vec{c}) = \frac{f(\vec{c}|t)f(t)}{f(\vec{c})} \tag{2.7}$$

Where

$$f(\vec{c}|t) = (p_1, p_2, \ldots, p_5)) = \frac{\Gamma(\sum_{i=1}^{5} c_i + 1)}{\prod_{i=1}^{5} \Gamma(c_i + 1)} \prod_{i=1}^{5} p_i^{c_i}$$

$$f(t|\alpha) = \frac{\Gamma(\sum_{i=1}^{5} \alpha)}{\prod_{i=1}^{5} \Gamma(\alpha)} \prod_{i=1}^{5} p_i^{\alpha - 1}$$

and $f(\vec{c})$ is the true probability that $\vec{c}$ is observed, which is a constant fixed number. α is also a constant. There fore we can express $f(t|\vec{c})$ proportional to the following functions. and may be ignored while computing the kernel of the distribution. Hence, we find the posterior distribution:

$$
\begin{aligned}
f(t|\vec{c}) \propto & \frac{\Gamma(\sum_{i=1}^{5} c_i + 1)}{\prod_{i=1}^{5} \Gamma(c_i + 1)} \frac{\Gamma(\sum_{i=1}^{5} \alpha)}{\prod_{i=1}^{5} \Gamma(\alpha)} \prod_{i=1}^{5} p_i^{c_i} \prod_{i=1}^{5} p_i^{\alpha - 1} \\
\propto & \prod_{i=1}^{5} p_i^{c_i} \prod_{i=1}^{5} p_i^{\alpha - 1} \\
\propto & \prod_{i=1}^{5} p_i^{c_i + \alpha - 1}
\end{aligned}
\tag{2.8}
$$

Since they do not contribute to the kernel of the distribution, they should be able to re-generate through normalization.

The kernel of the Dirichlet distribution may be found by dropping all constant terms. Suppose $\theta \sim Dir(\alpha_k)$, then

$$
\begin{aligned}
f(\theta) =& (\alpha_1, \ldots, \alpha_k)) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} p_i^{\alpha_i - 1} \\
\propto & \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}
\end{aligned}
\tag{2.9}
$$

Compare equations 2.8 and 2.9, we note the similarity of the kernel. More specifically, θ and t are random variables, and α_i and $c_i + \alpha$ are the parameters. This demonstrates that the kernel generated from equation 2.8 is from an asymmetric Dirichlet distribution with parameter $\vec{c'} = (c_1 + \alpha, c_2 + \alpha, \ldots, c_5 + alpha)$. Both the prior distribution and the posterior distribution are Dirichlet and we shown a proof of conjugacy for our example.

The conjugacy of Dirichlet-Multinomial also presents another interpretation of the hyperparameter α. Note that if c_i is the count of token w_i contained in document D_1, then α will be the prior belief of the number of times token w_i shows up in document D_1. When α is small, then the posterior distribution is more dependent on the observed document. When α is large relative to the document length, then the posterior is more dependent on α.

2.2 Topic Models

2.2.1 Unigram Model

The Unigram model is one of the earliest model developed for topic modeling. There are three assumptions for the unigram model:

- Each word/token is independent of other words.
- Each document only contains one topic
- Topics are not related between different documents.

Those assumptions are not been able to meet with real data. Consider a document that only contains a single sentence:

```
I have a ______ and I always love to play with him.
```

Semantically, a word like “boy”, “kid”, “dog”, “cat”, or “pet” make a lot more sense than “house”, “car” in the blank. And there exist words like “girl” or “mom” that don’t make sense at all because the word “him” points to male. This simple sentence shows that words are not independent of each other within a semantically consistent document. Random generation can be used to create research documents that satisfy this assumption, but that will be discussed later.

The second assumption could be true under a certain scenario. Suppose a writing test is given which asks participants to write about a specific topic. This should generate articles with a single topic. The problem here is that one can never verify if one article truly contains only a single topic. For example, suppose that we have two documents which are written by the participants of the writing test. Conceptually, these two documents should be both talking about the same topic. But these two documents should not be, and never will be, exactly the same. Even for a same large idea, people will write about it differently.

Mathematically, let $V = (w_1, w_2, \ldots, w_n)$ be the vocabulary. $D_1 = (w_{1,1}, w_{1,2}, \ldots, w_{1,n_1})$ be the document that contains n_1 tokens. Then for the Unigram model we have:

$$P(D_1) = P(w_{1,1})P(w_{1,2}) \ldots P(w_{1,n_1}) \tag{2.10}$$

where $P(w_{1,i}) = p_i$ is the true probability of the underline topic generate this token $w_{1,i}$. Hence, the likelihood function of the document D_1 is:

$$L = \prod_{i=1}^{n} p_i^{c_i} \tag{2.11}$$

where c_i is the number of times token w_i is presented in D_i.

The log-likelihood function is:

$$logL = \sum_{i=1}^{n} logp_i^{c_i} = \sum_{i=1}^{n} c_i logp_i$$

with the restriction $\sum_{i=1}^{n} p_i = 1$. Hence, we may compute the maximum likelihood estimation of $\vec{p}$:

$$\begin{aligned} \hat{\vec{p}} = &\quad argmax_{\vec{p}}(logL + \lambda(1 - \textstyle\sum_{i=1}^{n} p_i)) && (2.12)\\ \frac{\partial}{\partial p_i}(logL + \lambda(1 - \sum_{i=1}^{n} p_i)) = &\quad \tfrac{\partial}{\partial p_i} logL + \tfrac{\partial}{\partial p_i}(\lambda(1 - \textstyle\sum_{i=1}^{n} p_i)) \quad = 0 && (2.13)\\ &\quad \tfrac{\partial}{\partial p_i}\textstyle\sum_{i=1}^{n} c_i logp_i - \lambda \tfrac{\partial}{\partial p_i}\sum_{i=1}^{n} p_i \quad = 0 && (2.14)\\ &\quad p_i \quad = \frac{x_i}{\lambda} && (2.15) \end{aligned}$$

Using the property $\sum_{i=1}^{n} p_i = 1$, we find that $\lambda = n$. Hence,

$$\hat{\vec{p}}_{MLE} = (\frac{c_1}{n}, \frac{c_2}{n}, \ldots, \frac{c_n}{n}) \tag{2.16}$$

In other words, the maximum likelihood estimation of the parameter for the topic is the vector of frequency of each word/token in the document.

Other than the unrealistic assumptions, there are still many problems with the Unigram model. For example, the maximum likelihood estimation only takes tokens that showed up in the document. This means that the token that didn't show in the document will be assigned a probability zero. A smoothing algorithm might be used to solve this problem [6]. Bigram, Trigram, and N-gram models are built based on the Unigram model which relax the independent token assumption. These models do not use the Bayesian generative process and are considered as the traditional methods.

2.2.2 Latent Dirichlet Allocation

We briefly introduced the LDA model in Chapter 1. Latent Dirichlet Allocation (LDA), was originally introduced by Blei et al.[4] and it is one of the most important probabilistic topic model. It allows each single document to have multiple topics, thus introduces a document-topic distribution Θ, similarly with the topic-word distribution $\mathit{\Phi}$. Imagine that each topic is a unique-colored urn of water, then to write an article we would like to select some of the urns and mix the water from different urns. In this way, if we have a total of K topics, Θ will be a K dimensional multinomial distribution. Under this set up, while writing articles, it is natural to firstly pick up a topic using Θ, then pick up a word from the selected topic using $\mathit{\Phi}$.

The key idea of LDA is that $\mathit{\Phi}$ and Θ are conditionally independent to each other if we know which word and topic the token has. Figure 2.4 shows the unrolled graph model of LDA. The top part of the graph (above the $\Updownarrow$ sign) is equivalent with the bottom part. Each circled node represents a random variable. The shaded nodes are observed variables and others are latent variables. Arrows represent the dependency between the random variables. The letter N, M, and T located in the bottom part of the graph (below the $\Updownarrow$ sign) represent the number of repeated arrows in the top part of the graph.

2.2.2.1 Assumptions of LDA

- Documents contain multiple topics.
- The total number of topics of the corpus is a fixed number.
- Each document is assumed to be generated by a known process (to be describe next).
- Words are generated independently of other words (often called the bag-of-words assumption).

2.2.2.2 Generative process for LDA:

1. Choose K, the number of topics in the collection; Choose α and β, the hyperparameter.
2. Draw $\varphi_k \sim Dir(\beta), k = 1, ..., K$, V dimensional multinomial topic-word distribution for K topics.
3. Then for each document d in the corpus:

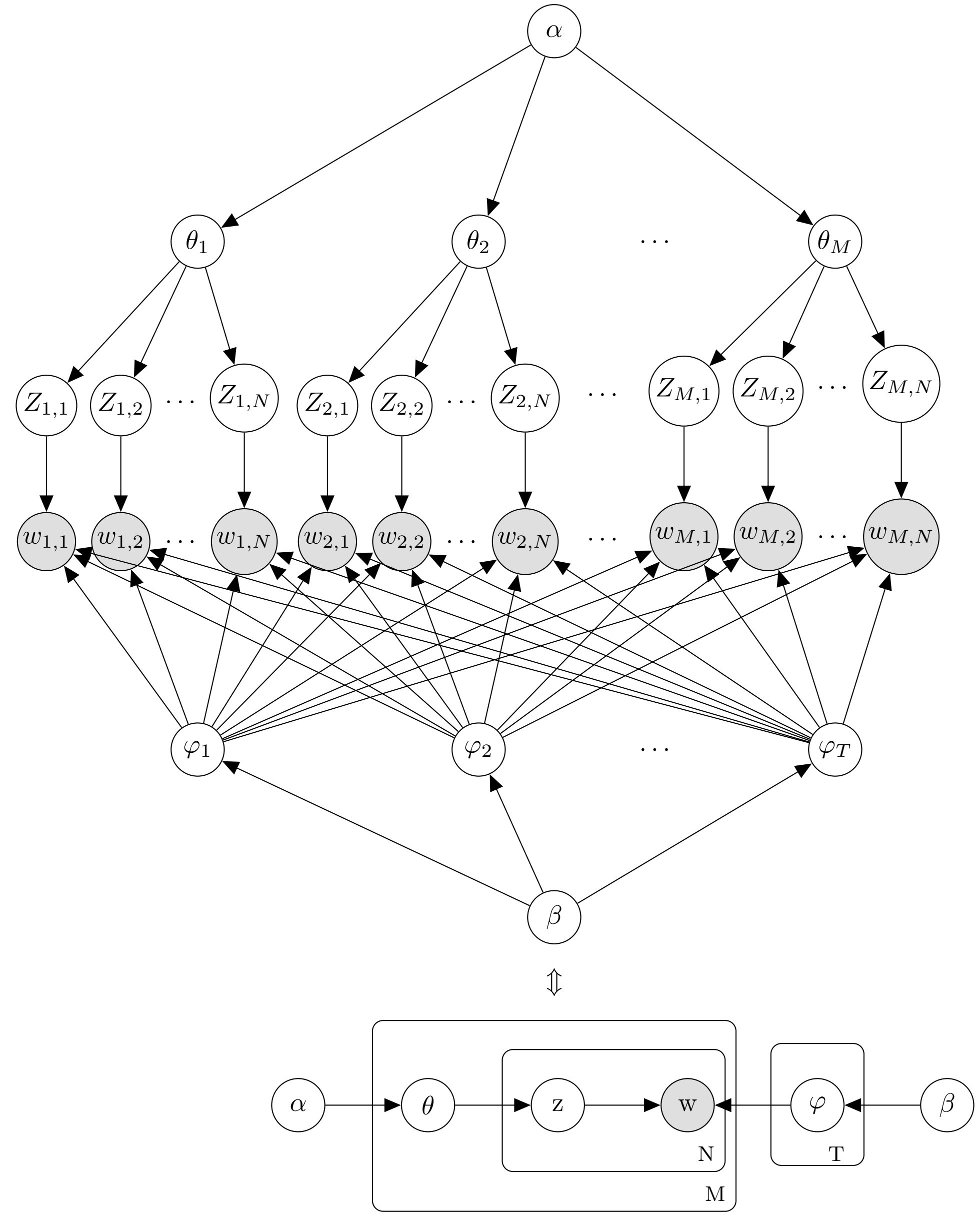

Figure 2.4: Unrolled graphical model representation of LDA

(a) Draw $\theta_d \sim Dir(\alpha)$ which is the parameter for the multinomial document-topic distribution. Thus θ_d determines how topics are mixed within any specific document.

(b) Then for each word w_i in document d:

i. Draw a topic index $z_i \sim Multinomial(\theta_d)$.

ii. Draw a word $w_i \sim Multinomial(\varphi_{z_i})$, which is the topic generated in the beginning of the process.

2.2.2.3 Estimation in LDA:

There are three common method to do parameter estimation in LDA: Variational EM, expectation propagation, and Gibbs Sampling. The most widely used approach among those three is the Gibbs Sampling approach, followed by the Variational EM algorithm. In this section we discuss both approaches and go through the Gibbs Sampling method in detail since we choose to use Gibbs Sampling for the simulation discussed later. Heinrich[15] provides more details.

2.2.2.3.1 Gibbs Sampling

Gibbs Sampling is a variant of the Metropolis-Hasting method which constructs a Markov chain whose states are parameter settings and whose stationary distribution is the true posterior over those parameters. There are the original Gibbs Sampling method and the collapsed Gibbs method. We will go through details of the original Gibbs method.

Using the notation from the generative process, suppose we know $\vec{z}, \vec{w}$, let z_d be the vector of topic assignment for words in document d [1], then from equation 2.9, we know that the posterior of the d-th documents' document-topic distribution is also a multinomial distribution that has parameter θ_d:

[1] $z_{d,i}$ might be considered as a multinomial distributed random variable that has parameter $n = 1$, or a categorical random variable that is equivalent with a discrete random variable that takes values from 1 to K

$$
\begin{aligned}
p(\theta_d|z_d,\alpha) &= \frac{p(z_d|\alpha,\theta_d)}{p(z_d|\alpha)}p(\theta_d|\alpha) && (2.17)\\
&\propto \prod_{n=1}^{N_d} Multi(z_{d,n}|\theta_d)Dir(\theta_d|\alpha) && (2.18)\\
&\propto \prod_{n=1}^{N_d} (\theta_{d,k})^{I_{(z_{d,n}=k)}} Dir(\theta_d|\alpha) && (2.19)\\
&\propto \prod_{k=1}^{K} \theta_{d,k}^{n_d^{(k)}} Dir(\theta_d|\alpha) && (2.20)\\
&\propto Dir(\theta_d|\vec{n_d}+\alpha), \vec{n_d} = \{n_d^{(k)}\}_{k=1}^{K} && (2.21)
\end{aligned}
$$

Where $n_d^{(k)}$ refers to the number of times that topic k has been observed with a word from document d.

Similarly,

$$
\begin{aligned}
p(\varphi_k|\vec{z},\vec{w},\beta) &= \frac{p(\vec{w}|\vec{z},\varphi_k,\beta)}{p(\vec{w}|\vec{z},\beta)}p(\varphi_k|\vec{z},\beta) && (2.22)\\
&\propto \prod_{i:z_i=k} Multi(w_i|\varphi_k)Dir(\varphi_k|\beta) && (2.23)\\
&\propto \prod_{i=1}^{V} \varphi_{k,i}^{n_k^{(i)}} Dir(\varphi_k|\beta) && (2.24)\\
&\propto Dir(\varphi_k|\vec{n_k}+\alpha), \vec{n_k} = \{n_k^{(i)}\}_{i=1}^{V} && (2.25)
\end{aligned}
$$

Where $n_k^{(i)}$ refers to the number of times that the word indexed by i in the vocabulary is initiated under topic k.

Since for the Dirichlet distribution $Dir(p)$ we have $E[X_i] = \frac{p_i}{\sum p_i}$, we will get the estimator:

$$
\begin{aligned}
\widehat{\varphi_{k,t}} &= \frac{n_k^{(t)}+\beta_t}{\sum_{t=1}^{V}(n_k^{(t)}+\beta_t)} && (2.26)\\
\widehat{\theta_{d,k}} &= \frac{n_d^{(k)}+\alpha_k}{\sum_{k=1}^{K}(n_d^{(k)}+\alpha_k)} && (2.27)
\end{aligned}
$$

Thus, the only problem left to solve is how to find an estimate of $\vec{z}$? This is where Gibbs

Sampling is useful. We need to find the full conditional distribution for each parameter z_i:

$$p(z_i = k|\vec{z_{\neg i}}, \vec{w}, \alpha, \beta) = \frac{p(\vec{z}, \vec{w}|\alpha, \beta)}{p(\vec{z_{\neg i}}, \vec{w}|\alpha, \beta)} = \frac{p(\vec{z}, \vec{w}|\alpha, \beta)}{p(\vec{w_{\neg i}}|\vec{z_{\neg i}}, \alpha, \beta)p(w_i|\vec{z_{\neg i}}, \alpha, \beta)p(\vec{z_{\neg i}}|\alpha, \beta)}$$

Using the bag-of-word assumption, w_i is independent with $\vec{w_{\neg i}}$ and also with $\vec{z_{\neg i}}$. Then the denominator turns out to be $p(\vec{z_{\neg i}}, \vec{w_{\neg i}}|\alpha, \beta)$. Thus, if we can find the joint distribution of $\vec{z}$ and $\vec{w}$ given α and β, the Gibbs sampler will be completed.

Note that the joint distribution can be factored:

$$p(\vec{w}, \vec{z}|\alpha, \beta) = p(\vec{w}|\vec{z}, \alpha, \beta)p(\vec{z}|\alpha, \beta)$$

based on LDA's assumption, $w \perp \alpha|\vec{z}$ and also $\vec{z} \perp \beta$. Thus,

$$p(\vec{w}, \vec{z}|\alpha, \beta) = p(\vec{w}|\vec{z}, \beta)p(\vec{z}|\alpha)$$

The first term can be find through an integral[2]:

$$\begin{aligned}
p(\vec{w}|\vec{z}, \beta) &= \int p(\vec{w}|\vec{z}, \mathit{\Phi})p(\mathit{\Phi}|\beta)d\mathit{\Phi} \\
&= \int \prod_{i=1}^{N} p(w_i|z_i)p(\mathit{\Phi}|\beta)d\mathit{\Phi} \\
&= \int \prod_{k=1}^{K}\prod_{t=1}^{V} p(w_i = t|z_i = k)p(\mathit{\Phi}|\beta)d\mathit{\Phi} \\
&= \int \prod_{k=1}^{K}\prod_{t=1}^{V} \varphi_{k,t}^{n_k^{(t)}} p(\mathit{\Phi}|\beta)d\mathit{\Phi} \\
&= \int (\prod_{k=1}^{K}\prod_{t=1}^{V} \varphi_{k,t}^{n_k^{(t)}})(\prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \varphi_{k,t}^{\beta_t - 1})d\mathit{\Phi} \\
&= \int \prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \varphi_{k,t}^{n_k^{(t)}+\beta_t - 1} d\mathit{\Phi} \\
&= \prod_{k=1}^{K} \frac{\Delta(\vec{n_k} + \beta)}{\Delta(\beta)}, \vec{n_k} = \{n_k^{(t)}\}_{t=1}^{V}
\end{aligned}$$

[2] Here we use the notation that $\Delta(p) = \frac{\prod_{k=1}^{dim(p)} \Gamma(p_k)}{\Gamma(\sum_{k=1}^{dim(p)} p_k)}$, as in [15]

Similarly, for the second term we have[3]:

$$
\begin{aligned}
p(\vec{z}|\alpha) &= \int p(\vec{z}|\Theta)p(\Theta|\alpha)d\Theta \\
&= \int \prod_{d=1}^{D} \frac{1}{\Delta(\alpha)} \prod_{k=1}^{K} \theta_{d,k}^{n_d^{(k)}+\alpha_k-1} d\Theta \\
&= \prod_{d=1}^{D} \frac{\Delta(\vec{n_d}+\alpha)}{\Delta(\alpha)}, \vec{n_d} = \{n_d^{(k)}\}_{k=1}^{K}
\end{aligned}
$$

Thus, we have the **Joint Distribution**:

$$
p(\vec{z}, \vec{w}|\alpha, \beta) = \prod_{k=1}^{K} \frac{\Delta(\vec{n_k}+\beta)}{\Delta(\beta)} \prod_{d=1}^{D} \frac{\Delta(\vec{n_d}+\alpha)}{\Delta(\alpha)}
$$

Now we have the **Full Conditional Distribution**[4]:

$$
\begin{aligned}
p(z_i = k|\vec{z_{\neg i}}, \vec{w}, \alpha, \beta) &= \prod_{z=1}^{K} \frac{\Delta(\vec{n_z}+\beta)}{\Delta(\beta)} \prod_{d=1}^{D} \frac{\Delta(\vec{n_d}+\alpha)}{\Delta(\alpha)} \cdot \prod_{z=1}^{K} \frac{\Delta(\beta)}{\Delta(\vec{n_{z,\neg i}}+\beta)} \prod_{d=1}^{D} \frac{\Delta(\alpha)}{\Delta(\vec{n_{d,\neg i}}+\alpha)} \\
&= \frac{\Delta(\vec{n_k}+\beta)}{\Delta(\vec{n_{k,\neg i}}+\beta)} \cdot \frac{\Delta(\vec{n_m}+\alpha)}{\Delta(\vec{n_{m,\neg i}}+\alpha)} \\
&= \frac{\Gamma(n_k^{(t)}+\beta_t)}{\Gamma(\sum_{t=1}^{V} n_k^{(t)}+\beta_t)} \cdot \frac{\Gamma(\sum_{t=1}^{V} n_{k,\neg i}^{(t)}+\beta_t)}{\Gamma(n_{k,\neg i}^{(t)}+\beta_t)} \cdot \frac{\Gamma(n_m^{(k)}+\alpha_k)}{\Gamma(\sum_{z=1}^{K} n_m^{(z)}+\alpha_z)} \cdot \frac{\Gamma(\sum_{z=1}^{K} n_{m,\neg i}^{(z)}+\alpha_z)}{\Gamma(n_{m,\neg i}^{(k)}+\alpha_k)} \\
&= \frac{n_{k,\neg i}^{(t)}+\beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)}+\beta_t} \cdot \frac{n_{m,\neg i}^{(k)}+\alpha_k}{\sum_{z=1}^{K} n_{m,\neg i}^{(z)}+\alpha_z} \\
&\propto \frac{n_{k,\neg i}^{(t)}+\beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)}+\beta_t}(n_{m,\neg i}^{(k)}+\alpha_k)
\end{aligned}
$$

Thus we have the Gibbs Sampler completed.

2.2.2.3.2 Variational EM

The variational EM algorithm converted the approximation problem to an optimization task. It is the method used by Blei[4]. Variational EM selects a family of probability distributions with parameters (called variational parameters) that simplifies the complex dependence of the latent variables θ, z, and φ in the latent Dirichlet allocation model.

Figure 2.5 illustrate the idea of variational EM algorithm. From [4], given a family of

[3] D is the number of documents within our corpus

[4] Suppose the word i is in document m and $w_i = t$ is known.

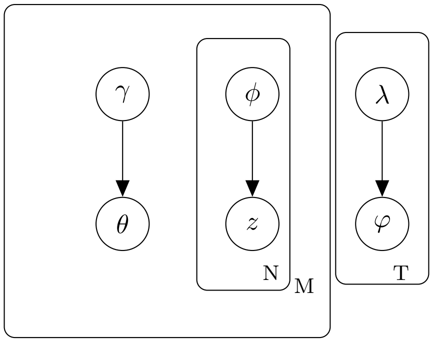

Figure 2.5: The Graphical model representation of the variation EM algorithm

distributions $q(\theta, z, \varphi|\gamma, \phi, \lambda)$, the log likelihood function of a document is:

$$\begin{aligned} logp(w|\alpha, \beta) = & \quad log \int \textstyle\sum_z p(\theta, z, w, \varphi|\alpha, \beta)d\theta & (2.28) \\ = & \quad log \int \textstyle\sum_z \frac{p(\theta,z,w,\varphi)q(\theta,z,\varphi)}{q(\theta,z,\varphi)} d\theta & (2.29) \\ \geq & \quad E_q[logp(\theta, z, w, \varphi|\alpha, \beta)] - E_q[logq(\theta, z, \varphi)] & (2.30) \end{aligned}$$

The difference between the left hand side and right hand side of 2.30 is due to the Kullback-Leibler divergence between the true posterior and variational posterior probability. So the estimation of the LDA is converted into a minimization of the KL divergence:

$$KL(q(\theta, z, \varphi|\gamma, \phi, \lambda)||p(\theta, z, \varphi|w, \alpha, \beta)$$

The EM algorithm may be applied over the three variational parameters. Based on Blei[4], we have:

- (E-step) For each document, find the optimizing valus of the variational parameters $\{\gamma^*_{dt}, \phi^*_{dt}, \lambda^*_{tw}\}$
- (M-step) Maximize the resulting lower bound on the log likelihood with respect to the model parameters α and β. This corresponds to find maximum likelihood estimates with expected sufficient statistics for each document under the approximate posterior which is computed in the E-step

Krstovski[17] showed that the following formula may be used in the EM algorithm.

$$\phi_{wt}^{d} \sim expE_q[log\theta_{dt}] + E_q[log\varphi_{tw}] \tag{2.31}$$

$$\gamma_{dt} = \alpha + \sum_{w=1}^{W} \phi_{wt}^{d} n_{w}^{d} \tag{2.32}$$

$$\lambda_{tw} = \beta + \sum_{d=1}^{M} n_{w}^{d} \phi_{wt}^{d} \tag{2.33}$$

2.3 Evaluation Methods

2.3.1 Perplexity

Perplexity is one of the most widely using intrinsic evaluation metrics for topic models. Ideally, a good model should be able to assign higher probability to the observed real data set (Documents). For example, if our corpus is only one sentence:

```
We have sufficient evidence to _____ that _____
```

Then a good model might assign higher probability to words like “conclude” or “say” for the blanks. A bad model might assign higher probability to words like “cat” or “table” for the blanks.

Mathematically, perplexity is a measurement of how well a probability distribution or probability model predicts a sample. For a discrete distribution p, the perplexity is defined as:

$$Perp = 2^{H(p)} = 2^{-\sum_x p(x) log_2 p(x)} \tag{2.34}$$

Notice that in equation 2.34, the base number of exponentiation and logarithm are both 2. From the property of the logarithm we know that the base need not be 2 as long as the exponentiation and the logarithm use the same base. The choice of 2 is simply for ease of interpretation because $H(p)$ is the entropy of the distribution.

Usually one does not know the true probability distribution for a topic model. Hence, authors like Blei et al [4] selected to use a “uniformed weight” instead of the “true distribution” when computing the perplexity. Their formula is:

$$Perp = 2^{-\frac{1}{\sum_{d=1}^{M} n_d} \sum_{d=1}^{M} log_2 p(w_d)} \tag{2.35}$$

Where M is the number of documents in the corpus and w_d is the word set of the d-th document. The denominator of the exponent is the total number of words in the data set.

The reason that we do not use the entire data set to compute perplexity is that the perplexity is already been optimized after fitting the model. The standard practice is to compute the perplexity on part of the dataset (denoted the training data set). Note that if we have multiple topic models over the same set of document, it might be good to calculate the perplexity of the whole data set over each model, and then use perplexity for model selection. Another reason to split corpus into the training set (and the remainder which is the test set) is to test for prediction, especially with real texts. But obviously, the specific way of splitting the corpus will change the computed perplexity value. Specifically two important issues are:

1. For a different split, the models are fitted over different data sets and hence will generate different results.

2. Different test sets also produce different perplexity.

These two issues make split-set perplexity very difficult to compare across different models. If a model is chosen based on the lowest perplexity for a particular corpus, compare to other models, it is difficult to tell whether the difference is coming from a better model or better luck in splitting the corpus.

2.3.2 Perplexity in LDA

In regards to the Latent Dirichlet allocation, there is another issue in using perplexity: the entropy, or more precisely, $p(w_d)$ is intractable. This is because LDA assumes multiple topics that might overlap with each other. Hence, a word w_i, will posses none-zero probability within more than one topic. In the generative process, one would first pick a topic from the document-topic distribution, and then pick a word token from the associated topic-world distribution. But we can not find out which topic the word is coming from in the test set. Mathematically:

$$Perp(D_{test}) = 2^{-\frac{1}{\sum_{d=1}^{M_{test}} n_d} \sum_{d=1}^{M_{test}} log_2(\sum_{i=1}^{n_d} \sum_{j=1}^{K} (\theta_{jd} \varphi_{ji}))} \tag{2.36}$$

Where θ_{jd} is the probability of j-th topic for d-th document in the test set and φ_{ji} is the probability of the i-th word from the j-th topic.

Another technical issue for computing the held-out perplexity is the unknown document-topic distribution for the test set. Since we assumed that the testing set and the training set are coming from the same population, then the hyperparameters of the document-topic distribution should be the same. Hence, we may use the estimated hyperparameters α and β to generate the document-topic distribution for each document in the test set. This is another source of randomness.

Overall, the perplexity depends on:

- Vocabulary
- Sample size: document-wise and corpus-wise
- Testing set and training set split
- Document-topic distribution generation
- Estimation of the intractable part

Wallach et al [29] summarized five different methods to estimate the intractable part $p(w_d)$.

2.3.3 Human Evaluation

Usually a human evaluation of topic modeling is based on extrinsic methods: people perform a task like clustering based on the result of topic modeling, then evaluate the result of the clustering. Chang et al [7] conducted a experiment to analyse the topics generated by LDA. Although there appears to be a longstanding assumption that the information discovered by topic models is meaningful and useful, the experiment tells a different story.

There are two specific parts of the experiment that show that topic model results are not always meaningful and useful: word intrusion and topic intrusion.

In the word intrusion component, participants were presented with six randomly ordered words and asked to find the word which was out of place or does not belong with the others (intruder). To construct the test cases, they randomly select a topic from the model and pull out the five most probable words for that topic. The word that was considered to be intruder was then selected from the set of low-probability words in that topic. To make sure that the intruder word was meaningful somehow and not rejected outright due solely to rarity, they also restricted the intruder as a word that possessed a high probability in some other topic.

In the topic intrusion component, participants were shown the title and a snippet from a document along with four topics that were represented by the eight highest-probability words within those topics. Three out of four of those topics were high probability topics assigned to that document and the remaining one was chosen randomly from the other low-probability topics in the model. And participants were instructed to choose the topics which did not belong with the document.

For the above two components, if the topics were meaningful and related to the document, then the word intruder and topic intruder should be easy to find.

Chang performed the experiment over three different models: LDA, CTM(Correlated Topic Model [3]), and pLSI(probabilistic latent semantic indexing [16]). He found that the model that has better perplexity does not have the better human interpretability.

In this chapter, we discussed the probability background of the LDA model, derived the estimation procedure of the LDA model, and went through current evaluation systems of topic modeling.

Chapter 3

Decomposing the Structure of Topic Models

3.1 Non-Structured Data and Structured Models

Text data usually doesn't have an explicit structure. Some of the articles or books might contain a certain form of structure, i.e. chapters or sections. But the text itself, a paragraph, a sentence, or a word, only contains an implicit structure. Moreover, the messages contained within the text can not be specified by the structure.

Although it is commonly mixed together, it is worth noting that topic models and language models are different from each other. The difference between language models and topic models is that language models assume the order of the current words preserve information to predict the next word, and the topic models assume words are independent from each other and the order is not important (the bag-of-words assumption). The language models are better in prediction tasks and topic models are better in information retrieval tasks. Logically speaking, topic models are a special type of language model from which the dependence assumption between words is minimized. Figure 3.1 illustrates the relationship between language models and topic models.

The choice of using language models or topic models depends on both the task to preform and one's belief. The discussion about which one is better is out of the scope of this dissertation. Wallach[27] discussed about this topic in detail.

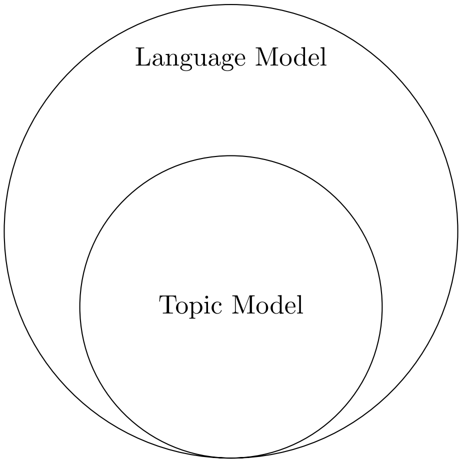

Figure 3.1: Venn's Diagram of the relationship between Language Models and Topic Models

With the bag-of-words assumption, documents can be completely represented by a word frequency distribution. As mentioned in Chapter 2, a pre-processing step will usually be applied to the documents and the following three tasks are performed:

1. Stemming: the process of reducing inflected (or sometimes derived) words to their word stem, base or root formgenerally a written word form.[32]

2. Stop-word removing: stop words are words which are filtered out before or after processing of natural language data (text).[25]

3. Frequency counting: Counting the number of times each word appears in the document.

This process is also called **tokenization**. Words and phrases are translated to tokens so that the tense of words and plural words will not affect the result. After the tokenization, the original meaning of the word is no longer important because only the frequencies of the tokens are used. Tokens are often be treated as integers without losing generality.

As pointed out in the above, there is no structure in plain text data. But there are logics behind each document. Let's consider a simple sentence:

The monitor gets a heart attack.

Technically, this sentence is correct. But one can immediately realize that it is meaningless and a false statement because a monitor doesn't even have a heart. This is the **logic** component of a document. One may argue that in some contexts (such as a fairy tale story) this sentence could be meaningful because a monitor can actually have a heart. While this assumption is made, one is indeed using his/her logic to classify the sentence, as what the topic model is trying to do.

The topic can be considered a logical idea, and one method of defining the logic is the co-occurrence of tokens in the document. In statistics, a multinomial distribution over the whole vocabulary represents the structure of this co-occurrence. Topic models are trying to capture the logical structure of the documents based on this definition. This is why the "True Condition" as discussed before does not really exist in the topic modeling methodology.

In fact, since the set of all possible documents written in English is countable, the number of topics is also countable infinite. If we apply a restriction on the document length, i.e. 5000 characters, then the number of possible documents and the number of topics is actually finite and can be enumerated. One problem is that the number of multinomial distributions is uncountable. This is because the parameters of multinomial distribution are real numbers. If we restricted the number of characters in documents, then the associated topic distribution is restricted by the accuracy supported by the number of characters. This can be easily illustrated in extreme cases, i.e. a document only contains one word, then the parameters of the associated topics must be either 1 or 0. In realistic scenarios this allows us to treat the number of possible topics as countable.

Recall from earlier that corpus is a collection of documents. One critical assumption LDA and other model makes is that there is a fixed number of topics in one corpus. These is usually guaranteed by the way that the corpus is collected. It should be noted that the selection of documents to form a corpus affects the performance of topic models because of the above assumption.

3.2 Characteristics of Data sets and Topic Models

One important objective of this research is to identify the impacts of different characteristics on topic modeling results. There are actually two sets of characteristics necessary to consider to when approaching this objective. The sets are: characteristics of the data set and characteristics of the model.

Important characteristic of a data set to consider are:

1. Number of documents in corpus

2. Size of the Vocabulary

3. Document length

4. Document length relative to the vocabulary size

5. Number of True Topics within the corpus

6. Number of Topics contained in each document

7. Type of distribution of Topics

Important characteristics of the Models to consider are:

1. Number of Fitted Topics

2. Number of True Topics related to number of Fitted Topics

3. Gibbs Sampler Parameters

4. Pre-set hyperparameters

We are going to focus on the characteristics of data set for two reasons:

1. Modification of characteristics of models is easier in real applications and has been considered in many other research studies.

2. In a simulation study, the true values of the characteristics of models are known and hence the performance of the model could be maximized using these values.

As we mentioned in Chapter 1, one of the biggest problem of evaluating topic models is the lack of the "True Condition". Without the "True Condition", the following characteristics are unknown variables:

- Number of True Topics.

- Numer of True Topics related to number of Fitted Topics

- Number of Topics contained in each Document.

- Type of Distribution of Topics.

Hence, if we could create a corpus such that the true values of the characteristics are known, the above characteristics are no longer required to be estimated. This permits us to focus on the topic and document structures instead of the model structure.

3.3 Pilot Study to Examine the Structure

In real world applications, researchers never know the true underlying topic distribution for a given corpus. LDA modeling attempts to provide a best guess on both the topic-word distribution and the document-topic distribution, conditioned on some acceptable assumptions. Our pilot simulation study starts from these assumptions and generates corpora that follow these assumptions exactly, and examines how well the modeling works under different scenarios based on varying these assumptions.

This pilot study was designed to determine if the simulation strategy actually works and also examine which evaluation method is the most suitable for the simulation.

3.3.1 Design

For this pilot study, there were only 2 topics and 10 documents containing 50 words. The total size of the vocabulary (number of terms) was 5. The Φ matrix (topic-word distribution) was:

$$\begin{array}{c} \\ TrueTopic1 \\ TrueTopic2 \end{array} \begin{array}{c} \begin{array}{ccccc} Term1 & Term2 & Term3 & Term4 & Term5 \end{array} \\ \left[\begin{array}{ccccc} 0.4 & 0.25 & 0.05 & 0.2 & 0.1 \\ 0.4 & 0.05 & 0.25 & 0.1 & 0.2 \end{array}\right] \end{array}$$

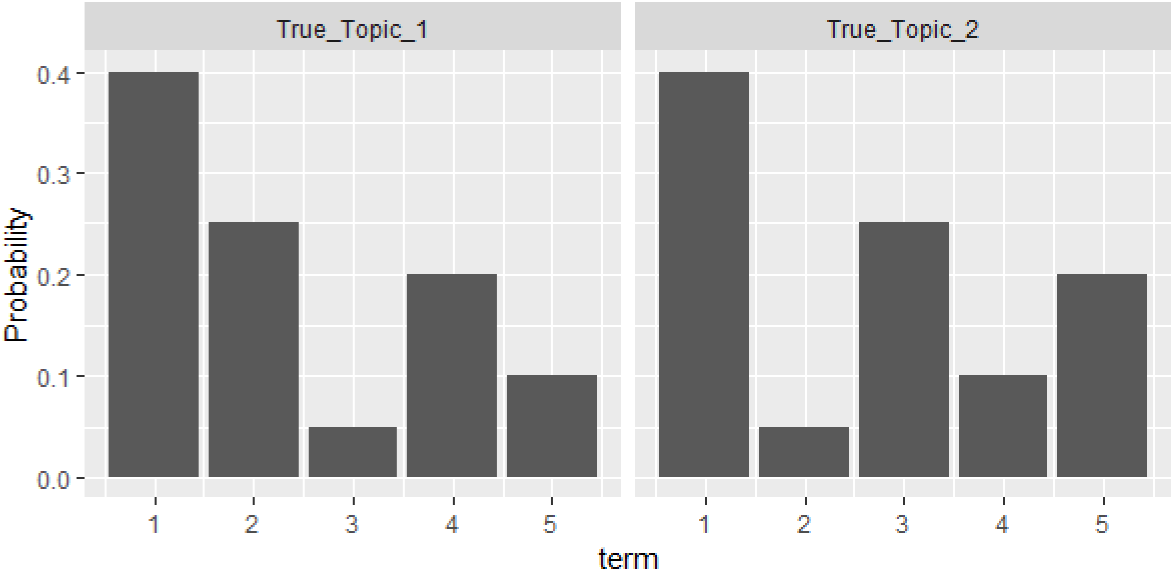

Figure 3.2: Topic-word distribution of Pilot Study

and the Θ matrix (document-topic distribution) was:

$$\begin{array}{c|cc|} & TrueTopic1 & TrueTopic2 \\ Document1 & 1 & 0 \\ Document2 & 0 & 1 \\ Document3 & 1 & 0 \\ Document4 & 0 & 1 \\ Document5 & 1 & 0 \\ Document6 & 0 & 1 \\ Document7 & 1 & 0 \\ Document8 & 0 & 1 \\ Document9 & 1 & 0 \\ Document10 & 0 & 1 \end{array}$$

Figure 3.2 illustrates Φ, the topic-word distribution we actually used in the pilot study. Notice that we did not use the actual word or token. Instead, an integer index is used to represent the tokens. Hence, in our pilot study, we have a total vocabulary of size 5 that contains 5 tokens: {1,2,3,4,5}.

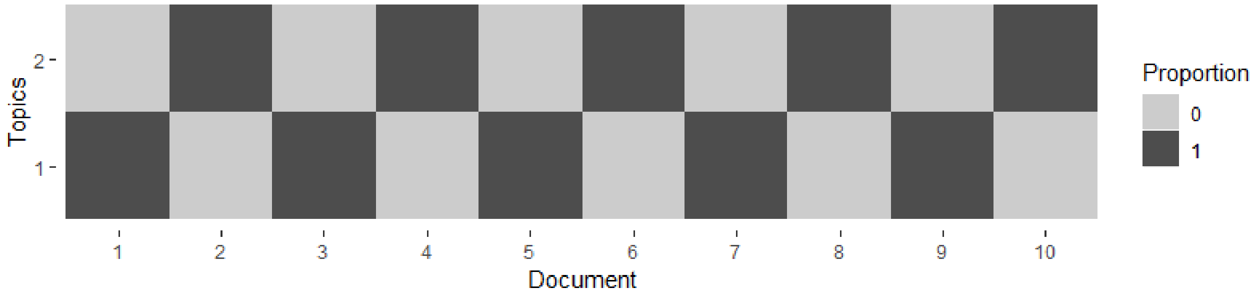

Figure 3.3: Document-topic distribution of Pilot Study

We select the true topics such that the token ‘1’ has the highest probability in both topics, and the second highest probability token is ‘2’ in topic 1 and ‘3’ in topic 2. Also, the token ‘2’ and ‘3’ has the lowest probability in topic 2 and 1, respectively. Hence, the topic model should be able to detect the difference between topic 1 and 2 based on the token ‘2’ and ‘3’. The token ‘4’ and ‘5’ are between the second highest and the lowest probability and hence are considered potential tokens that could “confuse” or “confound” the results.

Figure 3.3 illustrates Θ, the document-topic distribution we actually used in the pilot study. We selected the 10 documents so that each document solely depends on only one of the topics. In other words, document 1 is constructed based on true topic 1, which is illustrated in Figure 3.2 on the left, and document 2 is constructed based on true topic 2, which is illustrated in Figure 3.2 on the right.

We considered several document-topic distributions for the pilot study. The target of the pilot study was to evaluate how the LDA model would perform under scenarios that were not quite the best case. We chose this specific document-topic distribution since it was close to the best case and avoided possible confounding of factors. More specifically, we designed the pilot study in this way because the target was to examine how well the LDA model works under the circumstance that the true topics basically differ on two tokens (token ‘2’ and ‘3’) and have two other tokens that possibly confound the difference (token ‘4’ and ‘5’, sometimes denoted disturbances). In real data sets, there are usually multiple main difference and disturbance sets of tokens. By the nature of topics, usually researchers only care about the tokens that possess high probabilities and tend to ignore tokens with small probabilities (the ‘tail’ of the distribution). Hence, we would hope the model will identify token ‘2’ and ‘3’ properly in the correct topics.

We also set the document length to be 50 for our 10 documents. There are two reasons for this:

- Real documents' lengths ae usually less than 10 times the vocabulary size. The vocabulary size tend to increase as the number of documents increases, especially when the corpus contains documents from different fields(i.e. politics, statistics, and cooking). English has at least 171476 distinct words [10] and a typical document contains far less words. A published peer-viewed paper is "typically 3000 to 10000 words in length" [2]. Twitter and most social media texts contains only a few hundred words maximum. Hence, ten times the size of the vocabulary is a relatively large length of documents.

- When the sample size is small, the probability that in a random experiment, a specific event of interest not occurring can be relatively high. In our case, the probability of at least one token not being contained by a document is high if we have short documents. There are smoothing methods that are designed to solve this type of problem in real world applications. But since the target is to evaluate the performance of the topic model, we would like to exclude these small sample size factors from the simulation.

3.3.2 Data Generation

In an LDA model, documents are assumed to be generated through the following steps as mentioned in Chapter 2:

1. Choose K, the number of topics in the collection; Choose α and β, the hyperparameter. Choose V, the size of the vocabulary

2. Draw $\varphi_k \sim Dir(\beta), k = 1, ..., K$, for each k, φ_k is a V dimensional multinomial topic-word distribution.

3. For each document d in the corpus:

   (a) Draw $\theta_d \sim Dir(\alpha)$ which is the parameter for the multinomial document-topic distribution. This θ_d determines how topics are mixed within any specific document.

   (b) For each word w_i in document d:

      i. Draw a topic index $z_i \sim Multinomial(\theta_d)$.

ii. Draw a word $w_i \sim Multinomial(\varphi_{z_i})$, which is the topic generated in the beginning of the process.

In our pilot study, $K = 2$, Φ and Θ are known. Hence, we may discard some unnecessary steps and use the generative process as below:

For the $j - th$ word in the $i - th$ document:

1. Draw a topic index z from the multinomial distribution that takes parameters from the $i - th$ row in the document-topic matrix Θ.

2. Draw a word-token from the multinomial distribution that takes parameters from the$z - th$ row in the topic-word matrix Φ.

Here are typical data that were generated in the pilot study:

```
[1] "1 2 4 2 2 3 1 3 1 4 5 1 1 1 1 4 3 4 4 1 1 1 1 3 4 2 4 2 1 1
  3 1 1 5 1 1 2 1 2 4 1 1 2 2 2 1 2 5 1 4"
[2] "3 4 3 1 1 3 1 1 5 5 1 1 4 1 1 2 5 3 1 3 1 1 3 1 1 3 1 3 3 2 5
 3 1 5 3 5 1 1 2 1 1 1 3 4 3 5 1 1 4 1"
[3] "2 4 1 2 4 4 4 1 2 1 2 1 4 2 4 4 1 4 4 1 1 4 4 2 1 1 5 2 2 2 3
 5 4 1 2 4 1 2 3 4 1 3 1 1 1 4 1 1 1 4"
[4] "3 5 1 1 1 1 1 5 3 1 1 5 5 3 3 3 1 5 3 5 5 5 3 4 3 4 4 1 5 1 5
 5 3 1 3 1 1 1 3 1 1 2 1 3 1 1 1 2 5 1"
[5] "2 5 1 2 1 2 1 1 4 3 2 2 5 1 1 1 1 4 2 1 3 2 1 1 2 4 2 2 1 2 1
 5 2 1 1 1 4 2 1 1 2 1 2 4 1 4 1 1 2 1"
[6] "1 1 1 1 3 1 3 5 1 5 1 1 1 4 4 1 1 5 3 4 2 5 5 1 5 3 1 1 1 3 1
 4 1 1 4 3 1 4 4 3 1 3 3 1 1 1 4 3 4 5"
[7] "1 2 2 5 1 1 1 2 1 2 1 5 1 2 3 4 2 1 5 2 3 2 5 1 1 4 2 2 4 1 2
 4 4 2 2 4 4 1 1 4 1 1 4 1 1 1 2 1 1 1"
[8] "3 1 2 5 1 3 1 3 3 5 4 4 5 3 1 1 1 1 1 1 5 1 3 1 1 2 1 2 3 5 1
 5 1 5 1 3 1 1 5 1 1 5 1 1 1 2 4 5 3 5"
[9] "4 2 2 4 4 1 1 5 1 5 4 4 1 3 1 1 1 4 1 5 4 1 1 3 4 1 1 2 2 4 2
 4 1 4 5 4 2 2 4 2 1 4 5 5 1 1 4 1 2 4"
[10] "1 1 3 1 1 5 5 1 2 1 3 1 3 3 3 4 5 5 5 5 3 1 5 1 4 1 1 1 1 3
```

```
 1 1 2 5 1 1 1 1 1 2 4 1 1 1 1 3 1 3 1 2"
```

The code is provided in the Appendix.

3.3.3 Parameter selection

The R package **topicmodel** was used to fit the LDA model. The package provide two methods to estimate the topics: variational inference and Gibbs sampling. As stated in Chapter 2, we selected Gibbs sampling to fit the model. The default α and β values are used since those are the most widely selected hyperparameter settings. The number of fitted topics is set equal to 2, the number of true topics in the corpus.

For the Gibbs Sampler, the first 4000 iterations are discarded (burn-in period) and then every 500 (thinner) iteration is returned for a total of 2000 iterations.

3.3.4 Analysis of the Results

3.3.4.1 Model Fitting Result

Table 3.1 and Figure 3.4 present the fitted topic-word distributions. Table 3.2 and Figure 3.5 present the fitted document-topic distributions.

The overall observation is that it is a really bad fit. Some specific observations are:

1. Fitted topic 1 successfully found token 1 as the highest probability token and the estimated probability doesn't differ much from the truth.

2. Fitted topic 1 found tokens 1, 3, and 5 as the top 3. This indicates the fitted topic 1 is consistent with true topic 2 in term of the top tokens.

3. Fitted topic 2 picked tokens 1, 2, and 4 as the top 3. This indicates the fitted topic 2 is consistent with true topic 1 in the term of the top tokens.

4. Although the top 3 tokens are correctly selected in the fitted topics, the order of those terms is not in the correct order.

5. The estimated document-topic distributions successfully indicate the topics included in the documents, although the probabilities are still not correct. For example, in Document 1, the estimated topic distribution is 0.41 from fitted topic 1 and 0.59 from fitted topic 2. The actual

Fitted Topic	**Term**	**Probability**
1	1	0.425
1	2	0.000354
1	3	0.266
1	4	0.000354
1	5	0.308
2	1	0.220
2	2	0.362
2	3	0.00961
2	4	0.408
2	5	0.000458

Table 3.1: Pilot Study Fitted Topic-word distributions

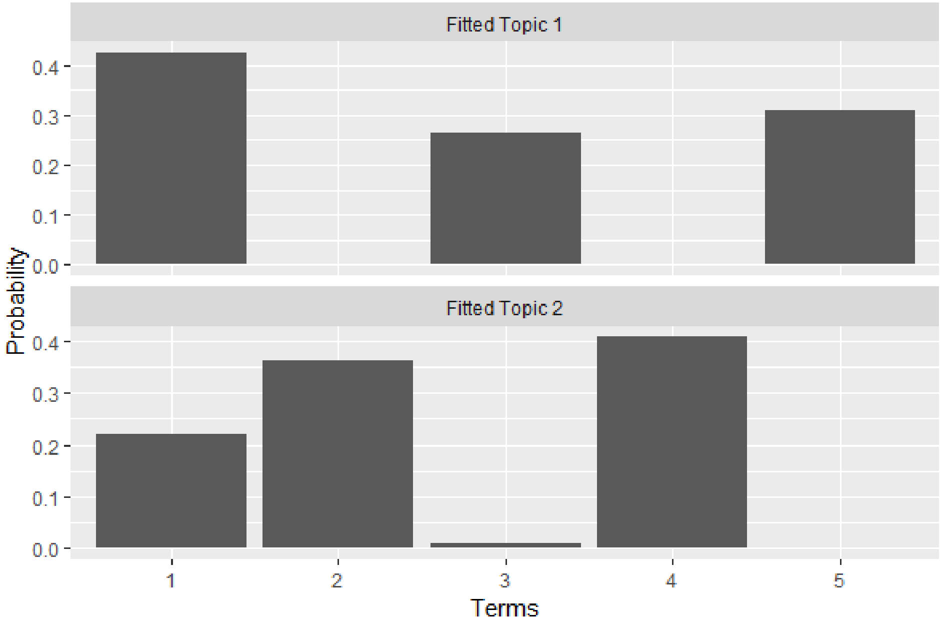

Figure 3.4: Pilot Study Fitted Topic-word distributions

Document	**Fitted Topic 1**	**Fitted Topic 2**
1	0.41	0.59
2	0.65	0.35
3	0.46	0.54
4	0.65	0.35
5	0.39	0.61
6	0.60	0.40
7	0.43	0.57
8	0.65	0.35
9	0.45	0.55
10	0.63	0.37

Table 3.2: Pilot Study Fitted Document-topic distributions

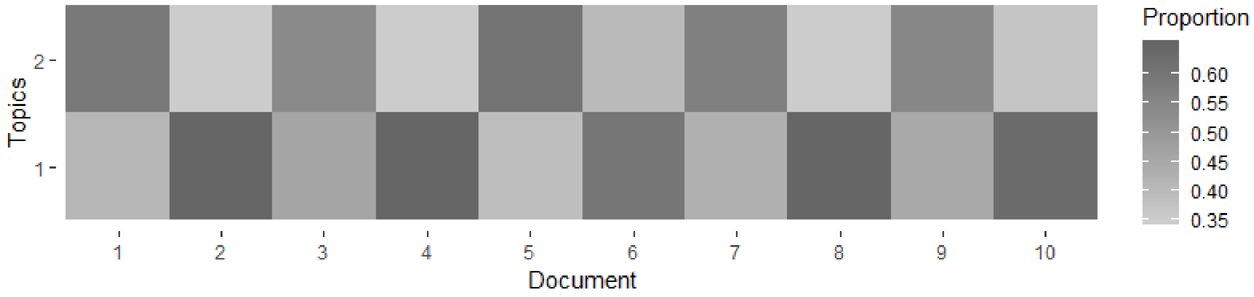

Figure 3.5: Pilot Study Fitted Document-topic distributions

distribution is 0 for true topic 2 and 1 for true topic 1, so while the overall classification is correct, the value of the probabilities are not correct.

6. The estimated document-topic distributions are also not correct. For example, in documents 1 and 3, the truth is that they both are constructed from only true topic 1. However, the estimated distribution for document 1 is 0.41 from fitted topic 1 and 0.59 from fitted topic 2; the estimated distribution for document 2 is 0.46 for fitted topic 1 and 0.54 for fitted topic 2.

It is difficult to define a proper overall metric that can quantify all of these various deviations between the estimates and the true values. This is partly because the purpose of using the modeling is not always exactly clear. Some of the possible purposes of using topic models include:

- Classification: In this example, the model works well for classifying the documents. But since our document-topic distribution was selected as an almost best possible scenario (only two topics, and one-half of the documents were 100 percent topic 1 and the other half was 100 percent topic 2). Classification as the evaluation metric works well in this scenario, but would not be useful in many other real-life scenarios. For example, when we have document-topic distribution that is uniform across all topics, classification is not even possible.

- Topic reproduction: In this example, the model does not work well for reproducing the true topic-word distributions.

- Prediction: Perplexity is the classical metric to measure the prediction quality. Although we do not have a training set and test set in this example, we can generate another 10 documents using the same parameters and process. The perplexity computed in this way is about 4.2. Compare this value with the uniform distribution (assigning $1/N$ as the probability to each term in the vocabulary), and the expected perplexity is 5 if the vocabulary size is 5. As we mentioned in Chapter 1, the perplexity value depends on the test set and the method of estimation.

- Finding the most important tokens in each topic: Sometimes the order of the top tokens are not of the interest. For example, a topic that has the highest probability word “statistics” and the second highest probability word “compute” is possibly not different from the topic that has the highest probability word “compute” and the second highest probability word “statistics”. Under this situation, one may treat the top probability words as a set instead of an ordered list.

	Topic	Token	True beta	topic	term	rank	LDA Fitted 1	LDA Fitted 2
1	True Topic 1	Term 1	0.4	1	1	1	1	3
2	True Topic 1	Term 2	0.25	1	2	2	4	2
3	True Topic 1	Term 4	0.2	1	4	3	5	1
4	True Topic 1	Term 5	0.1	1	5	4	2	5
5	True Topic 1	Term 3	0.05	1	3	5	3	4
6	True Topic 2	Term 1	0.4	2	1	1	1	3
7	True Topic 2	Term 3	0.25	2	3	2	3	4
8	True Topic 2	Term 5	0.2	2	5	3	2	5
9	True Topic 2	Term 4	0.1	2	4	4	5	1
10	True Topic 2	Term 2	0.05	2	2	5	4	2

Table 3.3: Pilot Study result Summary

Hence, the result depends on the numbers of important tokens of interest. The LDA model works perfect if top-3 tokens for each topic are of interest. But for more tokens of interest, the model preform poorly. Table 3.3 and Figure 3.6 illustrate this.

3.3.4.2 Matching Fitted Topics and True Topics

One problem that is identified in the pilot study is the method to match the fitted topics and the true topics. Since the topics generated by the LDA model does not robustly relate to the true model, a method to match the truth and the estimation is needed.

In our pilot study, from the estimated document-topic distributions, it is relatively easy to match the fitted topics with the true topics. This is because the true document-topic distributions provided a guideline for the topics. However, if the two topics are split 0.5 and 0.5 within each document in the corpus, this guideline will disappear. Hence, the document-topic distribution is not useful in matching topics in general.

One possible method for matching the topics is by comparing the top terms of topics. Consider the Table 3.4:

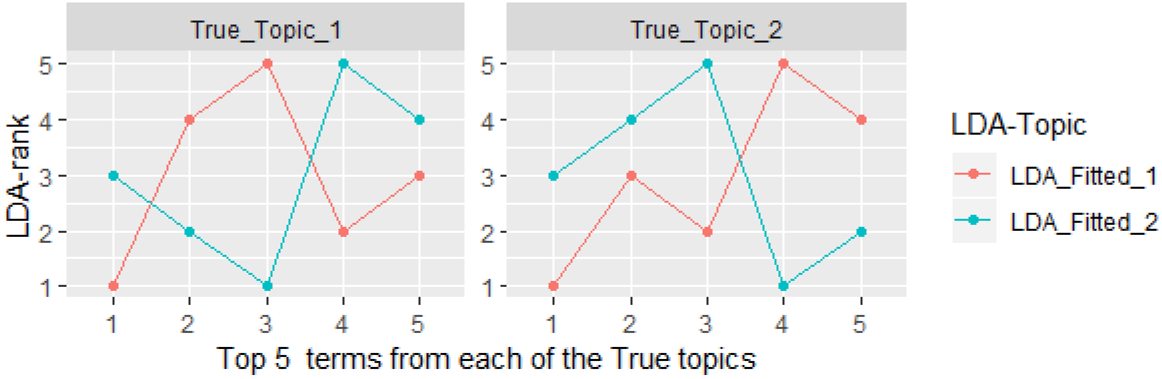

Figure 3.6: Rank plot of Pilot Study Fitted Topics

Token	Rank	Fitted Topic 1	Fitted Topic 2
1	1	1	3
2	2	4	2
4	3	5	1
5	4	2	5
3	5	3	4

Table 3.4: Matching Fitted topic with True topic by rank

The column denoted “Token” is the index of the 5 tokens in the vocabulary. The column denoted “Rank” is the true rank of each term in probability. “Fitted Topic 1” is the rank of tokens within the first fitted topic and the “Fitted Topic 2” is the rank of the tokens in the second fitted topic. Note that the topic number is not important. One approach is to start by picking a number of tokens of interest, or “level” that we want to use. Suppose level equals to 3 is chosen. Then the table is truncated after the 3rd row, leaving only tokens that have true rank less than or equal to 3. Next, we can compute how many ranks in the fitted columns are less than 3, and compute this percentage. For Table 3.4, fitted topic 1 has a percentage of 33.33% and fitted topic 2 has a percentage of 100%. The higher the percentage, the more that specific fitted topic is based on the tokens of interest . Figure 3.7 illustrate this method when the top 3 tokens are selected. It is clear that LDA2 matches true topic 1 and LDA1 matches true topic 2.

But the obvious question becomes how to pick the “level” to use.

For a simple case like this simulation, this might be easy. But in general, there is no universal algorithm to pick the ”best” level. In fact, some levels are really misleading. For example, if we pick level equals to 1 in this simulation for topic 1, the fitted topic 1 produce a 100% and the fitted topic 2 produce a 0%, which leads to matching the fitted topic 1 with the true topic 1. Figure 3.8

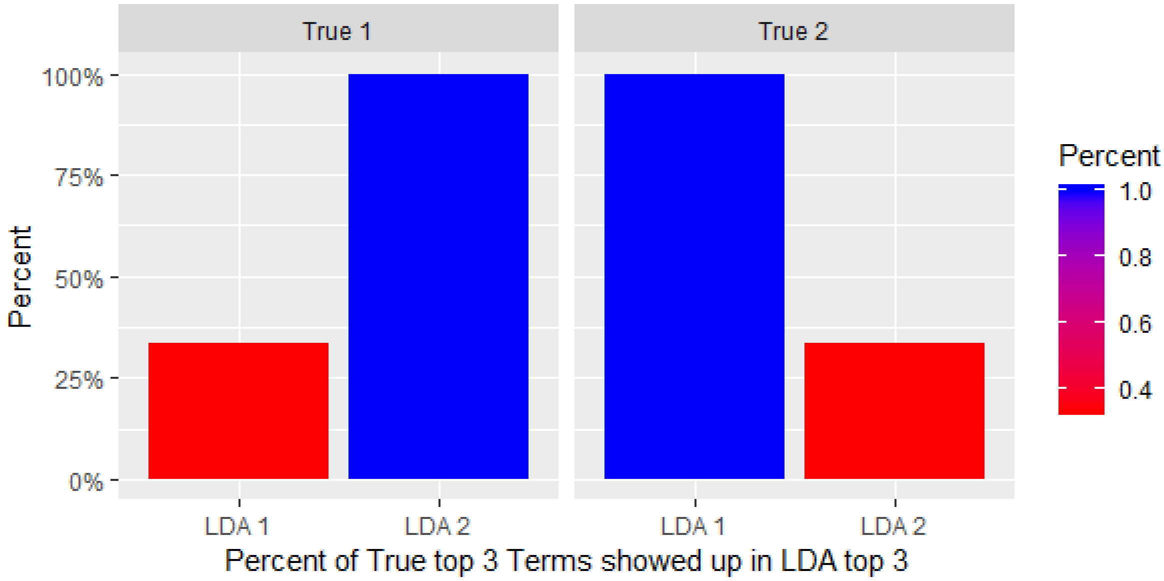

Figure 3.7: Match Topics by percentage of rank-matching

illustrate the level selection problem. From the right hand side of Figure 3.8, we can find that when level is selected to be 1, the fitted topic 1 (denoted as “LDA1”) produces a 100% matching to the true topic 1 and the fitted topic 2 produces a 0% matching, and we may match the fitted topic 1 with the true topic 1. When the level is selected to be 2, both the fitted topic 1 and the fitted topic 2 produce 50% matching and one can not figure out how to match the true topics with the fitted topics.

3.3.5 Correlation

Correlation is widely used in measuring the linear relationship between two variables. With the simulated date, the distribution is known, so it is possible to compute correlations between the true and the fitted values as follows.

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{3.1}$$

The main advantage of correlation is the ease of interpretation. There is no need to derive a new metric that maybe hard to interpret. On the other hand, the correlation coefficient may be misleading in long-tailed topic distributions. Consider two vectors x and y; Let $x = (0.2, 0.1, 0.5)$ and

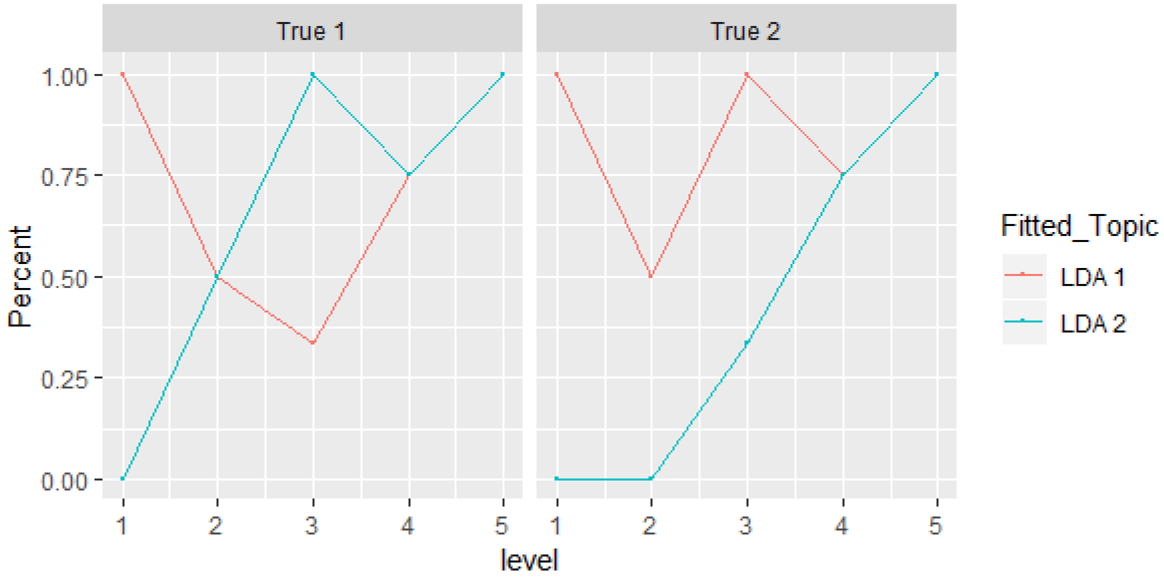

Figure 3.8: Match Topics by percentage of rank-matching - Level Selection

$y = (0.05, 0.3, 0.1)$. Let these be the probabilities of the top three tokens for the actual and estimated results of topic modeling. The correlation coefficient between x and y equals to -0.5447048. Now consider another two vectors x' and y'. For each vector we add fifteen small entries (a surrogate for the tokens with small probabilities in topic model) after the original x and y.

Let $x' = (0.2, 0.1, 0.5, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01)$ and $y' = (0.05, 0.3, 0.1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01)$. The correlation coefficient between x' and y' is now 0.4117698. In topic modeling, the long-tail distribution is frequently observed and must be taken into account.

In our experiment, correlation is applied in two steps:

1. Topic Matching
2. Quality Evaluation

For topic matching, to avoid the problem demonstrated above, we compare the correlation coefficient between each true topic and fitted topic, instead of setting a threshold for matching. Using the simulation data, the correlation coefficients are shown in Table 3.5.

Note that the Kullback-Laibler divergence as mentioned in Table 1.3 in Chapter 1 is a strong candidate in searching for the best possible metrics. But the KL-divergence is not commutative: for

	LDA 1	LDA 2
True 1	0.1367830	0.5797202
True 2	0.9372166	-0.4834648

Table 3.5: Pilot Study Correlation Coefficient for topic matching

any two distributions P and Q, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ and this is an issue in evaluating topic modeling results because the results of LDA model are not guaranteed to possess the same order as the true topics. There are modifications of KL-divergence but they are very hard to interpret. Note that if a version of table 3.5 is created using LK-divergence with certain assumptions, the overall conclusion would be the same. Hence, we decided to keep using correlation due to ease of interpretation.

When the number of true topics and fitted topics are the same, we can find a one-to-one match based on comparison of the correlation coefficients. Otherwise, there might be multiple true/fitted topics that are matched to one fitted/true topic. The matching algorithm is described in Algorithm 1

```
Data: C={Corr(i,j) for all i and j} ←
      Corr(i,j)={Correlation coefficient between the i-th true topic and j-th fitted topic}
Result: Match_Result ← ()
while $len(C) > 0$ do
    $M$ ← Maximum in C;
    $i$ ← Associated index of true topic for M;
    $j$ ← Associated index of fitted topic for M;
    Match_Result=Match_Result $\cup (i, j)$;
    $C = C \setminus$ {Corr(i,m) and Corr(m,j) for all m};
end
```

Algorithm 1: Topic Matching

We examined this algorithm through repeatedly measuring correlation coefficient with different seeds in Gibbs Sampling. Results shown in Table 3.6 shows that the algorithm works well.

Results in Table 3.6 show that correlation is a reasonable good metric in finding proper matches between the true topic and the fitted topic.

For the purpose of quality evaluation, the “long-tail” problem stated above suggest that a large value of correlation coefficient might not always indicate good model fitting. However, a small value of correlation coefficient always indicates a bad the model fitting.

In this chapter, we examined the structure and the assumptions of the LDA model in detail,

	LDA 1	LDA 2
True 1	0.1367830	0.5797202
True 2	0.9372166	-0.4834648
	LDA 1	LDA 2
True 1	0.5932316	0.1238470
True 2	-0.4758891	0.9381834
	LDA 1	LDA 2
True 1	-0.02451539	0.7453396
True 2	0.88443979	-0.2802809
	LDA 1	LDA 2
True 1	0.4035011	0.2164524
True 2	0.9777141	-0.7449990
	LDA 1	LDA 2
True 1	0.08862987	0.6338065
True 2	0.92626899	-0.4275508
	LDA 1	LDA 2
True 1	-0.07991428	0.7857483
True 2	0.85956584	-0.2172482
	LDA 1	LDA 2
True 1	-0.4531991	0.9289798
True 2	0.6087086	0.1013433
	LDA 1	LDA 2
True 1	-0.001858171	0.7264248
True 2	0.893780349	-0.3077362
	LDA 1	LDA 2
True 1	0.8466753	-0.1835992
True 2	-0.1109437	0.8065989
	LDA 1	LDA 2
True 1	0.90153719	-0.3395190
True 2	0.02155069	0.7008019

Table 3.6: Pilot Study Correlation Coefficient Topic Matching Rep

and conducted a pilot study to find the proper range of the parameters for the simulations in the next chapter.

Chapter 4

Simulation Study

A simulation study was conducted to determine the impact of four factors on the ability of topic modeling to find the true underlying topics. The four factors considered in this study were: 1) size of the vocabulary, 2) document length ratio, 3) topic-word distribution, and 4) document-topic distribution. The first three factors all depend on the overall topic structure. To define the overall topic structure we will first discuss the prior topic-word distribution, then the size of vocabulary, and then the document length ratio. Both the real corpus data and the simulated data will be used in these discussions. Finally, the document-topic distribution will be considered. More details about the factors and the levels for the four factors will be discussed below. Figure 4.1 illustrate the structure of the simulation in this section.

Prior to the complete simulation study, a pilot study was performed to determine appropriate levels of the factors. For a given document-topic distribution matrix and a given topic-word distribution matrix, ten documents are generated. The most widely used default setting of Gibbs Sampling for LDA estimation from the R package **topicmodel** was applied: burn-in period equals to 4000 iterations, picking value for every 2000 iterations after the burn-in period, and picking 2000 values in all. The number of topics is selected as the true number of topics. The corpus size is selected to be 10 documents so that the result better in interpreting. The number of true topics is selected to be two.

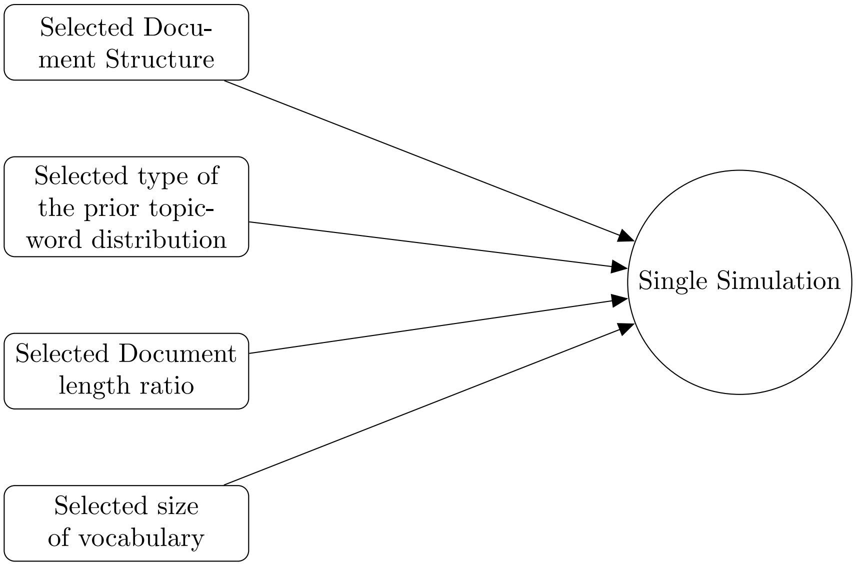
Selected Document Structure
Selected type of the prior topic-word distribution
Selected Document length ratio
Selected size of vocabulary
Single Simulation

Figure 4.1: Structure of Simulations

4.1 Overall Topic Structure

In topic modeling, the assumed components of the topic structure are very important. The models are developed based on those assumptions, and the results are dependent to those assumptions. Three of the primary characteristics of the overall topic structure of the LDA model are three of the factors in the simulation, namely:

1. Size of the vocabulary.
2. Document length ratio.
3. Type of the prior topic-word distribution.

For the target three characteristics listed above, we analyzed the four data sets below to determine if the pilot study results were reasonable.

- **Reuters21578[18]**: Currently the most widely used test collection for text categorization research. The data was originally collected and labelled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system.
- **20 Newsgroups[11]**: The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. To the

Data set	Reuters21578	20 Newsgroups	AP	aclIMDB
Type	Short News	Mail-list discussion	Full reports	Peer-Review
Number of documents	19042	19791	2246	100000
Size of vocabulary	33255	96509	10473	171770
Number of words	1520283	1237217	435838	23645581
Avg. number of words per document	79.8	62.5	194	236

Table 4.1: Basic information about the selected Corpus

best of my knowledge, it was originally collected by Ken Lang, probably for his Newsweeder: Learning to filter netnews paper, though he does not explicitly mention this collection. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. There are three versions of 20 Newsgroups data set and we are using the original 20news-19997 version.

- **Associated Press-AP[14]**: The AP data set is from the First Text Retrieval Conference (TREC-1) 1992 and contains 2246 documents. This is the data set that is utilized in the original LDA paper from Blei[4].
- **aclIMDB[19]**: This is a dataset that originally constructed for binary sentiment classification containing substantially more data than previous benchmark datasets. It provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided.

Since the pilot and the real data showed some agreement, this allowed us to go forward and develop levels for the formal simulation.

4.1.1 Topic-Word Distributions

In the pilot study, we chose the topic-word distribution so that it had the desired property. In general, we would like to have a rule for choosing the topic-word distribution so that one can generate topic-word distributions similar to the real corpus. Since the topics are represented by the probability of words being presented, a multinomial distribution is the natural choice for the topic-word distribution. But how to generate the multinomial distribution is still in question. Zipf[33] states that given a large sample of words, the frequency of any word is inversely proportional to its rank in the frequency table. So word number n has a frequency proportional to 1/n. This is usually referred as **Zipf's Law** and widely accepted by researchers. However, Zipf's law is only an empirical law summarized over the whole corpus and does not consider multiple topics. Hence, we should not fully rely on Zipf's law when creating topic-word distributions.

The Bayesian topic models such as LDA assume a prior structure over the topic-word multinomial distributions. Two most widely used priors are the Dirichlet prior and the Multi-normal prior. These two distributions are used because both simplify the computations. The Dirichlet distribution is the conjugate prior of multinomial distributions. This means that the posterior distribution is also a Dirichlet distribution when we use it as a prior distribution. Multi-normal distribution is usually used when researchers are considering the correlation between topics. The covariance matrix of the multi-normal distribution provides a natural tool to analyze the correlation structure between the topics.

Both the Dirichlet and the multi-normal distributions have drawbacks. For the Dirichlet distribution, as illustrated in Chapter 1, the concentration parameter controls how the probabilities are distributed among each categories (Figure 1.3). Griffiths[12] stated that the LDA model preforms best in term of perplexity when the concentration parameter of topic-word distributions is selected as 0.01. This criteria has been widely accepted and software packages use this as the default value. However, when the concentration parameter is selected as 0.01, more than 90 percent of the probability will be allocated to at most 5 words and other words share the rest of the probability. Figure 4.2 illustrate this situation when the size of vocabulary is set to be 100. This violates Zipf's Law and hence is not a proper method to generate the corpus.

The multi-normal prior distribution often has results that are hard to interpret, and samples from a multi-normal distribution are not guaranteed to have a sum of one. This is not a problem during the fitting process of topic modeling since the Gibbs Sampling process only takes kernels into consideration. A normalization step is required if we want to use the multi-normal prior distribution to generate the corpus.

We compare the performance of the Dirichlet prior distribution and the multi-normal distribution in this study. For the Dirichlet distribution, we made the assumption that the top 20

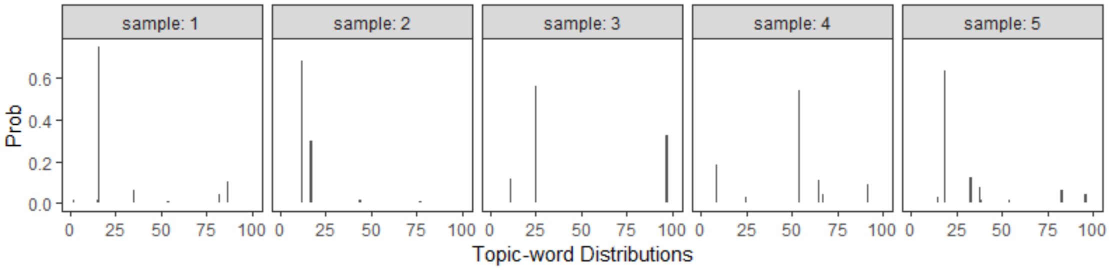

Figure 4.2: Five samples of Topic-word distributions generated from Dirichlet prior that has concentration paramter 0.01

words shares 70 percent of the probability, and the rest of words share the remaining 30 percent. These values are generated from Zipf's law. Suppose we have a document that has vocabulary size of 100. Let $[w_1, w_2, \ldots, w_{100}]$ be the list of words ordered by frequency. i.e., w_1 is the word that appears the most times in the document, w_2 is the word that appears the second most times in the document. Let $[n_1, n_2, \ldots, n_{100}]$ be the number of appearance of each word. By Zipf's Law, $n_2 = \frac{1}{2}n_1$, $n_3 = \frac{1}{3}n_1$,..., $n_{100} = \frac{1}{100}n_1$. Hence, the total number of words N in this document is

$$N = \sum_{i=1}^{100} \frac{n_1}{i} \approx 5.187378 n_1$$

Hence, we can compute the proportion of the summation from w_1 to w_{20}:

$$Proportion = \frac{\sum_{i=1}^{20} \frac{n_1}{i}}{N} \approx 0.7$$

In general, for a Dirichlet prior, we first randomly select 20 words from the whole vocabulary. A Dirichlet distribution with concentration parameter 1 is used to generate the probabilities associated with each of these 20 words. Then these probabilities are normalized through multiplying 0.7. The probabilities of the rest of words are then generated from the Dirichlet distribution with concentration parameter 1. These probabilities are normalized through multiplying 0.3. We are generating documents that are mostly following Zipf's empirical law with Dirichlet distributions.

For the multi-normal prior distribution, we assume no correlations between topics and the covariance matrix is a diagonal with variances all equal to 1. A more disperse multi-normal distribution is also possible to be considered in some of the simulations where the variance are equal to 10.

Data set	Min	Q1	Median	Mean	Q3	Max
Reuters21578	0.1418	0.1797	0.1909	0.1919	0.2024	0.3194
20 Newsgroups	0.1045	0.1231	0.1297	0.1345	0.1392	0.4384
AP	0.1421	0.1710	0.1774	0.1776	0.1840	0.2186
aclIMDB	0.2015	0.2398	0.2517	0.2529	0.2650	0.3590

Table 4.2: Sample Corpus Basic Descriptive Statistics of the Document Length Ratio

4.1.2 Document length ratio

4.1.2.1 Analysis of Real Corpus

Since the simulated corpus contains 10 documents, we used the data sets listed in Table 4.1 to determine the reasonable document length ratio.

We randomly selected 10 documents from one real corpus and computed the document length ratio between the average number of words per document and the size of vocabulary for the selected 10 documents. The results are summarized in Table 4.2.

4.1.2.1.1 Reuters21578

10000 samples that contains 10 randomly selected documents were taken from the original data set. Figure 4.3 presents the fitted density plot of the sampled data.

4.1.2.1.2 20 Newsgroups

10000 samples that contains 10 randomly selected documents were taken from the original data set. Figure 4.4 presents the fitted density plot of the sampled data.

4.1.2.1.3 Associated Press

10000 samples that contains 10 randomly selected documents were taken from the original data set. Figure 4.5 presents the fitted density plot of the sampled data.

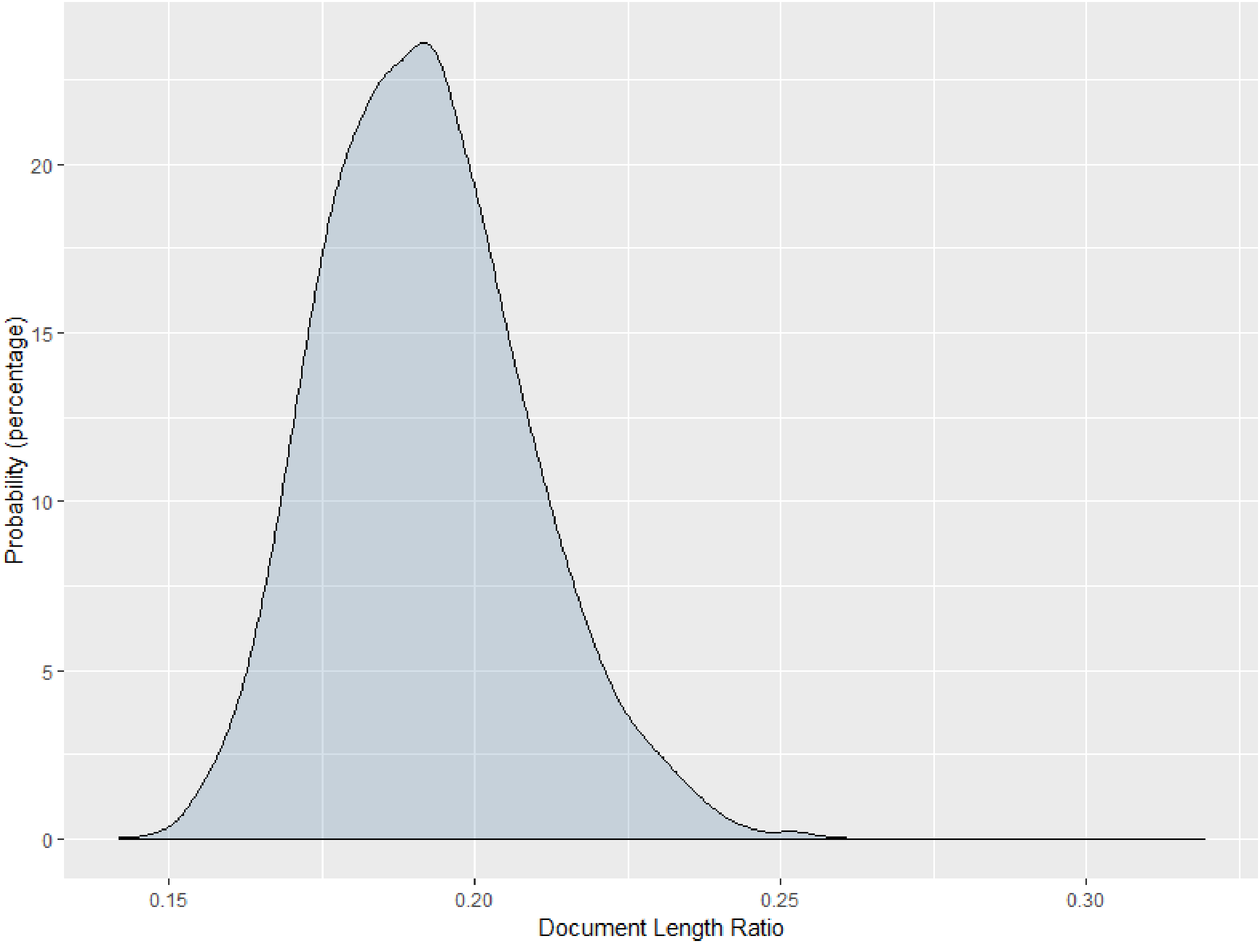

Figure 4.3: Document Length Ratio of Reuters21578 Data

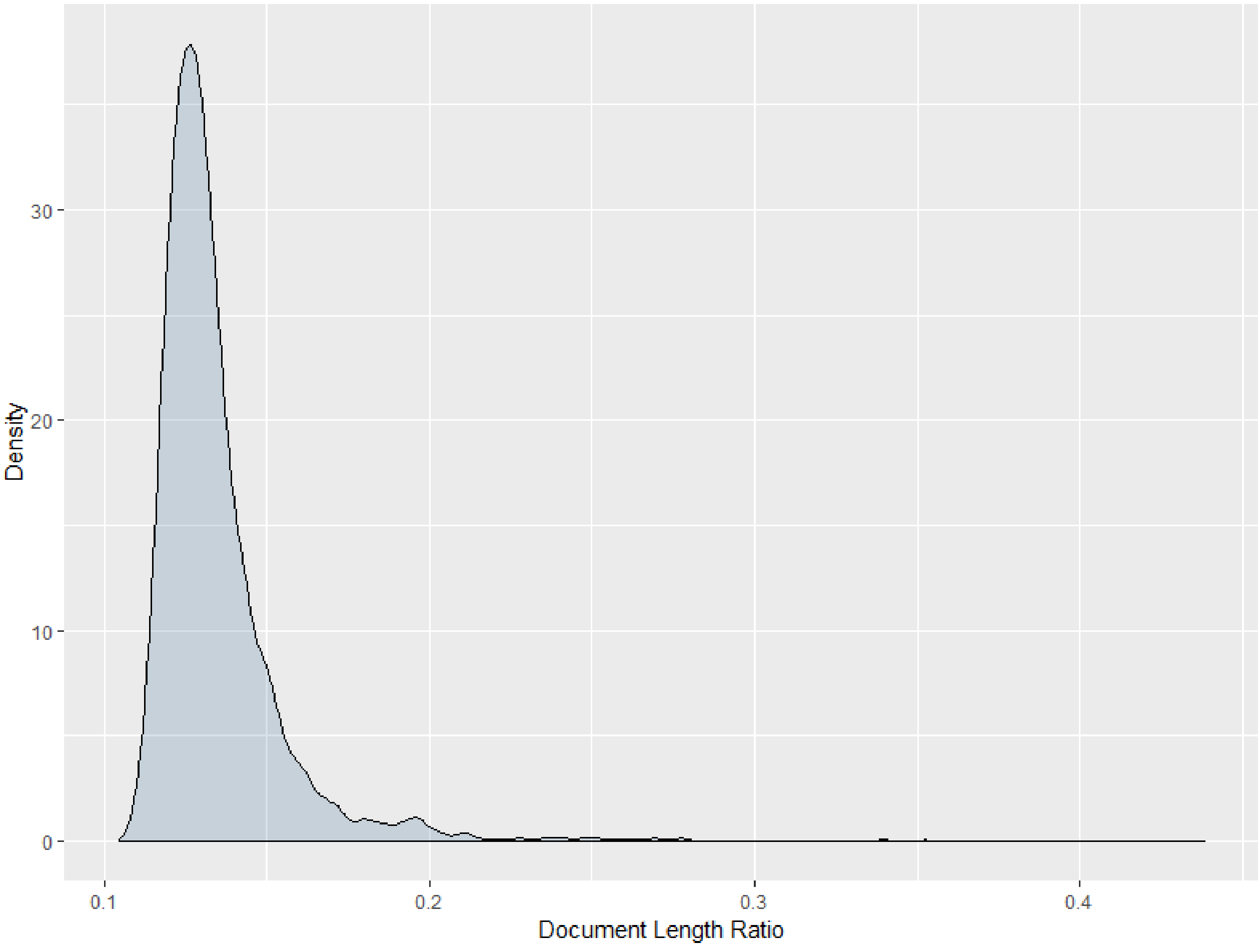

Figure 4.4: Document Length Ratio of 20 Newsgroups Data

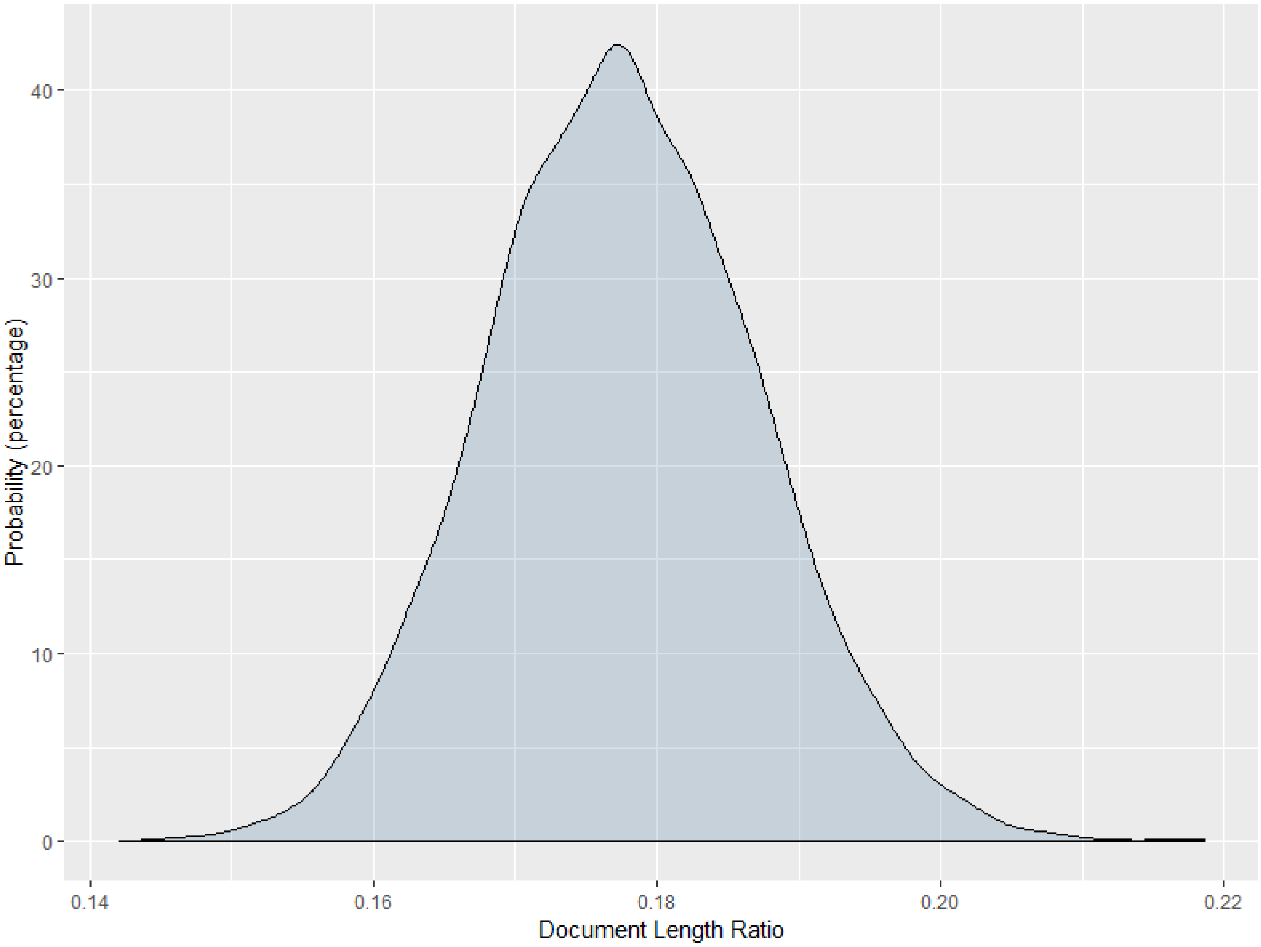

Figure 4.5: Document Length Ratio of Associated Press Data

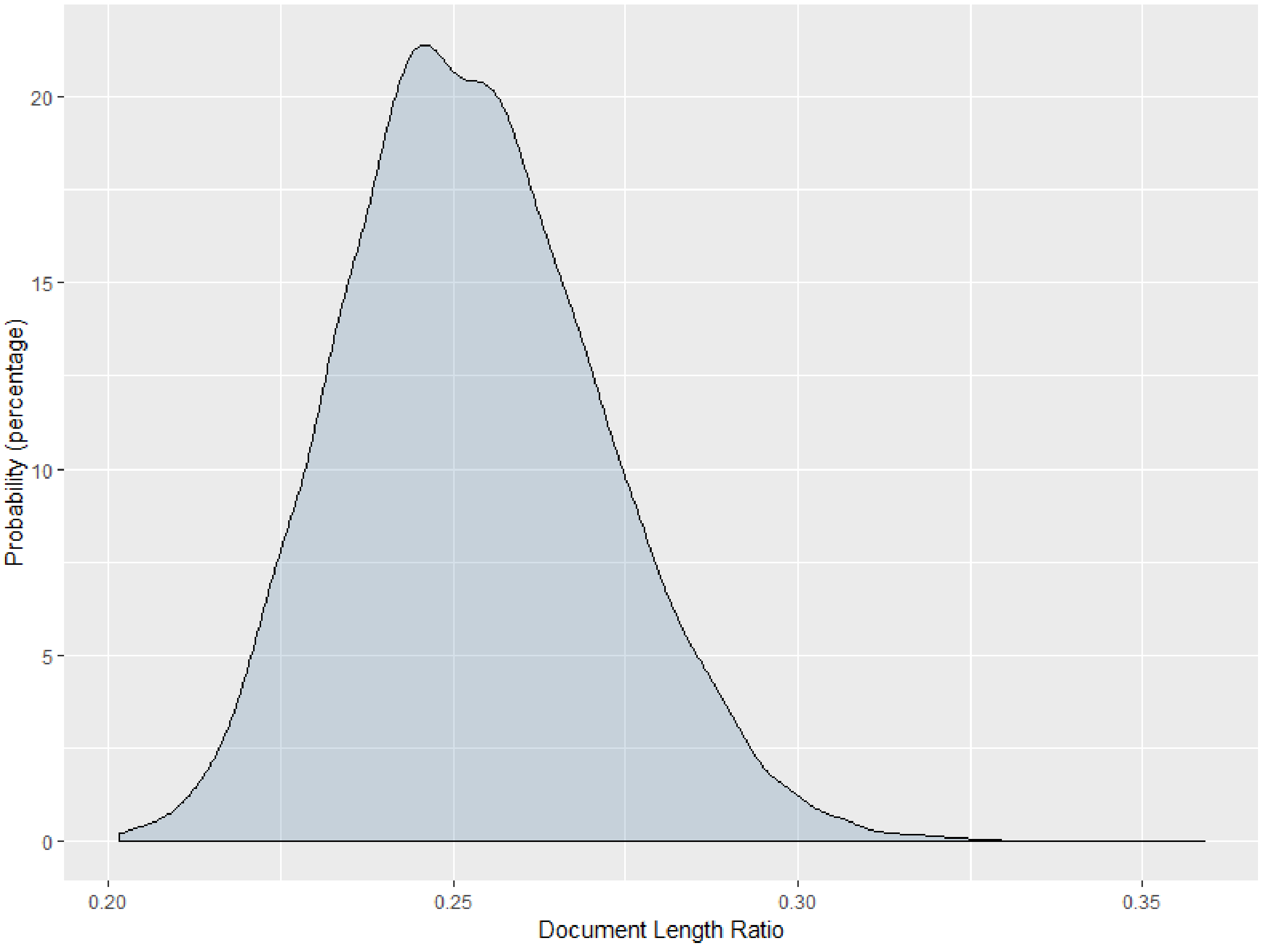

Figure 4.6: Document Length Ratio of aclIMDB Data

4.1.2.1.4 aclIMDB

10000 samples that contains 10 randomly selected documents were taken from the original data set. Figure 4.6 presents the fitted density plot of the sampled data.

4.1.2.2 Analysis of Simulated Corpus

Besides examining the general truth about the document length ratio, we also examined the quality of LDA results based on correlation between the true topics and the fitted topics for different document length ratio. Based on the previous examination of the true corpus, we noticed that the document length ratio is greater than 0.1 for the samples of 10 documents. To fully examine the effect of the document length ratio, we selected 19 levels trying to cover most of the possible scenarios:

$$[\frac{1}{10},\frac{1}{9},\frac{1}{8},\frac{1}{7},\frac{1}{6},\frac{1}{5},\frac{1}{4},\frac{1}{3},\frac{1}{2},1,2,3,4,5,6,7,8,9,10]$$

Ten documents are constructed for each level based on the following document-topic distributions:

$$\begin{array}{c|cc|} & TrueTopic1 & TrueTopic2 \\ Document1 & 1 & 0 \\ Document2 & 0 & 1 \\ Document3 & 1 & 0 \\ Document4 & 0 & 1 \\ Document5 & 1 & 0 \\ Document6 & 0 & 1 \\ Document7 & 1 & 0 \\ Document8 & 0 & 1 \\ Document9 & 1 & 0 \\ Document10 & 0 & 1 \end{array}$$

Both the Dirichlet prior distribution and the multi-normal prior distribution are considered as discussed above. The vocabulary size is selected as 100 based on the analysis in the next section.

Figure 4.7 and Figure 4.8 show the result of the simulation. We observe that the quality of LDA model in term of correlation is positively correlated with the document length ratio. The Dirichlet prior results in a better correlation for all document length ratios.

We also find that the correlation is relatively stable after a ratio value of 8. Hence, when analyzing the effect of size of the vocabulary, we will set the document length ratio to 10 in order to exclude the affect from the document length ratio.

4.1.3 Size of the vocabulary

4.1.3.1 Analysis of Real Corpus

From the basic summary statistics listed in Table 4.1, we learned that the size of vocabulary is positively correlated with the number of documents in the corpus. We first analysed the real corpus to examine this effect. After that, we drew samples of 10 documents from the corpus to find how many words should we have in the vocabulary for the simulation of documents.

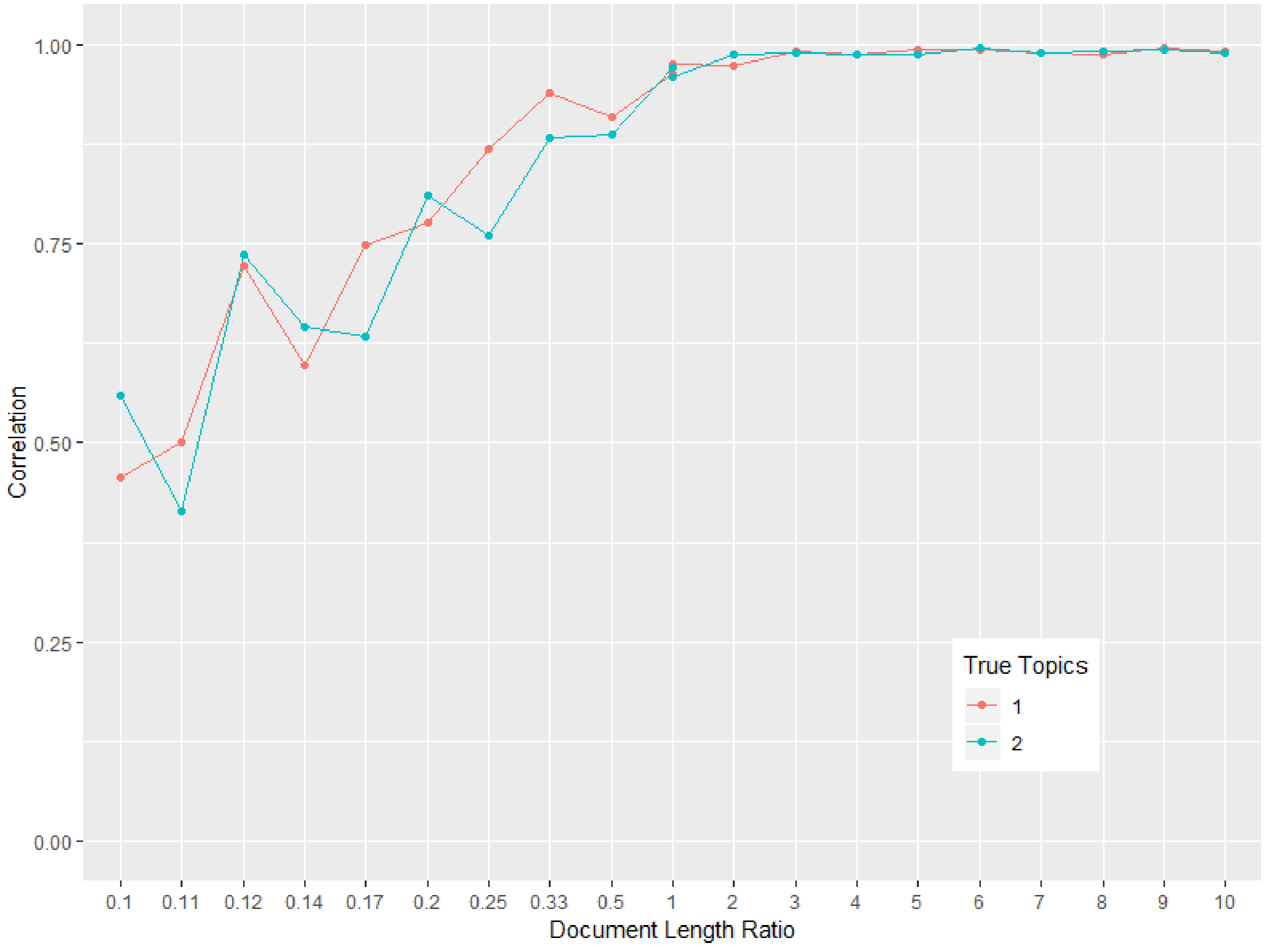

Figure 4.7: Simulated Analysis of Document length ratio for Dirichlet prior

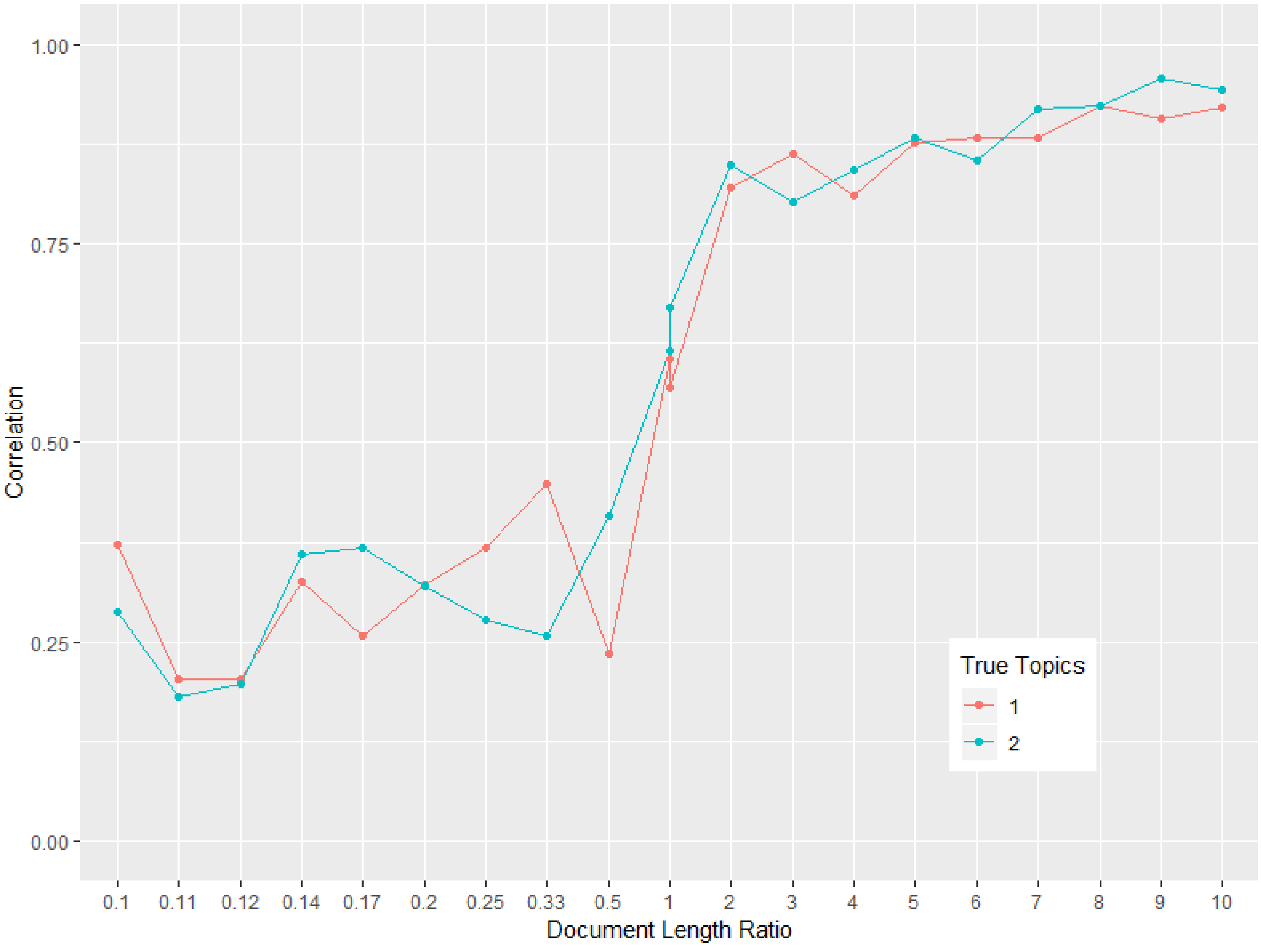

Figure 4.8: Simulated Analysis of Document length ratio for Multi-normal prior

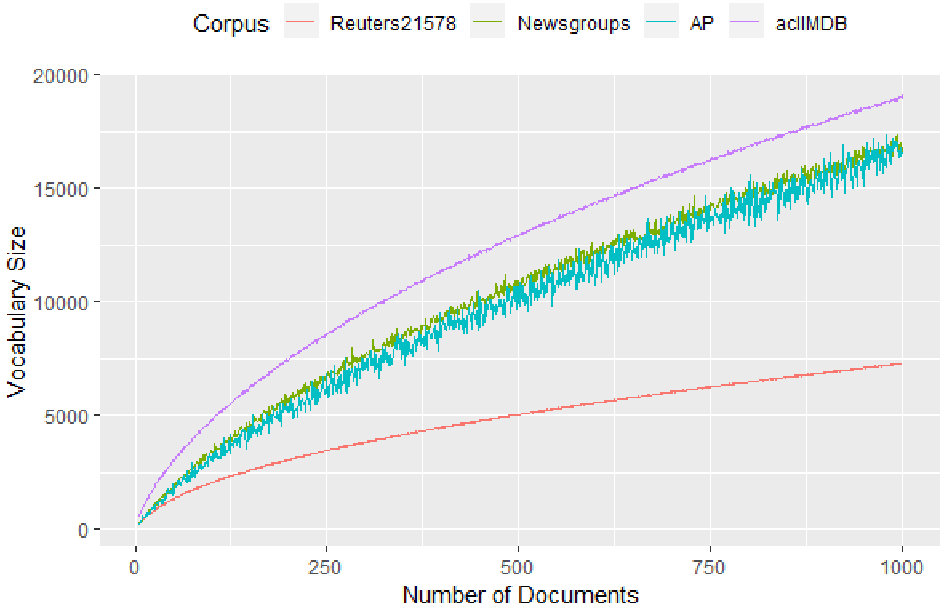

Figure 4.9: Vocabulary Size changes based on the Number of Documents for Four different Corpus

To analyse the positive correlation between the number of documents in the corpus and the size of vocabulary, we randomly drew 100 samples of n documents from each corpus and computed the mean of the size of the vocabulary. We considered n from 5 to 1000. A summary of the results is shown in Table 4.3. The result is also plotted in Figure 4.9.

We found that for a corpus of 10 documents, the size of vocabulary is less than 1000. We noticed that a higher order relationship might exist. After testing for logarithmic, exponential, and quadratic relationships, the square root of the number of documents vs the vocabulary size produced the best fit. The analysis results are shown in the following sections.

4.1.3.1.1 Reuters21578

The sampling result is illustrated in Figure 4.10. The transformed sampling result is illustrated in Figure 4.11.

The linear regression from an R basic package produced the following result:

```
Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -373.21016    1.41915    -263    <2e-16 ***
```

Documents	Reuters21578	20 Newsgroups	AP	aclIMDB
5	239	204	183	534
6	263	273	213	616
7	315	319	263	692
8	335	355	276	787
9	384	390	466	865
10	414	412	541	910
11	458	482	487	1010
12	469	488	531	1069
13	497	551	488	1126
14	528	554	574	1184
15	578	626	726	1287
16	597	707	576	1320
17	632	735	683	1362
18	650	827	625	1408
19	676	760	756	1549
20	702	779	689	1551
30	941	1239	1227	2107
40	1146	1673	1384	2563
50	1354	1939	2010	3035
60	1469	2256	1936	3473
70	1671	2488	2138	3807
80	1802	2886	2513	4141
90	1924	3245	2779	4518
100	2068	3435	3071	4797
200	3058	5618	5145	7439
300	3821	7683	7085	9582
400	4487	9263	8812	11327
500	5044	10938	10004	12919
600	5543	12206	10728	14370
700	6021	13445	12986	15620
800	6471	14718	13378	16867
900	6895	16000	15346	17920
1000	7283	16780	16565	19061

Table 4.3: Mean size of vocabulary of sample size 100

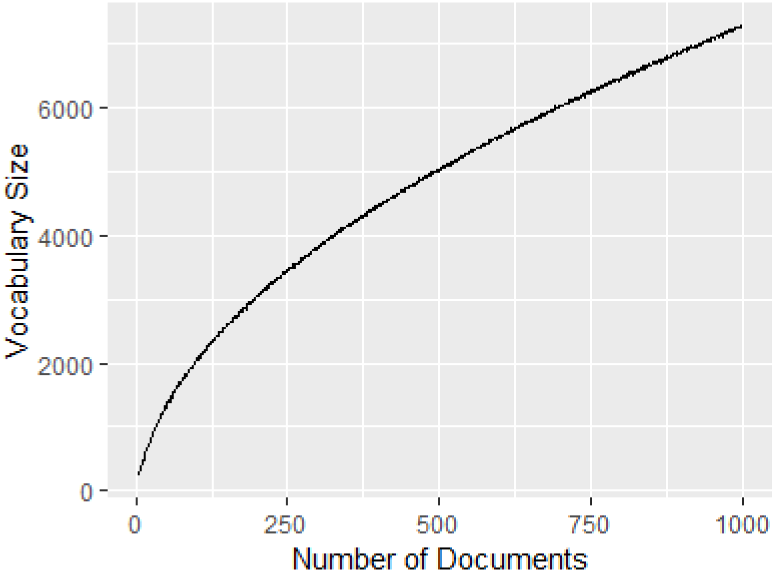

Figure 4.10: Plot of Number of documents vs Vocabulary size for sample size 100, from 5 to 1000, data set Reuters21578

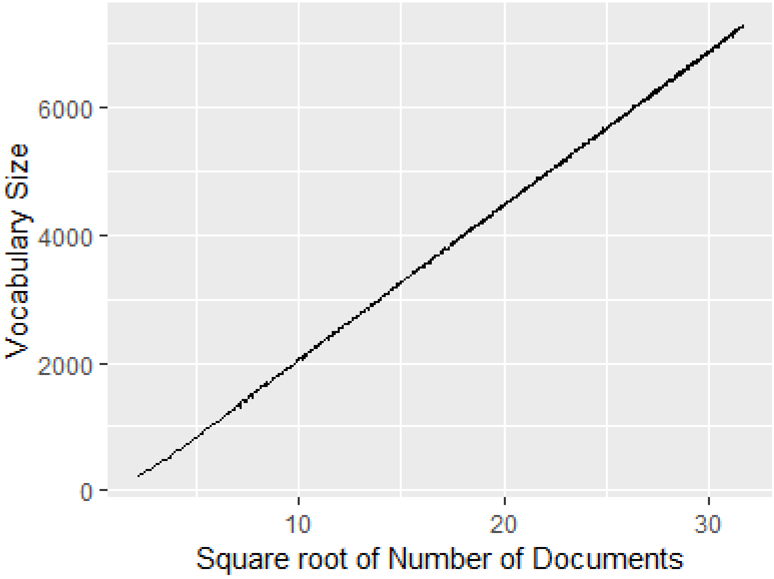

Figure 4.11: Plot of Square root of Number of documents vs Vocabulary size for sample size 100, from 5 to 1000, data set Reuters21578

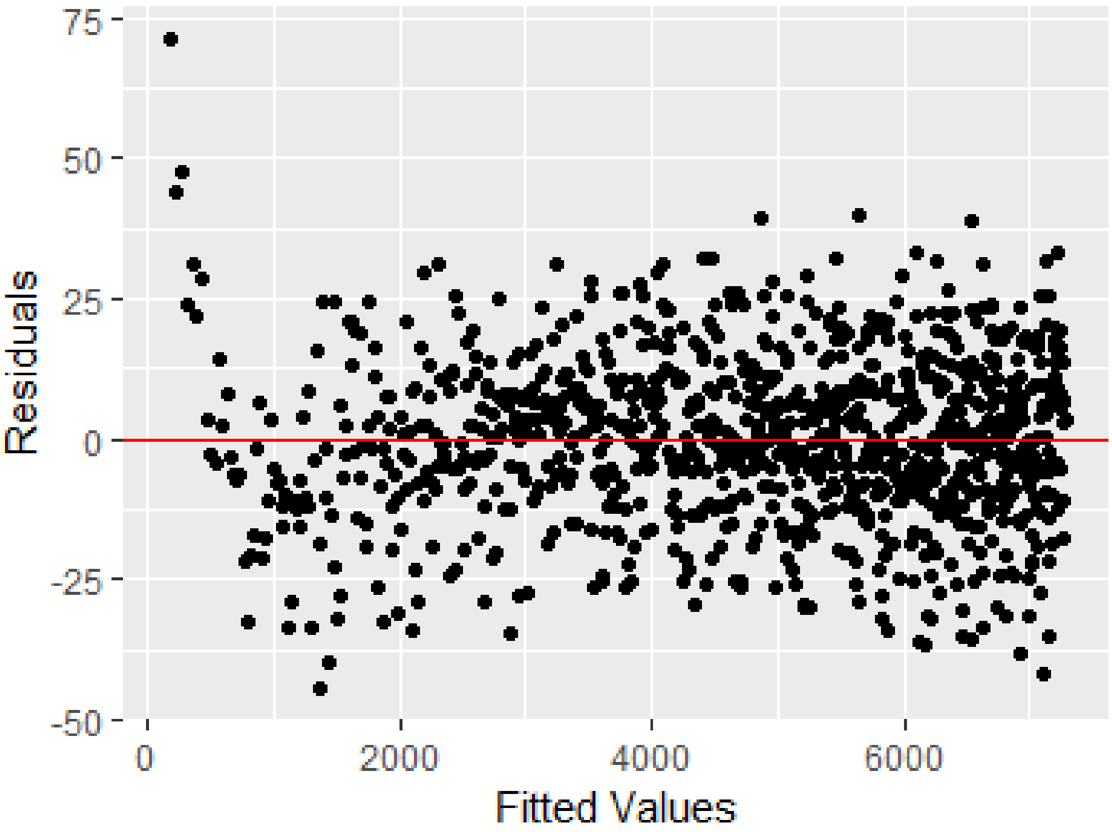

Figure 4.12: Residual Plot for Reuters21578 data

```
n_sqrt        242.01432    0.06331    3823   <2e-16 ***
```

The residual plot Figure 4.12 indicated a good fit.

Hence, the following equation can be used to estimate the size of vocabulary for Reuters21578 data. We can also examine the prediction line plot in Figure 4.13.

$$\text{Size of Vocabulary} = -373 + 242 \times \sqrt{\text{Number of Documents}} \tag{4.1}$$

From Equation 4.1, we can estimate that for a sample from Reuters21578 data set of size 10, the size of vocabulary is approximately 392.

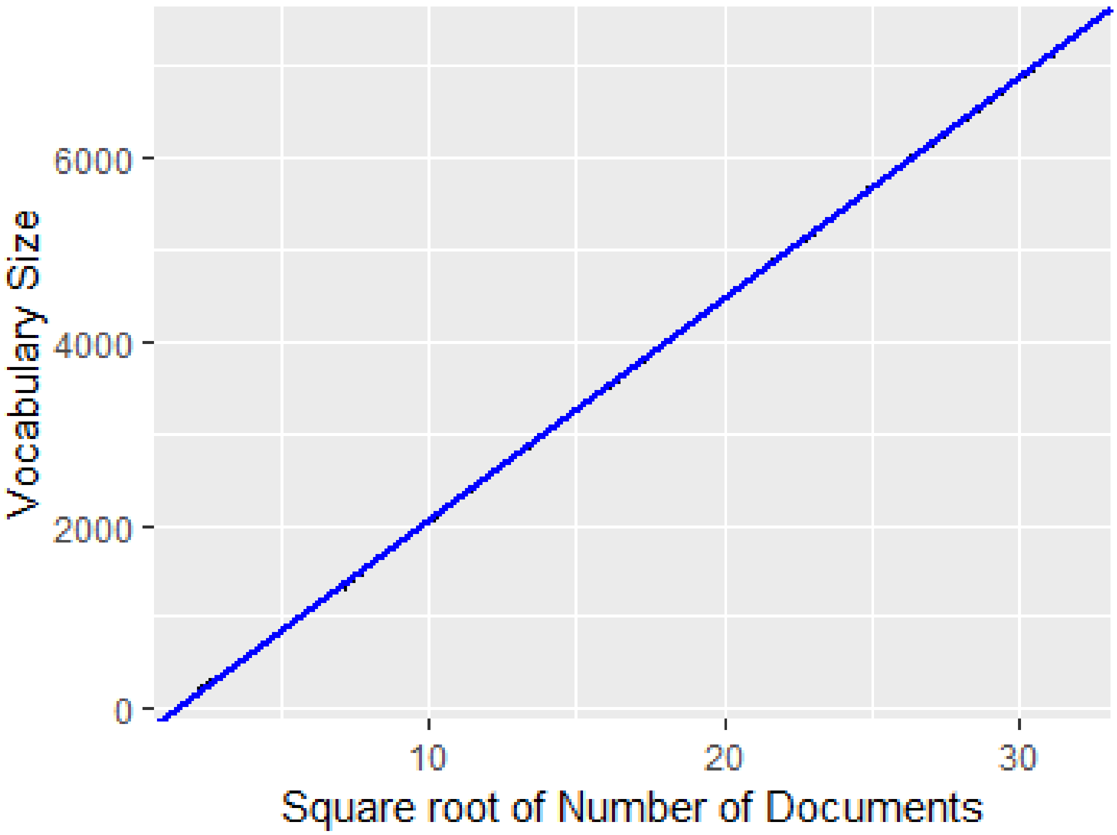

Figure 4.13: Regression Line with Reuters 21578 Data Set

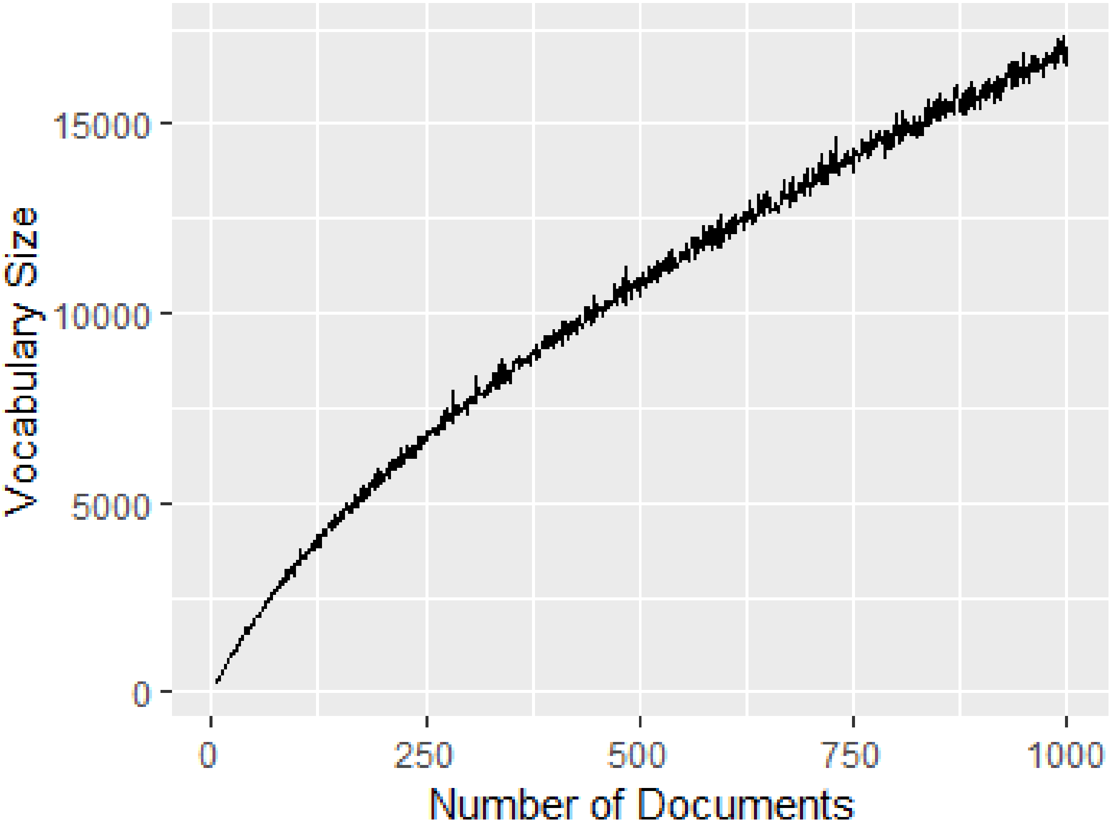

Figure 4.14: Plot of Number of documents vs Vocabulary size for sample size 100, from 5 to 1000, data set 20 Newsgroups

4.1.3.1.2 20 Newsgroups

The sampling result is illustrated in Figure 4.14. We observed that a linear model of a square root and a cubic root of the number of documents generated the best fit. The regression model is:

$$\text{Size of Vocabulary} = \beta_0 + \beta_1 \times \sqrt{\text{Number of Documents}} + \beta_2 \sqrt[3]{\text{Number of Documents}} \quad (4.2)$$

The linear regression from an R basic package produced the following result:

```
Coefficients:

Estimate Std. Error t value Pr(>|t|)
```

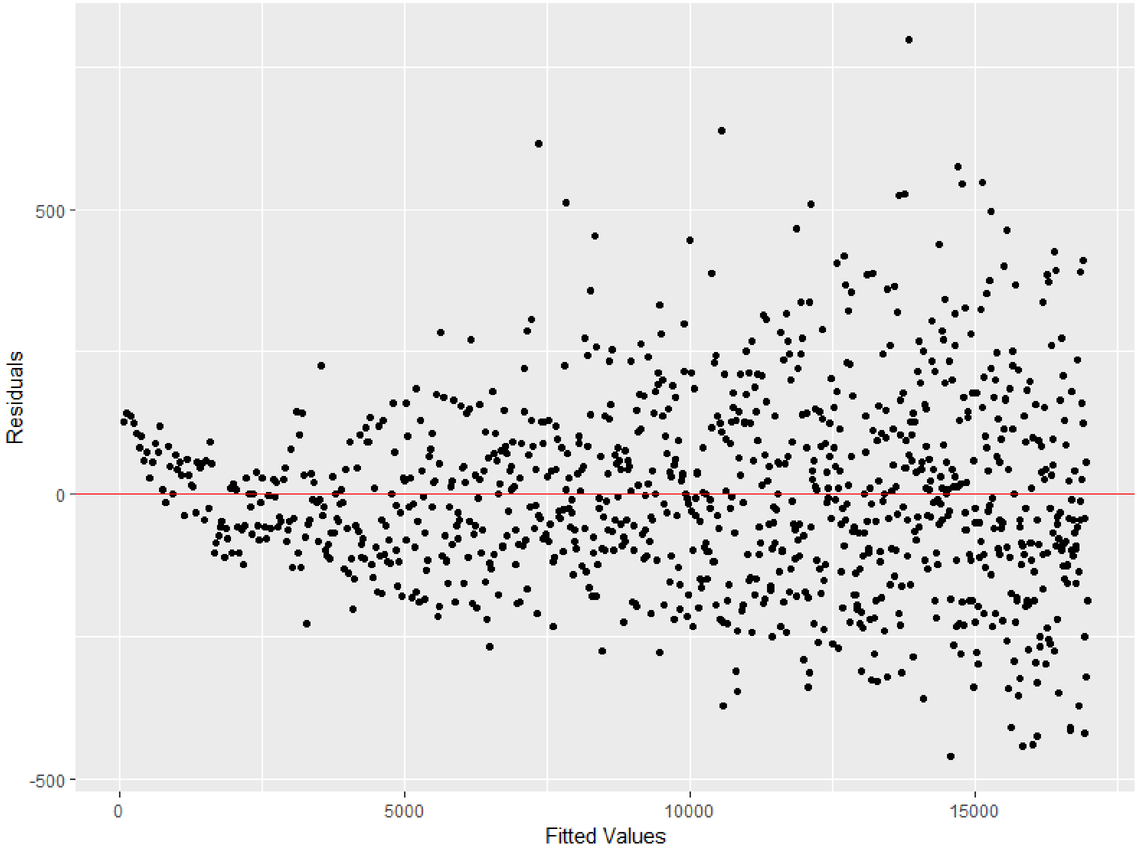

Figure 4.15: Residual Plot for 20 Newsgroups data

```
(Intercept)   432.322     73.935   5.847 6.77e-09 ***
n_sqrt       1003.229      8.876 113.024  < 2e-16 ***
n_cubic     -1519.085     34.451 -44.094  < 2e-16 ***
```

The residual plot Figure 4.15 shows that there is possibly a heteroscedasticity problem. But the regression line in Figure 4.16 shows a good fit when the number of documents is relatively large ($\sqrt{\text{Number of Documents}} \geq 10 \Leftrightarrow \text{Number of Documents} \geq 100$).

Hence, the following equation can be used to estimate the size of vocabulary for 20 Newsgroups data. We can also examine the prediction line plot in Figure 4.16.

$$\text{Size of Vocabulary} = -432.322 + 1003.229 \times \sqrt{\text{Number of Documents}} - 1519.085 \times \sqrt[3]{\text{Number of Documents}} \tag{4.3}$$

From Equation 4.3, we can estimate that for a sample from data set of size greater than

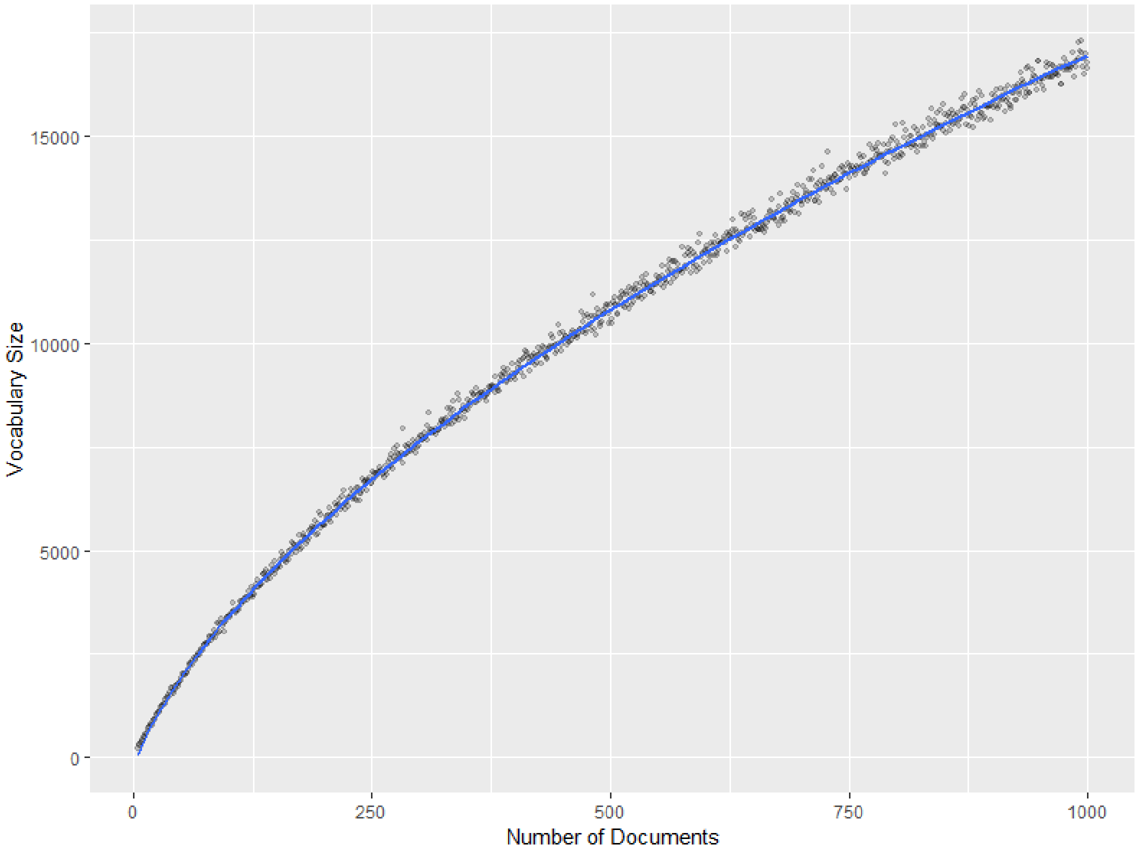

Figure 4.16: Regression Line with 20 Newsgroups Data Set

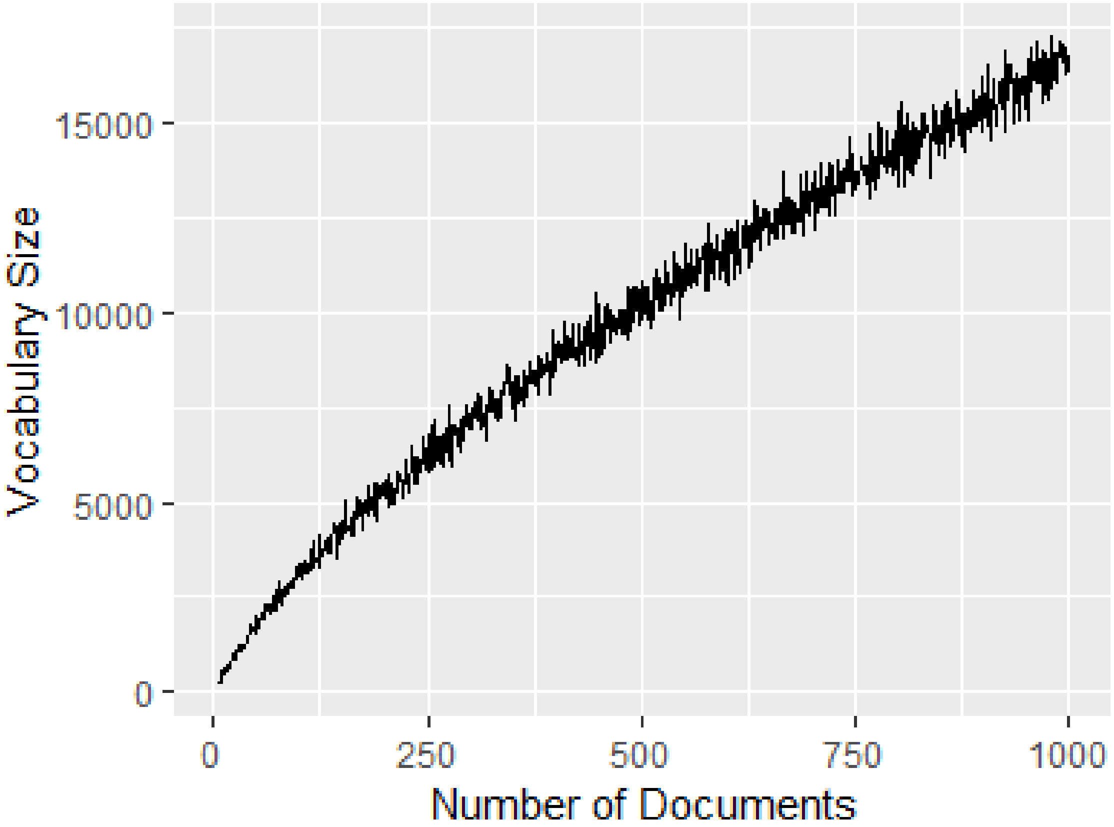

Figure 4.17: Plot of Number of documents vs Vocabulary size for sample size 100, from 5 to 1000, data set Associated Press

100. The estimation is 332.0399. Since the number is smaller than the data from Table 4.3, which is 412, we will use the larger number.

4.1.3.1.3 AssociatedPress

The sampling result is illustrated in Figure 4.17. We observed that a linear model of a square root and a cubic root of the number of documents generated the best fit. The regression model is:

$$\text{Size of Vocabulary} = \beta_0 + \beta_1 \times \sqrt{\text{Number of Documents}} + \beta_2 \sqrt[3]{\text{Number of Documents}} \quad (4.4)$$

The linear regression from an R basic package produced the following result:

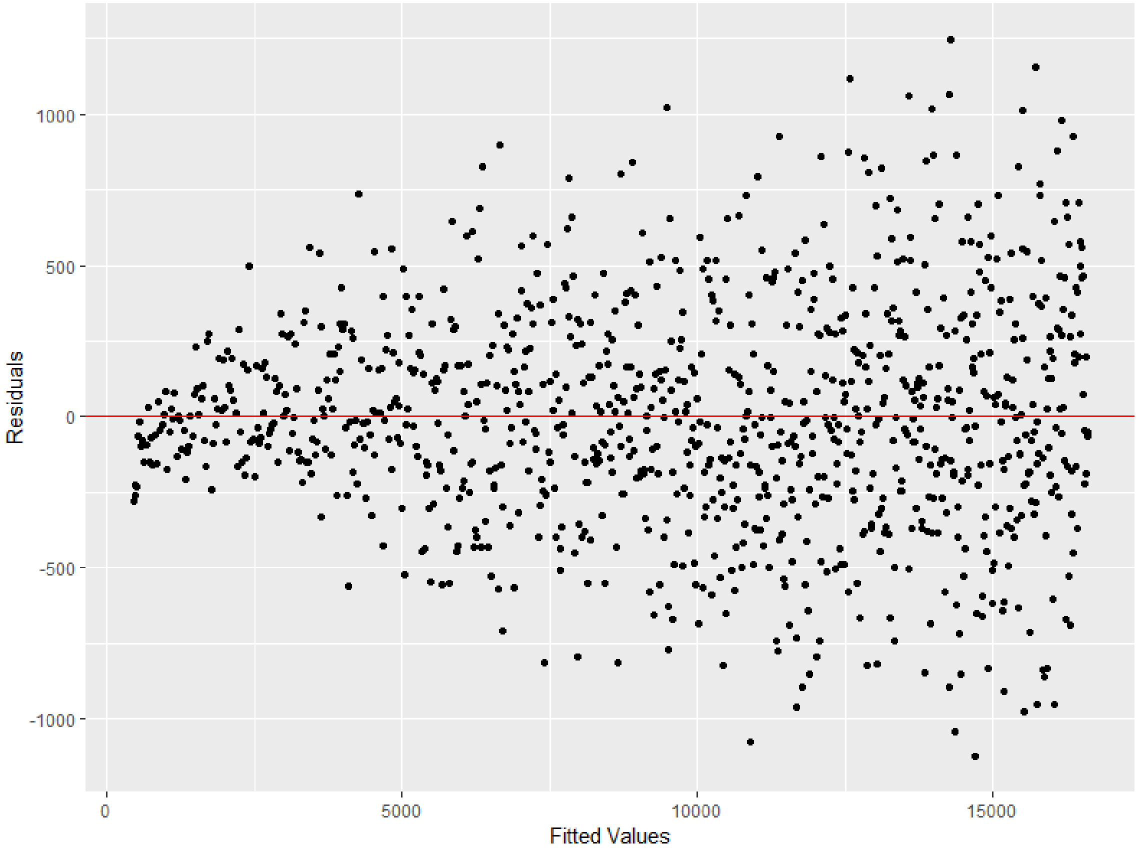

Figure 4.18: Residual Plot for 20 Associated Press data

```
Coefficients:

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept)  1675.12     160.91   10.41   <2e-16 ***

n_sqrt       1188.00      19.32   61.50   <2e-16 ***

n_cubic     -2262.96      74.98  -30.18   <2e-16 ***
```

The residual plot Figure 4.18 shows that there is possibly a heteroscedasticity problem. But the regression line in Figure 4.19 shows a good fit when the number of documents is relatively large ($\sqrt{\text{Number of Documents}} \geq 10 \Leftrightarrow \text{Number of Documents} \geq 100$).

Hence, the following equation can be used to estimate the size of vocabulary for 20 News-

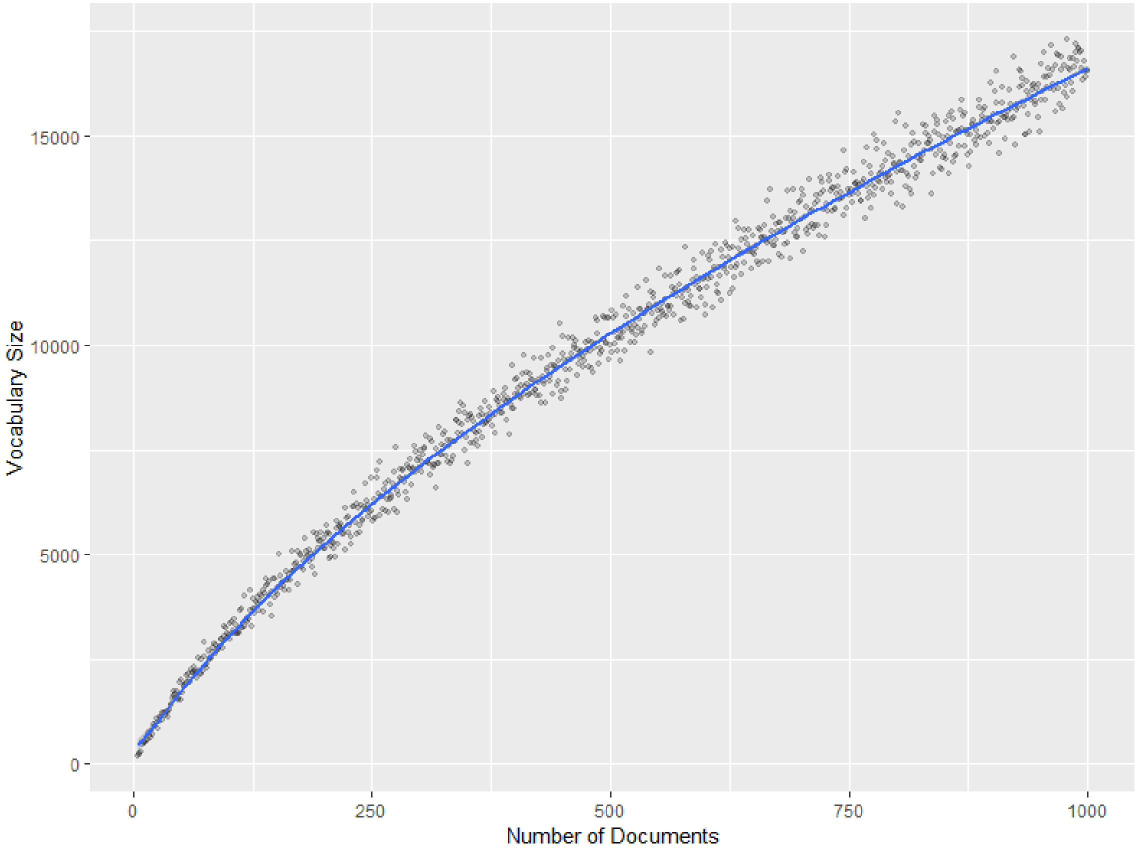

Figure 4.19: Regression Line with Associated Press Data Set

groups data. We can also examine the prediction line plot in Figure 4.19.

$$\text{Size of Vocabulary} = 1675.12 + 1188 \times \sqrt{\text{Number of Documents}} - 2262.96 \times \sqrt[3]{\text{Number of Documents}} \tag{4.5}$$

From Equation 4.5, we can estimate that for a sample from data set of size greater than 100. The estimation is 556.5027 when the number of documents equals to 10 and approximately the same with the data from Table 4.3, which is 541.

4.1.3.1.4 aclIMDB

The sampling result is illustrated in Figure 4.20. We observed that a linear model of a square root of the number of documents, a cubic root of the number of documents, and the number

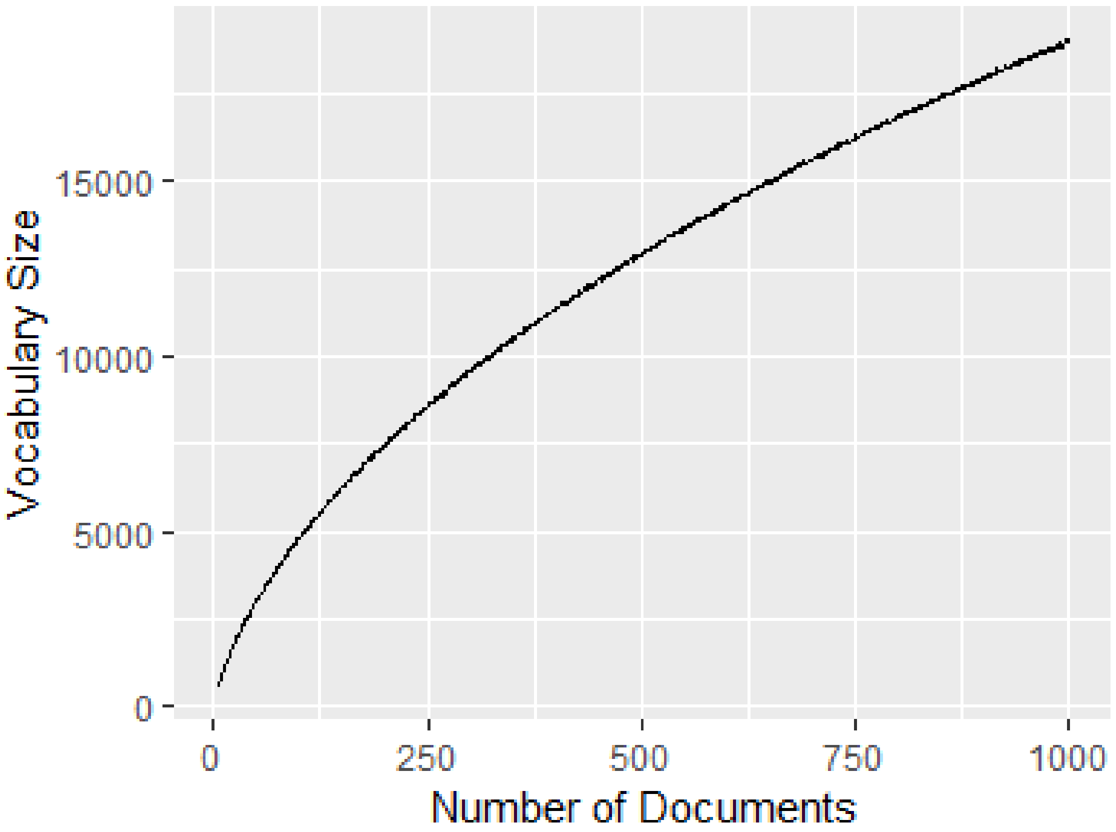

Figure 4.20: Plot of Number of documents vs Vocabulary size for sample size 100, from 5 to 1000, data set aclIMDB

of documents generated the best fit. The regression model is:

$$\text{Size of Vocabulary} = \beta_0+\beta_1\times\sqrt{\text{Number of Documents}}+\beta_2\sqrt[3]{\text{Number of Documents}}+\beta_3\text{Number of Documents}$$

The linear regression from an R basic package produced the following result:

```
Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept)  6.309e+02  3.252e+01   19.40   <2e-16 ***

n_sqrt       1.193e+03  9.344e+00  127.66   <2e-16 ***

n_cubic     -1.598e+03  2.573e+01  -62.12   <2e-16 ***

n           -3.361e+00  7.283e-02  -46.15   <2e-16 ***
```

The residual plot Figure 4.21 shows that there are some problems when the number of

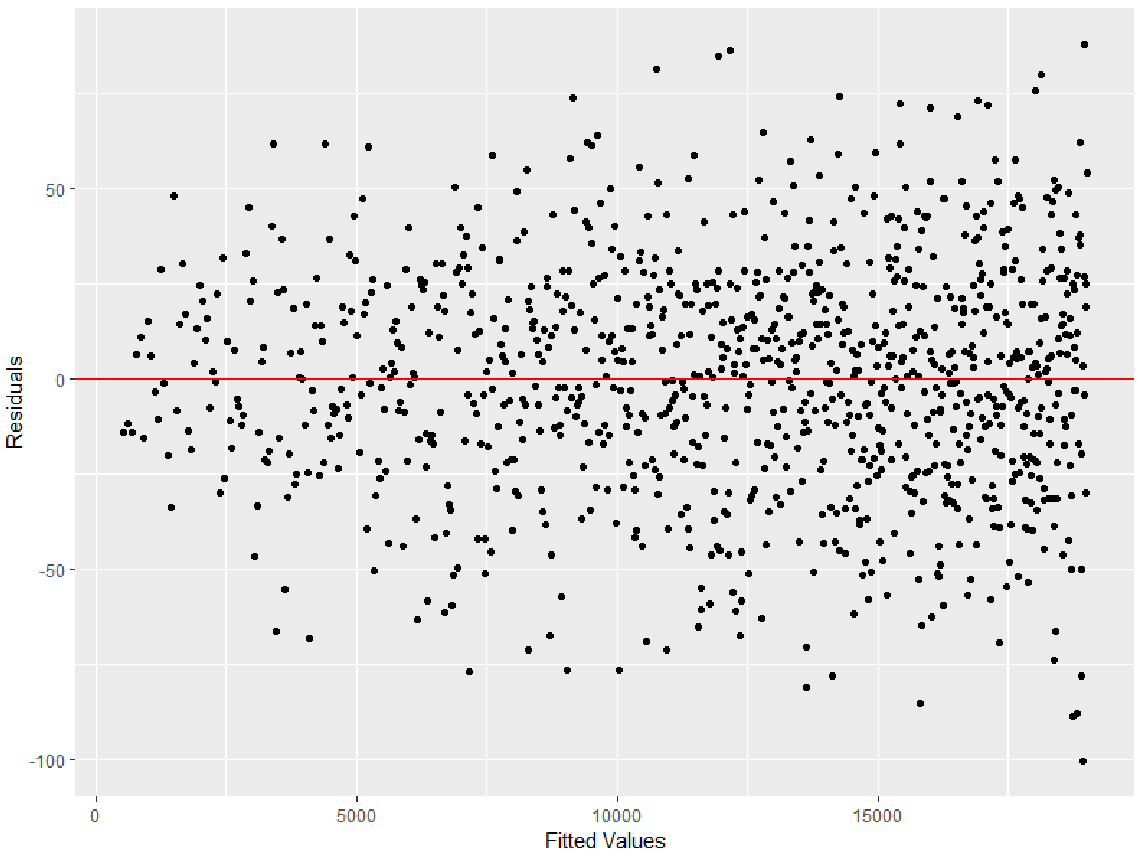

Figure 4.21: Residual Plot for 20 aclIMDB data

documents is small. But the regression line in Figure 4.22 shows a good fit when the number of documents is relatively large ($\sqrt{\text{Number of Documents}} \geq 10 \Leftrightarrow \text{Number of Documents} \geq 100$).

Hence, the following equation can be used to estimate the size of vocabulary for 20 Newsgroups data. We can also examine the prediction line plot in Figure 4.22.

$$\text{Size of Vocabulary} = 630.9 + 1193 \times \sqrt{\text{Number of Documents}} - 1598 \times \sqrt[3]{\text{Number of Documents}} - 3.361 \times \text{Number of Documents} \quad (4.6)$$

From Equation 4.6, we can estimate that for a sample from data set of size greater than 100. The estimation is 925.8767 when the number of documents equals to 10 and approximately the same with the data from Table 4.3, which is 910.

From the analysis of the real corpora, we find that the size of vocabulary ranged from 300

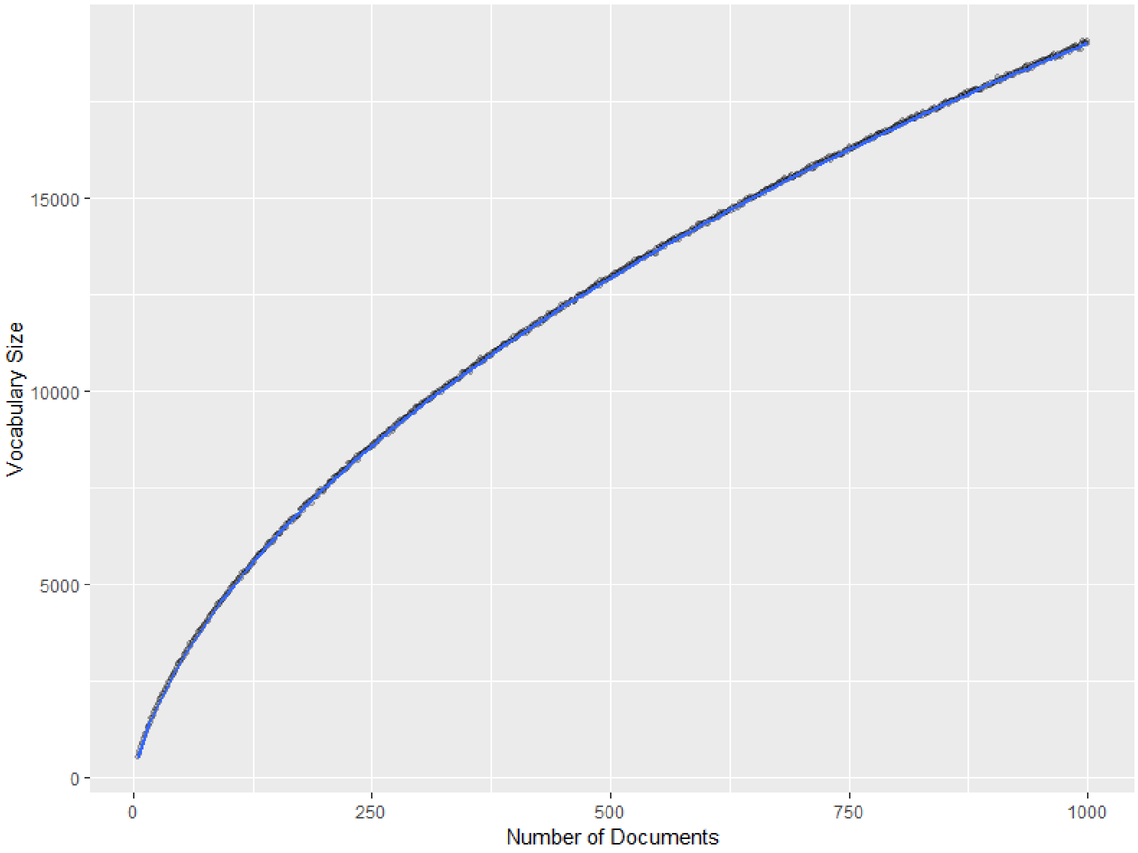

Figure 4.22: Regression Line with aclIMDB Data Set

to 925 when there are 10 documents in the corpus.

4.1.3.2 Analysis of Simulated Corpus

Similar with the analysis of document length ratio, we constructed corpora such that the vocabulary size is from 5 to 1000. The upper bound 1000 is selected based on the observation of the real corpora. For these constructed corpora, we choose to construct 10 documents and 2 topics with the document-topic distribution Θ as following:

$$\Theta = \begin{array}{rc} & \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{array}{r} Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} & \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array} \right] \end{array}$$

For each level of size of vocabulary, the document length ratios were selected to be 10, which means that the number of words in each document is 10 times of the size of vocabulary for all the documents we created. In the real corpora, the document length will not increase along with the vocabulary size. But we are trying to exclude the effect of document length ratio as much as possible. Two different priors of topic-word distributions are selected: the Dirichlet distribution and the multinormal distribution.

From Figure 4.23 and Figure 4.24, we find that the correlations are relatively stable when the size of vocabulary is greater than 100.

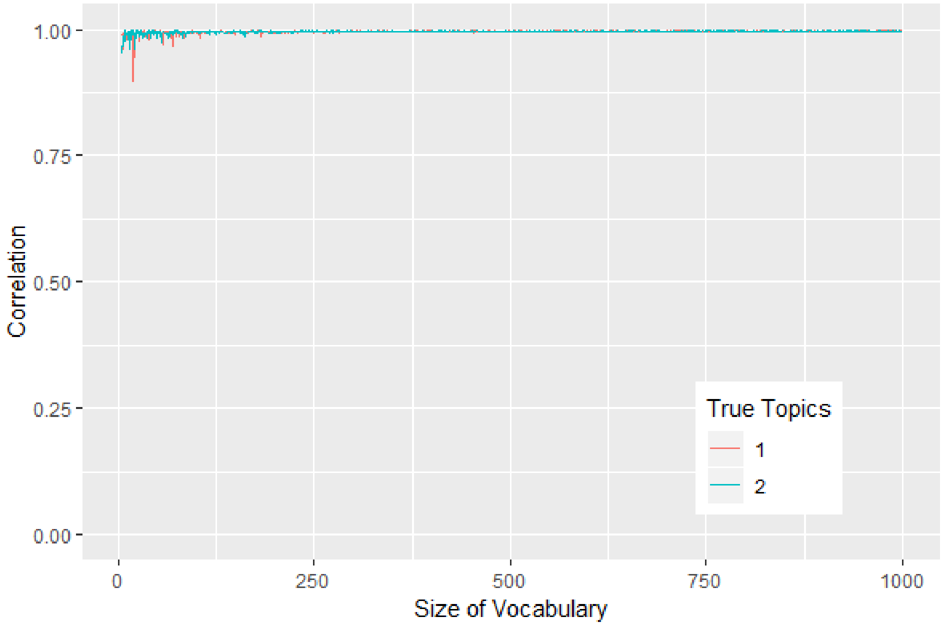

Figure 4.23: Simulated Analysis of Size of Vocabulary for Dirichlet prior

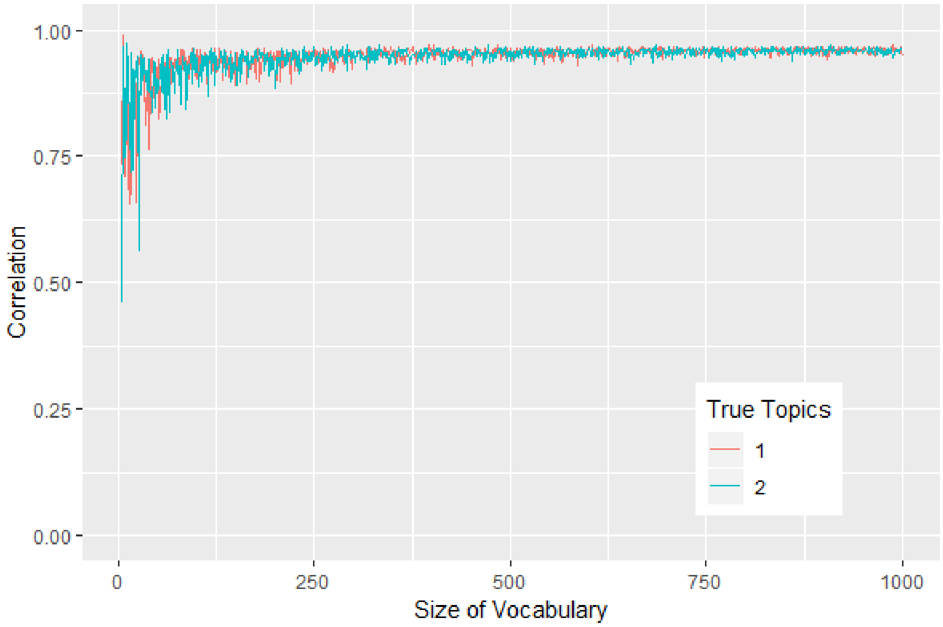

Figure 4.24: Simulated Analysis of Size of Vocabulary for Multi-normal prior

4.2 Document Structure

4.2.1 Factor Levels

Based on the analysis of the real corpus and simulated corpus, it is reasonable to select the factor levels of type of topic-word distribution, document length ratio, and size of vocabulary as following:

1. Type of Topic-word distributions: the Dirichlet prior and multi-normal prior.

2. Document Length ratio: 19 levels as following:

$$[\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$$

3. Size of vocabulary: From 5 to 100.

The only factor that does not depends on the overall topic structure is the document-topic distribution. This factor is not observable and hence can not be observed in the real corpora. For simplicity purposes, we selected the number of topics to be 2 as before. The number of topics can be increased if needed, but that is not contained in this study. In the previous sections, we selected the document-topic distribution Θ such that each document is solely constructed based on one of the two topics. However, as stated in Chapter 2, this is usually not true in real data sets. We would like to analyze the effect of the mixture of topics through changing the proportion that each topic has in documents.

Different mixtures result in different document-topic distributions, and can be represented through the document-topic matrices. The mixtures in this study can be categorized into three groups.

1. Extreme cases

2. Mixture cases

3. Analytical Mixture cases

The extreme cases represent “complete” separation of topics between documents and “no” separation of topics between the documents. Mixture cases are used to observe the quality of the LDA model under different levels of topic mixtures. The analytical mixture cases are used to answer even specific questions about mixtures that will be discussed later. We denote the document-topic matrix by Θ_i where i is the index of the case.

For each case in the following discussion, we will examine the quality of the LDA model results based on correlation. Based on the observations from the previous section we set the number of documents to be 10, vary the document length ratio from 0.1 to 10, and vary the size of vocabulary from 5 to 100. Both the Dirichlet and the Multi-normal priors will be considered.

4.2.2 Extreme Cases

There are two extreme cases:

1. Case 1: Θ_1 Five documents are constructed of one topic only, and the other five documents are constructed of another topic.

2. Case 2: Θ_2 Each document is constructed evenly of the two topics.

Θ_1 is the case we used in some of the previous analyses:

$$\Theta_1 = \begin{array}{r|cc|} & TrueTopic1 & TrueTopic2 \\ Document1 & 1 & 0 \\ Document2 & 0 & 1 \\ Document3 & 1 & 0 \\ Document4 & 0 & 1 \\ Document5 & 1 & 0 \\ Document6 & 0 & 1 \\ Document7 & 1 & 0 \\ Document8 & 0 & 1 \\ Document9 & 1 & 0 \\ Document10 & 0 & 1 \end{array}$$

Figure 4.25 is the Case 1 simulation result with varying document length ratios. Figure 4.26 is the simulation result for Case 1 with varying size of vocabulary.

Case 1 is an ideal case in that the data follow the assumptions of the LDA model. The results may be referred as the benchmark of a “good” fit. The left hand side of Figure 4.26 indicates

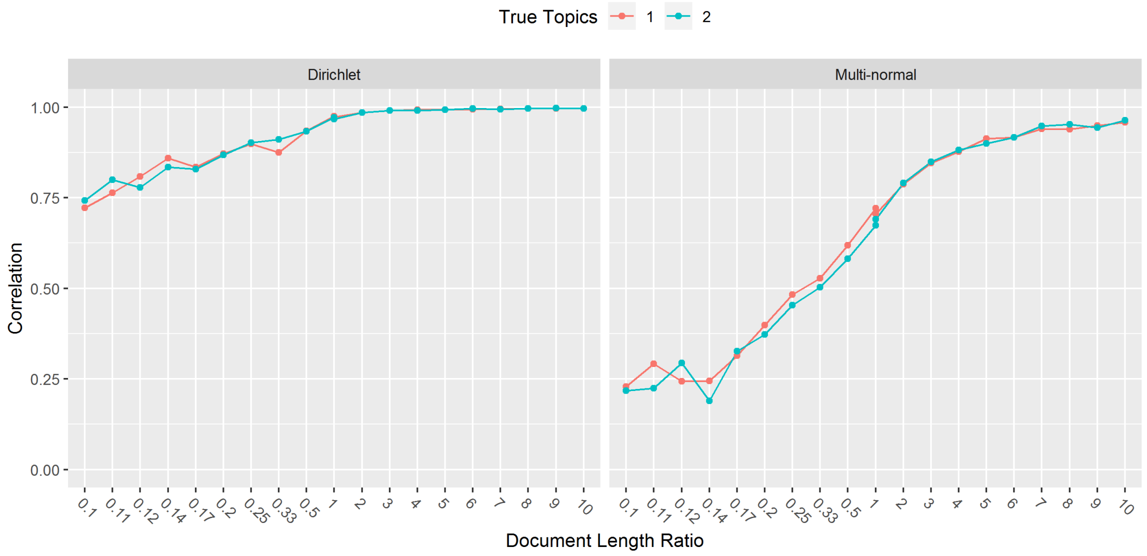

Figure 4.25: Correlation between true and fitted topics with varying document length ratio for Case 1

that the fitted topics are highly correlated with the true topics using the Dirichlet prior, especially when the document length ratio exceeds 1. The right hand side of Figure 4.26 shows that the fitted topics are highly correlated with the true topics using the multi-normal prior only when the document length ratio exceeds 4. Comparing the two sides of the figure indicates that the choice of the prior distribution is important even under the most ideal scenario. We note that the type of the prior distribution often results in a larger impact on the correlation than the document length ratio. In fact, when the document length ratio is located within the interval $[0.13, 0.25]$, which was range of the mean document length ratios of the real corpora we analyzed in the previous section, the multi-normal prior scenario results in correlations lower than 0.5. By comparing Figures 4.25 and 4.26, we also found that the topic distribution within the documents impacted the quality of the LDA model results based on correlation, and the multi-normal prior distribution is not preferred when the document length ratio is relatively small.

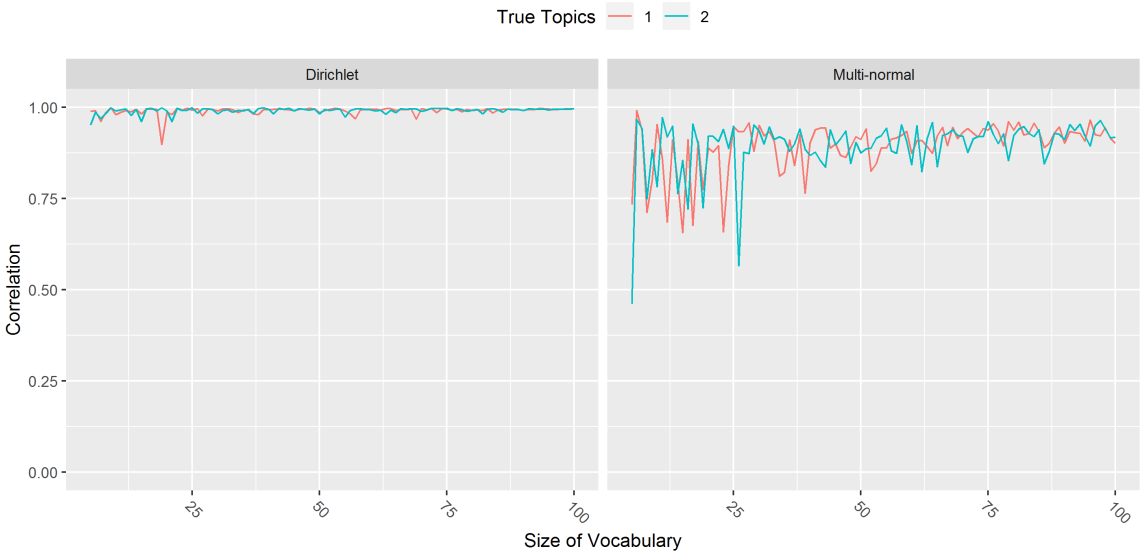

Figure 4.26: Correlation between true and fitted topics with varying size of vocabulary for Case 1

Θ_2 can be represented as:

$$\Theta_2 = \begin{array}{c c} & \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{array}{c} Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} & \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$$

Figure 4.27 is the Case 2 simulation result with varying document length ratios. Figure 4.28 is the simulation result for Case 2 with varying size of vocabulary.

Case 2 is the "worst case" scenario since topics are evenly mixed in each document. There is no surprise in the poor results shown in Figures 4.27 and 4.28. These results may be used

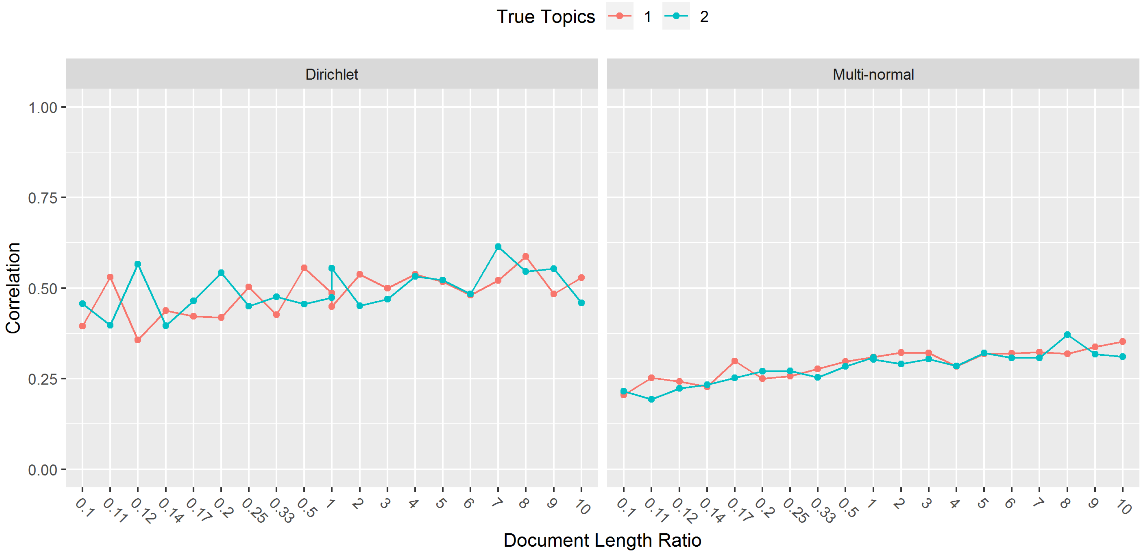

Figure 4.27: Correlation between true and fitted topics with varying document length ratio for Case 2

as benchmark values of a "bad" fit. In both figures, the Dirichlet prior shows better correlation than the Multi-normal prior. One interesting observation is that in Figure 4.28, the correlation is decreasing as the size of vocabulary increasing. Also in Figure 4.27, we notice that the correlation is increasing very slowly as the document length ratio increase compared with the Case 1 Figure 4.25. This observation indicates that when topics are evenly distributed within documents, increasing the document length ratio helps very little in LDA quality based on correlation.

4.2.3 Mixture Cases

There are two cases in which the documents are "simple" mixtures of the two topics:

3. Case 3: Θ_3 Each document is constructed as 25% from one topic and 75% from the other topic.

4. Case 4: Θ_4 Within the corpus, four documents are constructed from one topic only, four documents are 25% from one topic and 75% from the other topic, and the remaining two documents are constructed 50% from one topic and 50% from the other topic.

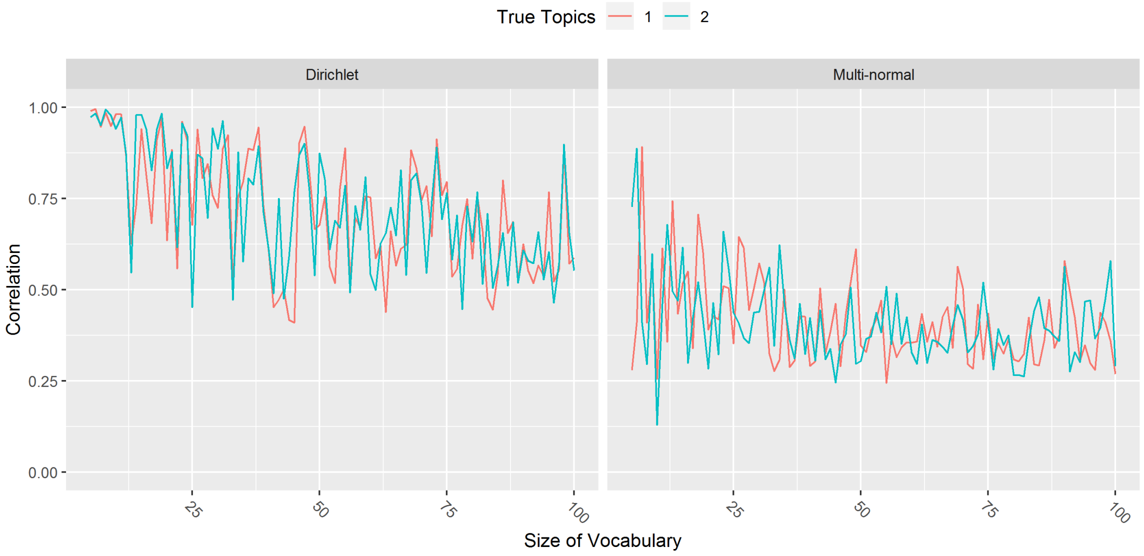

Figure 4.28: Correlation between true and fitted topics with varying size of vocabulary for Case 2

Θ_3 can be represented as:

$$\Theta_3 = \begin{array}{c} \\ Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} \begin{array}{c} \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \\ 0.75 & 0.25 \\ 0.25 & 0.75 \\ 0.75 & 0.25 \\ 0.25 & 0.75 \\ 0.75 & 0.25 \\ 0.25 & 0.75 \\ 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix} \end{array}$$

Figure 4.29 is the Case 3 simulation result with varying document length ratios. Figure 4.30 is the simulation result for Case 3 with varying size of vocabulary.

Case 3 is a combination of the two previous cases. By comparing Case 3 with Case 1 and Case 2 we can get some sense of the impact of mixtures of topics on the LDA results. Figure 4.29

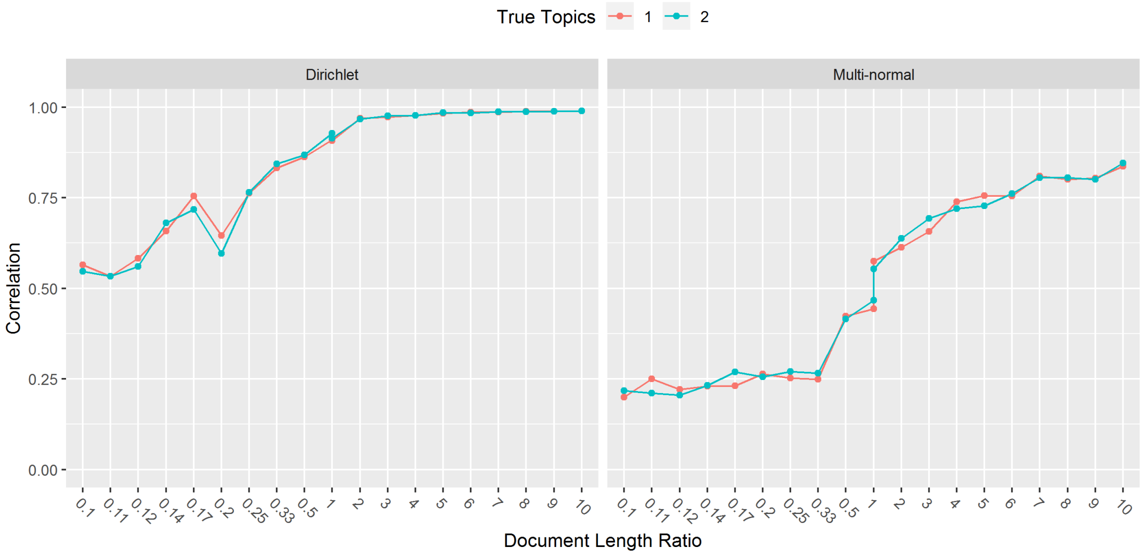

Figure 4.29: Correlation between true and fitted topics with varying document length ratio for Case 3

indicates that the multi-normal prior distribution is more affected by the mixture of topics in the documents than the Dirichlet prior distribution. Specifically, the correlation on Dirichlet-prior-side fo the figure drops about 20 percent compared with Figure 4.25 while the multi-normal-prior-side of Figure 4.29 shows that the correlation is almost constant (around 0.25) when the document length ratio is less than 0.5 and is very close to the benchmark of "bad" results in figure 4.27. This again shows the multi-normal is not the preferable prior in LDA. The same observation happens in Figure 4.30. The left hand side of Figure 4.30 is almost identical with the left hand side of Figure 4.26, but the right hand side of Figure 4.30 tends to converge to 0.85 instead of 1. Although Case 3 is designed as a mixture somewhere near the middle of the two extreme cases, the results reasonably consistent with Case 1. This shows that the impact of the document structure is not a simple "linear". We will discuss this later in the "Analytical mixture cases".

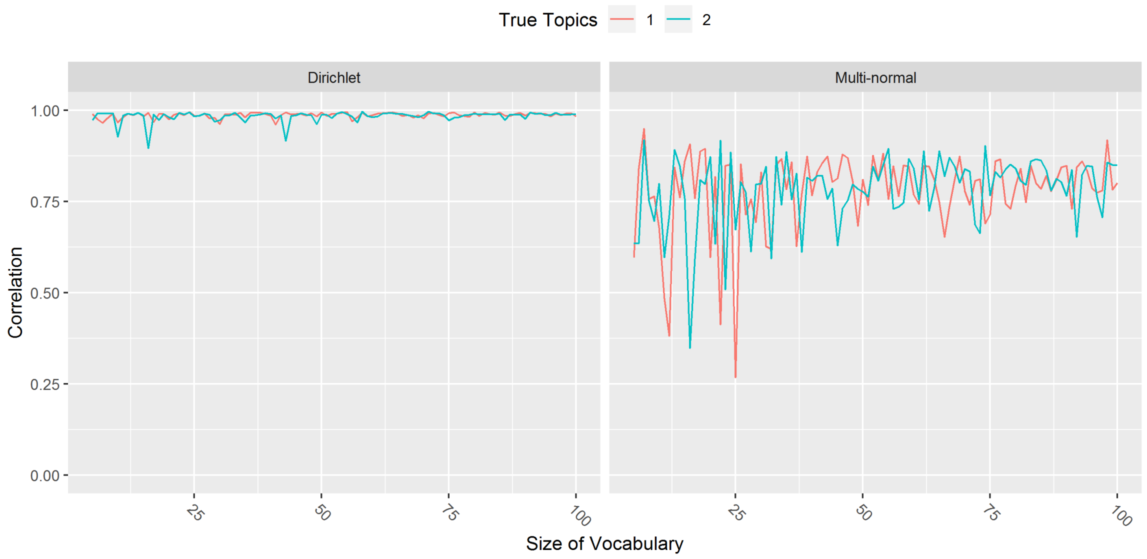

Figure 4.30: Correlation between true and fitted topics with varying size of vocabulary for Case 3

Θ_4 can be represented as:

$$\Theta_4 = \begin{array}{r|cc|} & TrueTopic1 & TrueTopic2 \\ Document1 & 0.5 & 0.5 \\ Document2 & 0.5 & 0.5 \\ Document3 & 1 & 0 \\ Document4 & 0 & 1 \\ Document5 & 1 & 0 \\ Document6 & 0 & 1 \\ Document7 & 0.75 & 0.25 \\ Document8 & 0.25 & 0.75 \\ Document9 & 0.75 & 0.25 \\ Document10 & 0.25 & 0.75 \end{array}$$

Figure 4.31 is the Case 4 simulation result with varying document length ratios. Figure 4.32 is the simulation result for Case 4 with varying size of vocabulary.

Case 4 is a mixture of Case 1, Case 2, and Case 3. Case 4 can also be considered the first attempt to simulate a collection of real documents. In general, comparing with the previous three

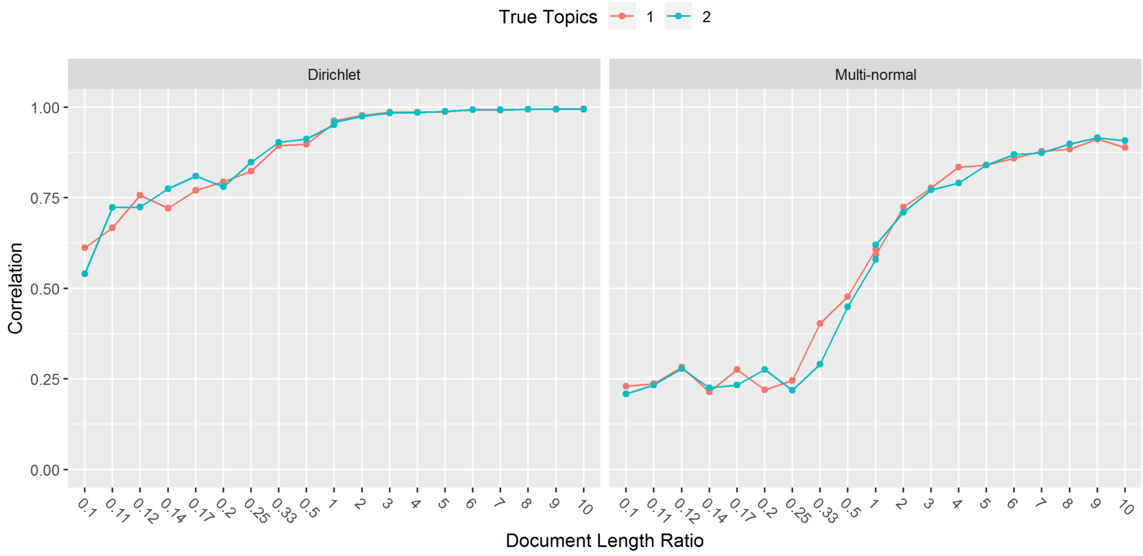

Figure 4.31: Correlation between true and fitted topics with varying document length ratio for Case 4

cases, Case 4 results are most similar to Case 3.

4.2.4 Analytical Mixture Cases

There were eight analytical mixture cases considered to further understand the impact of mixtures. There are two methods to modify the "mixture" structure of the documents: change the proportion of topics in a single document, or change the proportion of extreme cases' documents in a corpus. There are categories of cases constructed based on the two methods and each category is constructed to study one specific question.

The first category of cases is constructed through changing the proportion of topics in a single document. The specific question for this category is: what is the impact of close to evenly distributed topics within documents impact the fitted results? It consists of three cases:

5. Case 5: Θ_5 Five documents are constructed as 42.5% from one topic and 57.5% from the other topic and the five other documents are constructed as 57.5% from one topic and 42.5% from the other topic.

6. Case 6: Θ_6 Five documents are constructed as 45% from one topic and 55% from the other topic and the five other documents are constructed as 55% from one topic and 45% from the other topic.

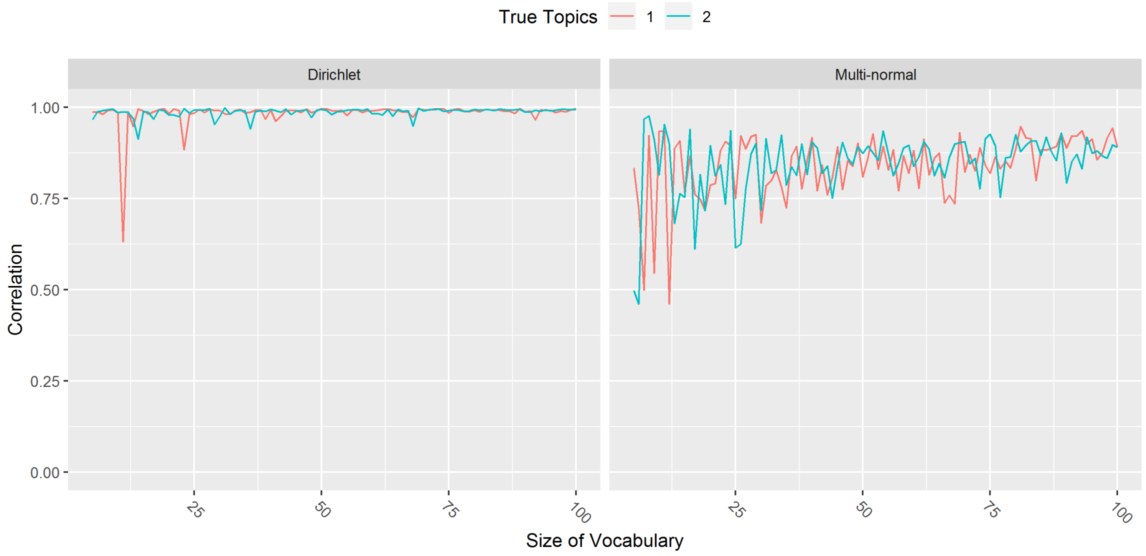

Figure 4.32: Correlation between true and fitted topics with varying size of vocabulary for Case 4

7. Case 7: Θ_7 Five documents are constructed as 47.5% from one topic and 52.5% from the other topic and the five other documents are constructed as 52.5% from one topic and 47.5% from the other topic.

   Θ_5 can be represented as:

$$\Theta_5 = \begin{array}{r} \\ Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} \begin{array}{c} \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{bmatrix} 0.425 & 0.575 \\ 0.575 & 0.425 \\ 0.425 & 0.575 \\ 0.575 & 0.425 \\ 0.425 & 0.575 \\ 0.575 & 0.425 \\ 0.425 & 0.575 \\ 0.575 & 0.425 \\ 0.425 & 0.575 \\ 0.575 & 0.425 \end{bmatrix} \end{array}$$

Θ_6 can be represented as:

$$\Theta_6 = \begin{array}{c} \\ Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} \begin{array}{c} \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{bmatrix} 0.45 & 0.55 \\ 0.55 & 0.45 \\ 0.45 & 0.55 \\ 0.55 & 0.45 \\ 0.45 & 0.55 \\ 0.55 & 0.45 \\ 0.45 & 0.55 \\ 0.55 & 0.45 \\ 0.45 & 0.55 \\ 0.55 & 0.45 \end{bmatrix} \end{array}$$

Θ_7 can be represented as:

$$\Theta_7 = \begin{array}{c} \\ Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} \begin{array}{c} \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{bmatrix} 0.475 & 0.525 \\ 0.525 & 0.475 \\ 0.475 & 0.525 \\ 0.525 & 0.475 \\ 0.475 & 0.525 \\ 0.525 & 0.475 \\ 0.475 & 0.525 \\ 0.525 & 0.475 \\ 0.475 & 0.525 \\ 0.525 & 0.475 \end{bmatrix} \end{array}$$

Figure 4.33 is the Case 5 simulation result with varying document length ratios. Figure 4.34 is the simulation result for Case 5 with varying size of vocabulary. Figure 4.35 is the Case 6 simulation result with varying document length ratios. Figure 4.36 is the simulation result for Case 6 with varying size of vocabulary. Figure 4.37 is the Case 7 simulation result with varying document

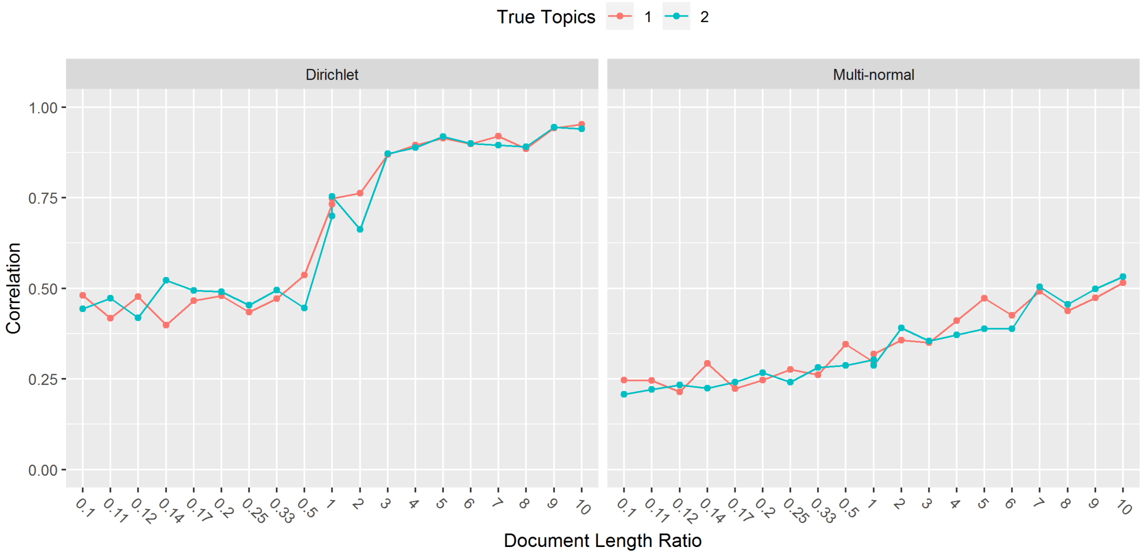

Figure 4.33: Correlation between true and fitted topics with varying document length ratio for Case 5

length ratios. Figure 4.38 is the simulation result for Case 7 with varying size of vocabulary.

Results from cases 5, 6, and 7 will be discussed together. Under the multi-normal prior situation, the Case 6 and Case 7 results are similar to the Case 2 results. Figure 4.33 shows the correlation increasing faster as the document length ratio increased compared to Figure 4.35 and Figure 4.37. The results suggest that the difference between the two topic proportions need to be more than 0.1 for the LDA model to provide fitted topics that correlate with the real topics under the multi-normal prior distribution. When using the Dirichlet prior distribution, the correlation decrease when the size of vocabulary increases for Case 7 (Figure 4.38). Both Figure 4.34 and Figure 4.36 show that the correlations are usually close to 1 under the Dirichlet prior. Hence, the difference between the proportions of the two topics needs to be more than 0.05 for the LDA model to provide fitted topics that correlate with the true topics using the Dirichlet prior.

The second category of cases is constructed through changing the numbers of documents that are extreme cases (all one topic or the other topic). The specific question for this category is: how does the number of extreme case documents impact the fitted results? It consists of five cases:

8. Case 8: Θ_8 Within the corpus, five of the documents are constructed from 100% of one topic or the other topic, the other five documents are constructed as 50% from one topic and 50% from the other topic.

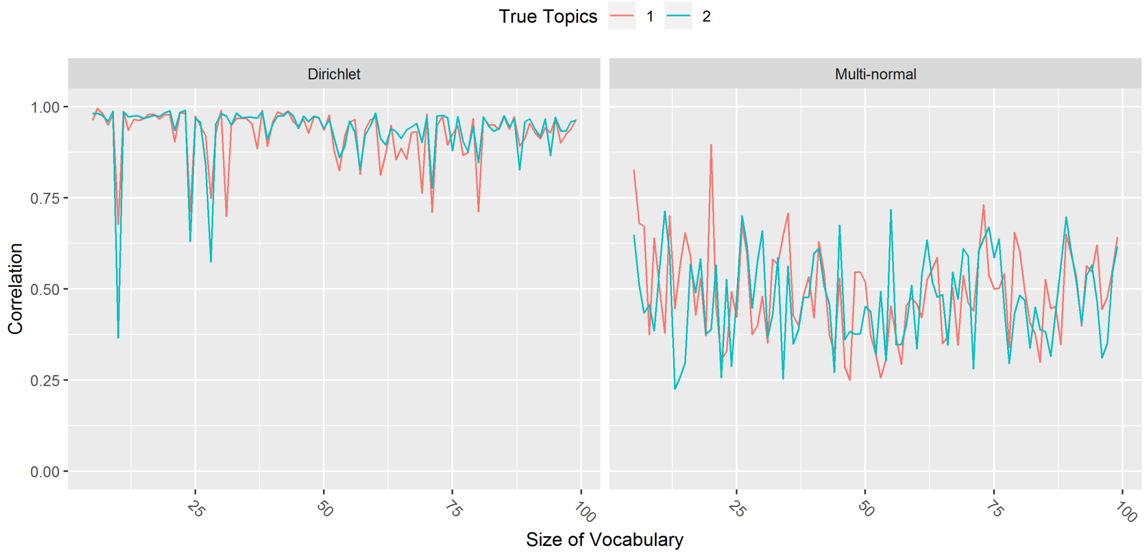

Figure 4.34: Correlation between true and fitted topics with varying size of vocabulary for Case 5

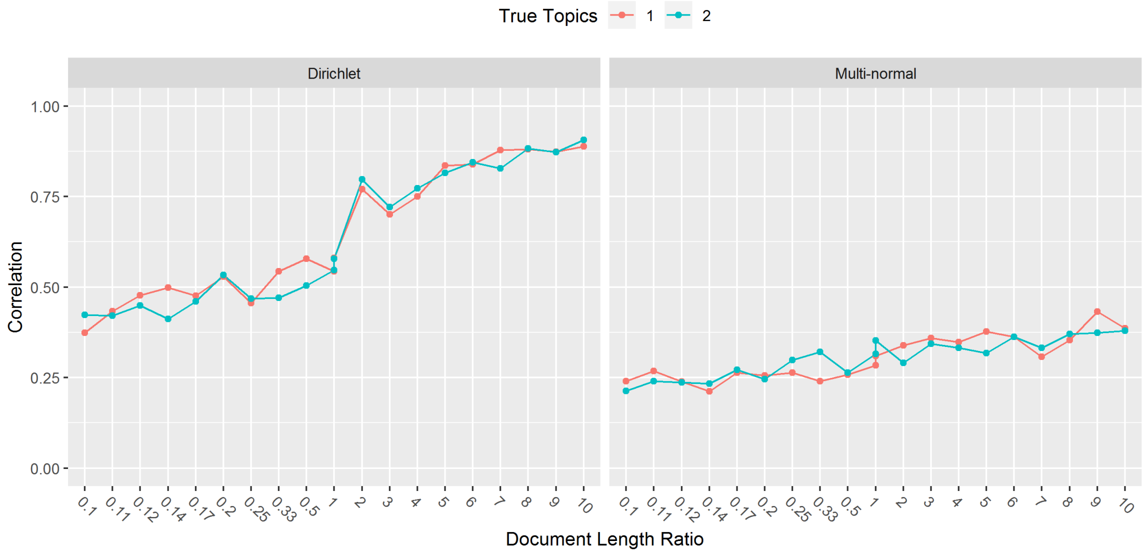

Figure 4.35: Correlation between true and fitted topics with varying document length ratio for Case 6

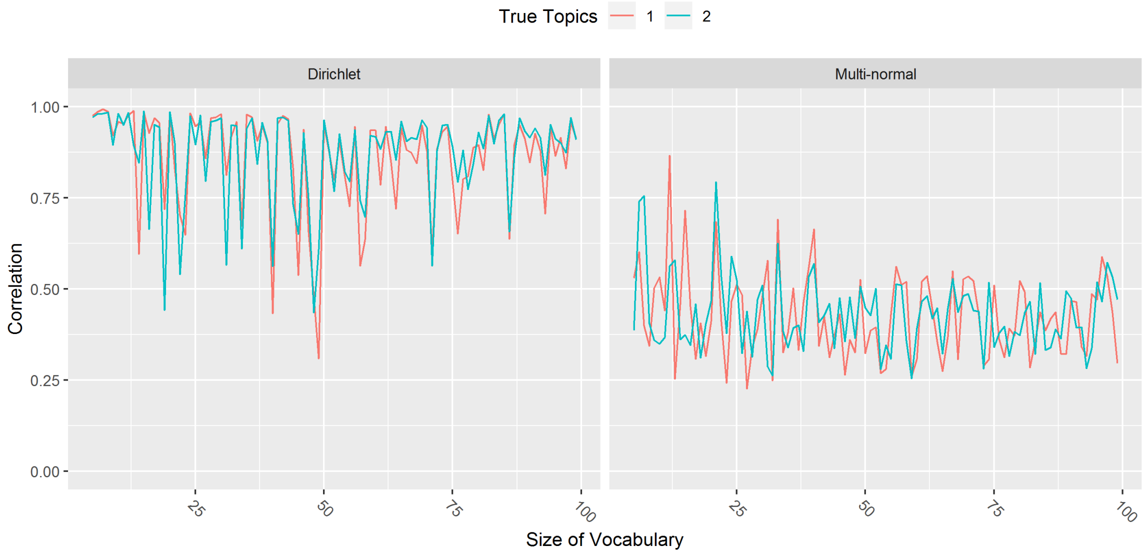

Figure 4.36: Correlation between true and fitted topics with varying size of vocabulary for Case 6

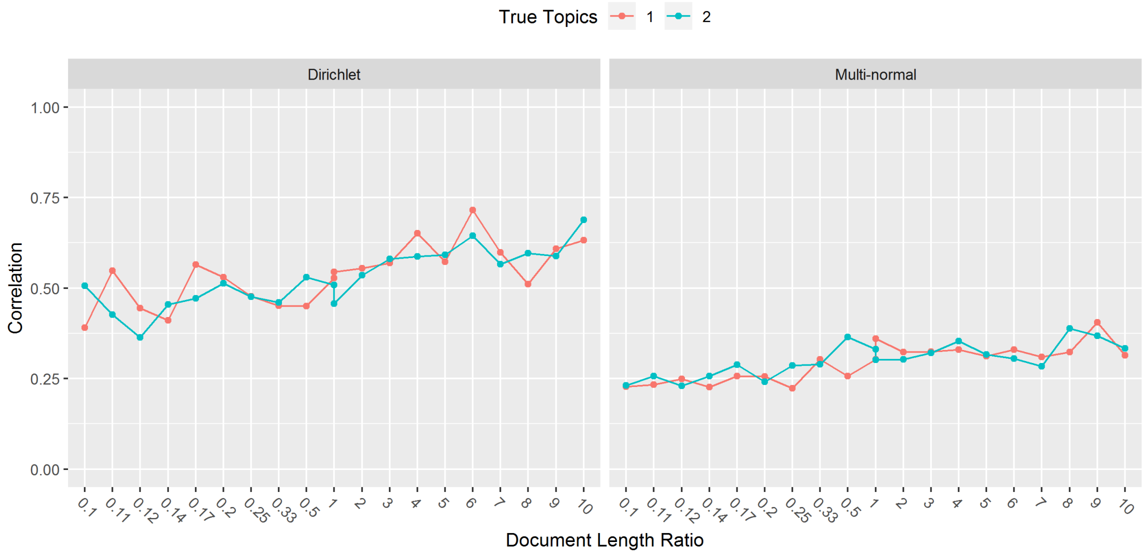

Figure 4.37: Correlation between true and fitted topics with varying document length ratio for Case 7

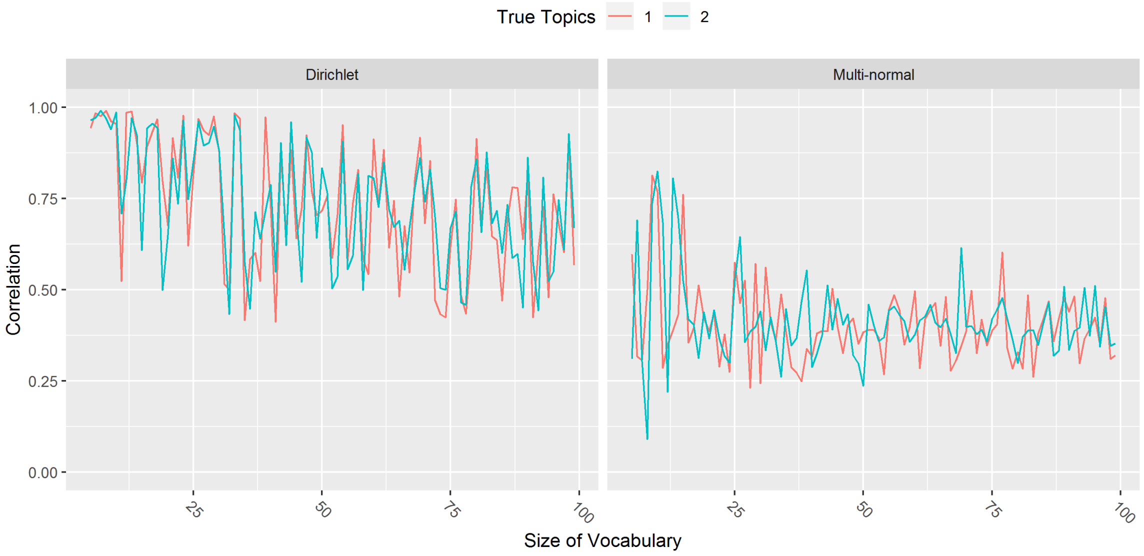

Figure 4.38: Correlation between true and fitted topics with varying size of vocabulary for Case 7

9. Case 9: Θ_9 Within the corpus, three of the documents are constructed from 100% of one topic or the other topic, the other seven documents are constructed as 50% from one topic and 50% from the other topic.

10. Case 10: Θ_{10} Within the corpus, two of the documents are constructed from 100% of one topic or the other topic, the other eight documents are constructed as 50% from one topic and 50% from the other topic.

11. Case 11: Θ_{11} Within the corpus, one of the documents are constructed from 100% of one topic or the other topic, the other nine documents are constructed as 50% from one topic and 50% from the other topic.

12. Case 12: Θ_{12} Within the corpus, one of the documents are constructed from 75% of one topic and 25% of the other topic, the other nine documents are constructed as 50% from one topic and 50% from the other topic.

Θ_8 can be represented as:

$$\Theta_8 = \begin{array}{c} \\ {\scriptstyle Document1} \\ {\scriptstyle Document2} \\ {\scriptstyle Document3} \\ {\scriptstyle Document4} \\ {\scriptstyle Document5} \\ {\scriptstyle Document6} \\ {\scriptstyle Document7} \\ {\scriptstyle Document8} \\ {\scriptstyle Document9} \\ {\scriptstyle Document10} \end{array} \begin{array}{cc} {\scriptstyle TrueTopic1} & {\scriptstyle TrueTopic2} \\ \left[\begin{array}{c} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 1.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 1.0 \end{array}\right. & \left.\begin{array}{c} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.0 \\ 1.0 \\ 0.0 \\ 1.0 \\ 0.0 \end{array}\right] \end{array}$$

Θ_9 can be represented as:

$$\Theta_9 = \begin{array}{c} \\ {\scriptstyle Document1} \\ {\scriptstyle Document2} \\ {\scriptstyle Document3} \\ {\scriptstyle Document4} \\ {\scriptstyle Document5} \\ {\scriptstyle Document6} \\ {\scriptstyle Document7} \\ {\scriptstyle Document8} \\ {\scriptstyle Document9} \\ {\scriptstyle Document10} \end{array} \begin{array}{cc} {\scriptstyle TrueTopic1} & {\scriptstyle TrueTopic2} \\ \left[\begin{array}{c} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 1.0 \\ 0.0 \\ 1.0 \end{array}\right. & \left.\begin{array}{c} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.0 \\ 1.0 \\ 0.0 \end{array}\right] \end{array}$$

Θ_{10} can be represented as:

$$\Theta_{10} = \begin{array}{c} \\ {\scriptstyle Document1} \\ {\scriptstyle Document2} \\ {\scriptstyle Document3} \\ {\scriptstyle Document4} \\ {\scriptstyle Document5} \\ {\scriptstyle Document6} \\ {\scriptstyle Document7} \\ {\scriptstyle Document8} \\ {\scriptstyle Document9} \\ {\scriptstyle Document10} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle TrueTopic1} & {\scriptstyle TrueTopic2} \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \end{array}$$

Θ_{11} can be represented as:

$$\Theta_{11} = \begin{array}{c} \\ {\scriptstyle Document1} \\ {\scriptstyle Document2} \\ {\scriptstyle Document3} \\ {\scriptstyle Document4} \\ {\scriptstyle Document5} \\ {\scriptstyle Document6} \\ {\scriptstyle Document7} \\ {\scriptstyle Document8} \\ {\scriptstyle Document9} \\ {\scriptstyle Document10} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle TrueTopic1} & {\scriptstyle TrueTopic2} \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1.0 & 0.0 \end{bmatrix} \end{array}$$

Θ_{12} can be represented as:

$$\Theta_{12} = \begin{array}{c} \\ Document1 \\ Document2 \\ Document3 \\ Document4 \\ Document5 \\ Document6 \\ Document7 \\ Document8 \\ Document9 \\ Document10 \end{array} \begin{array}{c} \begin{array}{cc} TrueTopic1 & TrueTopic2 \end{array} \\ \begin{bmatrix} 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.50 & 0.50 \\ 0.75 & 0.25 \end{bmatrix} \end{array}$$

Figure 4.39 is the Case 8 simulation result with varying document length ratios. Figure 4.40 is the simulation result for Case 8 with varying size of vocabulary. Figure 4.41 is the Case 9 simulation result with varying document length ratios. Figure 4.42 is the simulation result for Case 9 with varying size of vocabulary. Figure 4.43 is the Case 10 simulation result with varying document length ratios. Figure 4.44 is the simulation result for Case 10 with varying size of vocabulary. Figure 4.45 is the Case 11 simulation result with varying document length ratios. Figure 4.46 is the simulation result for Case 11 with varying size of vocabulary. Figure 4.47 is the Case 12 simulation result with varying document length ratios. Figure 4.48 is the simulation result for Case 12 with varying size of vocabulary.

Case 8 to Case 11 are mixtures of earlier cases. For Case 8, the quality of the LDA results are not impacted as much as might be expected based on the earlier 50% mixture results, especially when comparing Figure 4.40 and 4.26. The left hand side of Figure 4.40 has some low correlation values when the size of vocabulary is less than 25, but the correlation is almost always close to 1 after that. For Figure 4.39, the left hand side of the graph shows the correlation higher than 0.75 when the document length ratio is greater than 0.14. This correlation value is very good compared with the "good" benchmark in Figure 4.25. However, the right hand side shows the correlation is around 0.25 when the document length ratio is less than 0.25. This is a low correlation value compared with the “bad” benchmark in Figure 4.27. The above observation indicates that the LDA model is sensitive to the prior distribution and the impact of the document length ratio appears greater than the impact of the size of the vocabulary.

Also, we noted earlier that the difference between the left hand side of Figure 4.26, Figure 4.28, and Figure 4.40 are not “linear” indicating that the impact of the document structure is not linear. To fully study the impact of the document structures, we constructed Case 9, 10, and 11. Cases 9, 10, and 11 contains 30%, 20%, and 10% documents that solely depend on the one or the other topic, respectively. For the analysis of how size of vocabulary impacts the correlation, in Figure 4.42, 4.44, and 4.46, the left hand sides (Dirichlet prior) still have correlations close to 1. However, the right hand sides (Multi-normal prior) show clearly a lower correlation level. For the analysis of the document length ratio, in Figure 4.41, 4.43, and 4.45, the right hand sides show correlation higher than 0.75 when the document length ratio is higher than 0.5. But when the document length ratio is lower than 0.5, Figure 4.45 show the correlation close to 0.5, which near the “bad” benchmark in Figure 4.27. We observe that as the proportion of Case 1 documents decreases, the quality of LDA results decrease. Moreover, the rate of the decrease is negatively correlated with the document length ratio.

Since our corpus only contains 10 documents, Case 11 is closest to Case 2. Since we can not lower the proportion of Case 1 documents any further, we need to change the proportion of topics in the documents so that we can generate a simulation that is closer to Case 2. In Figures 4.47 and 4.48, we find that the right hand sides (Multi-normal prior) is very close to the “bad” benchmark in Figure 4.27 and 4.28, but the left hand sides are still showing relatively good correlations.

In this chapter, we performed simulations that possessed different document and topic structures. The characteristics discussed in Chapter 3 are further explained and treated as factors in the simulations. We also used some real world data sets to help us determine some of the reasonable factor levels. We finally observed the results of the simulations of document structures and discussed how they are related.

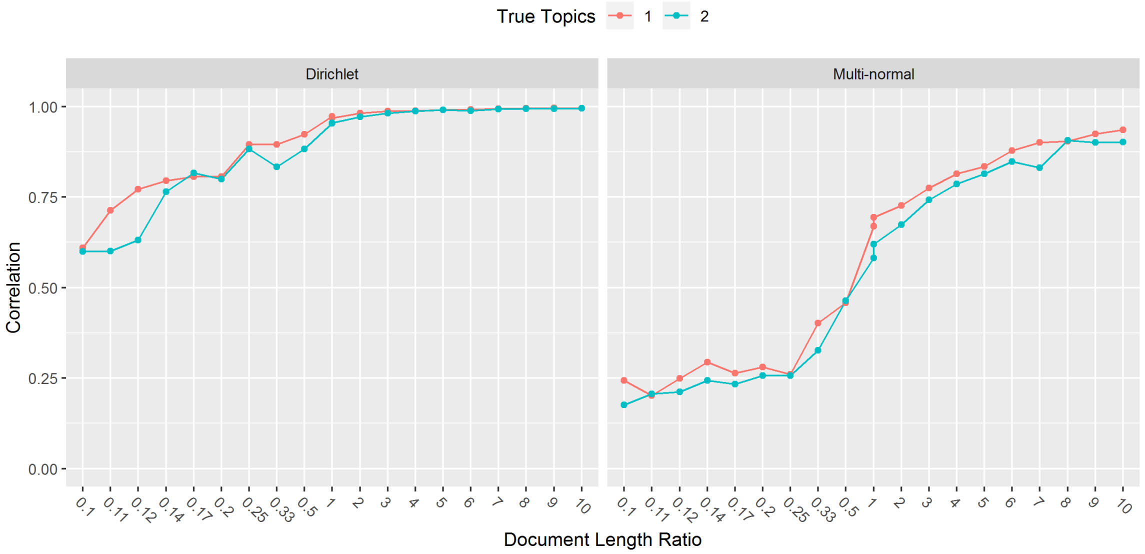

Figure 4.39: Correlation between true and fitted topics with varying document length ratio for Case 8

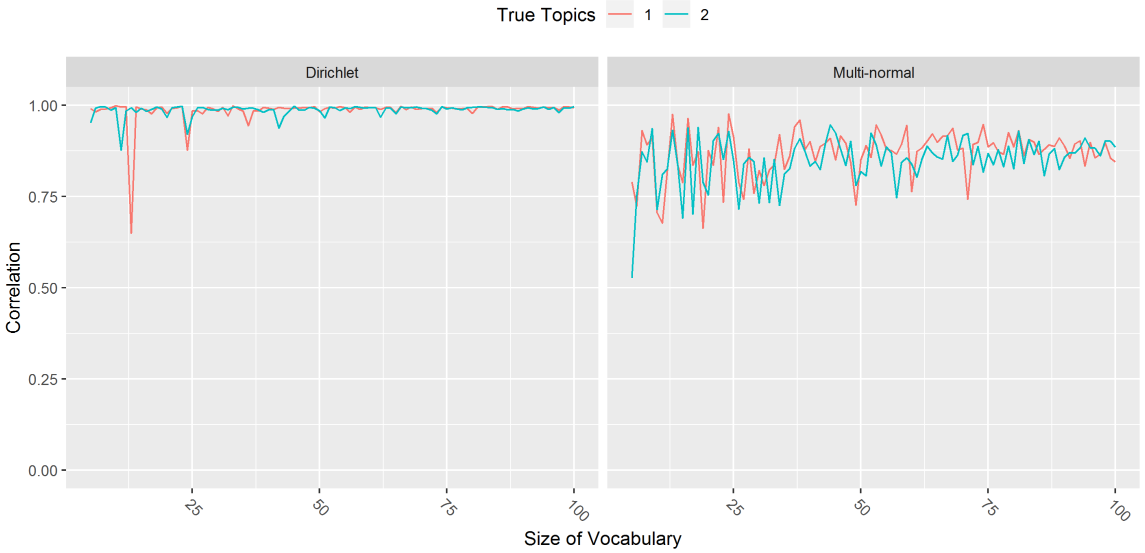

Figure 4.40: Correlation between true and fitted topics with varying size of vocabulary for Case 8

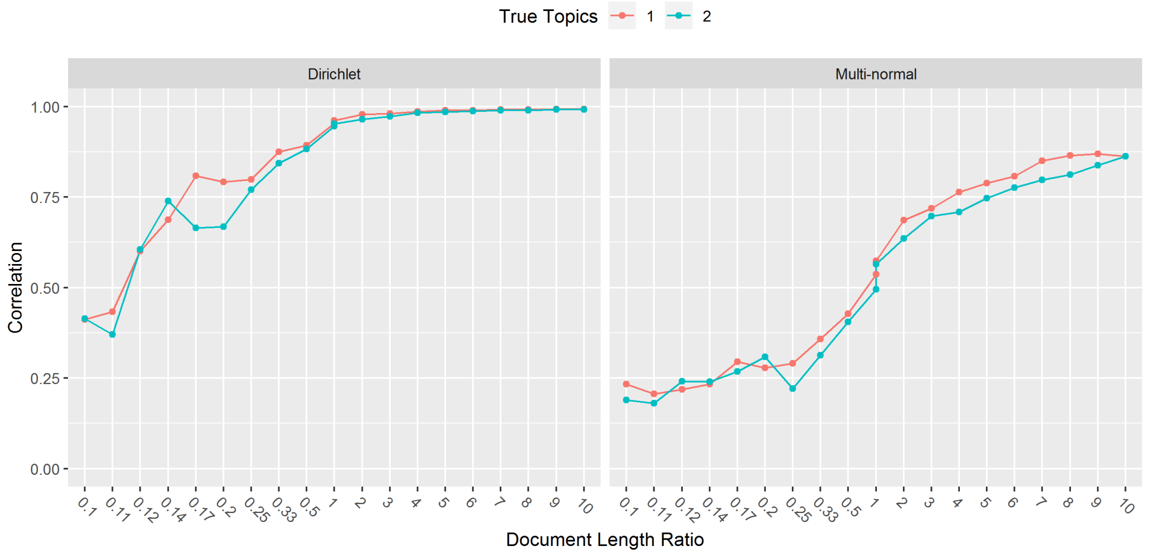

Figure 4.41: Correlation between true and fitted topics with varying document length ratio for Case 9

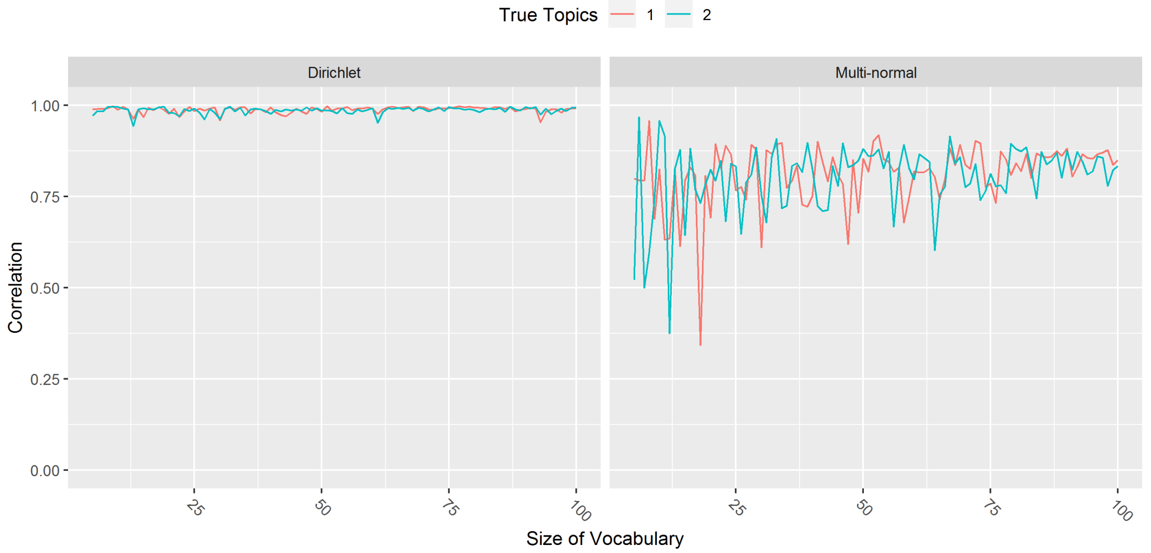

Figure 4.42: Correlation between true and fitted topics with varying size of vocabulary for Case 9

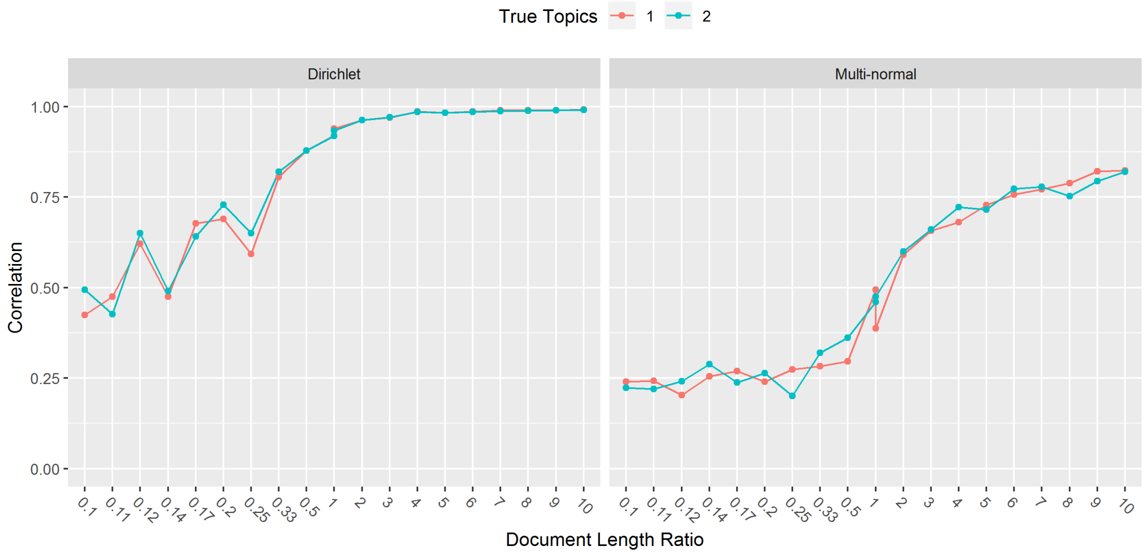

Figure 4.43: Correlation between true and fitted topics with varying document length ratio for Case 10

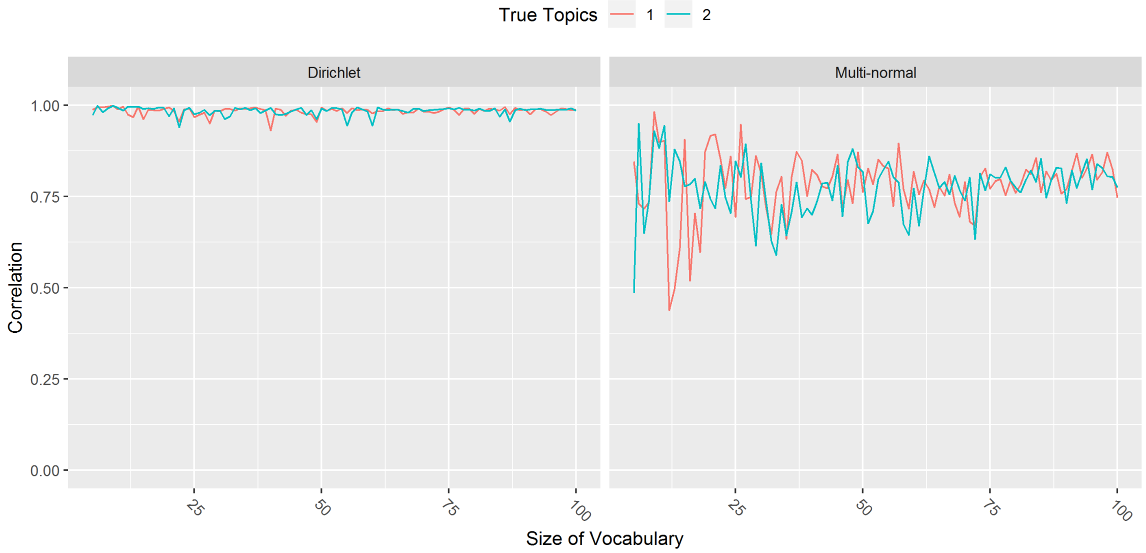

Figure 4.44: Correlation between true and fitted topics with varying size of vocabulary for Case 10

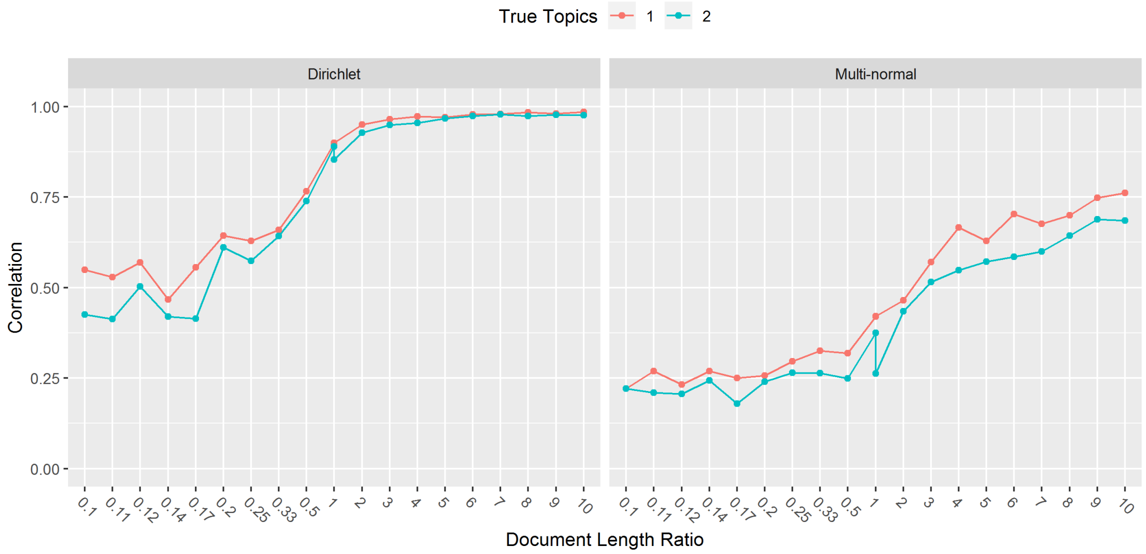

Figure 4.45: Correlation between true and fitted topics with varying document length ratio for Case 11

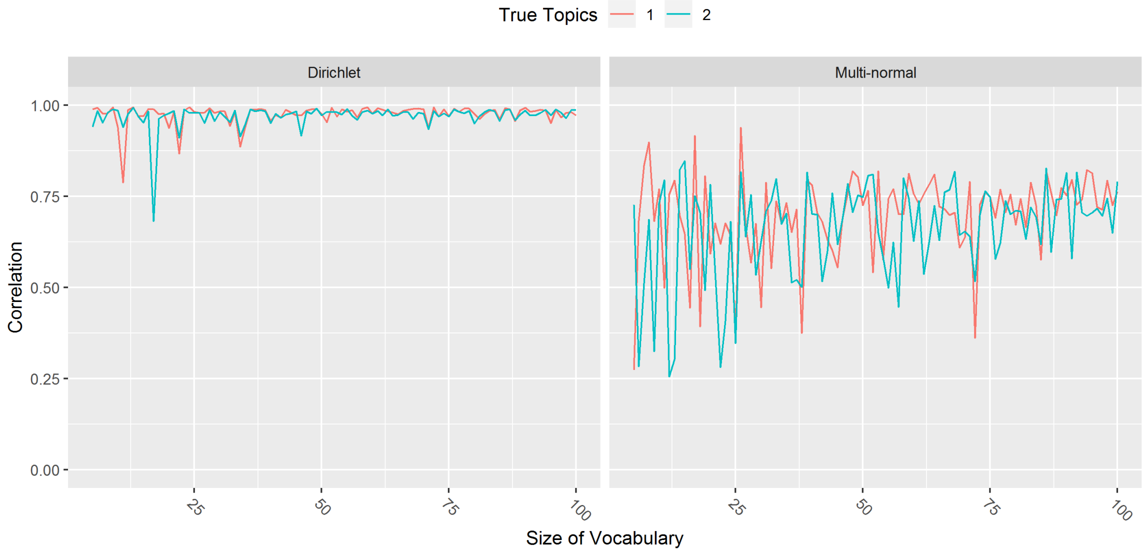

Figure 4.46: Correlation between true and fitted topics with varying size of vocabulary for Case 11

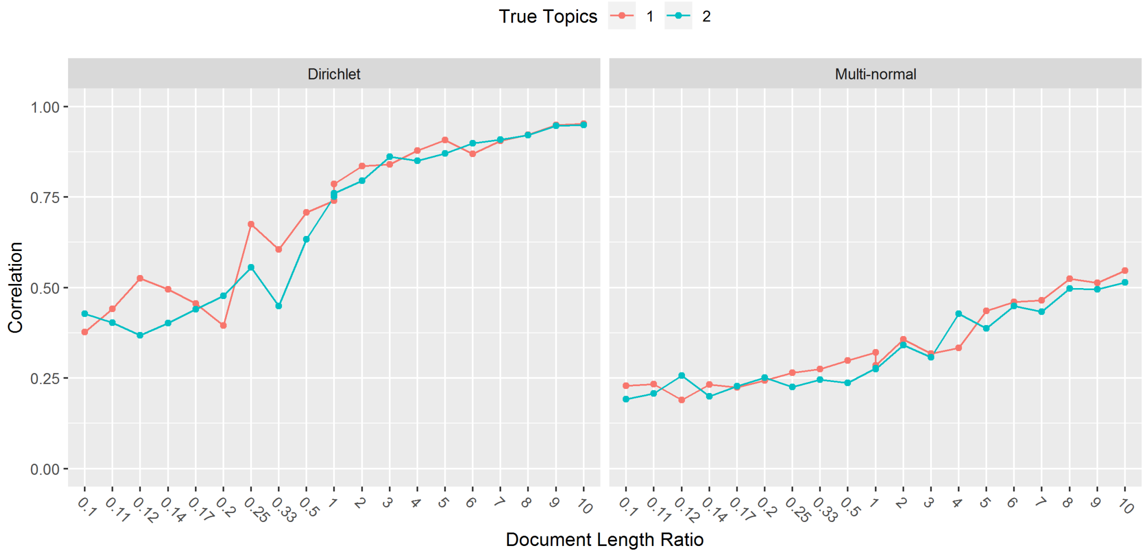

Figure 4.47: Correlation between true and fitted topics with varying document length ratio for Case 12

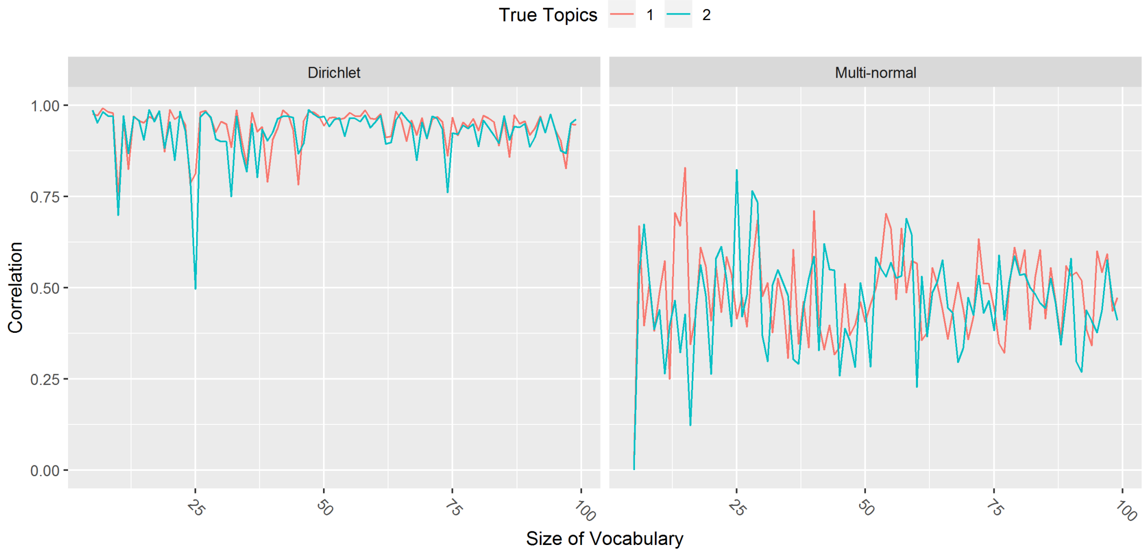

Figure 4.48: Correlation between true and fitted topics with varying size of vocabulary for Case 12

Chapter 5

Analysis of Results and Future Work

5.1 Analysis of Result

5.1.1 Summary of the Analysis of Single Factors

We found that although the LDA model results are supposed to applicable for any type of text documents, the selection of the prior distribution has an important impact on the quality of LDA results. When switching from the Dirichlet prior to the Multi-normal prior, the quality of the LDA results (as measured by correlation between actual and fitted topics) is decreased on average from 0.845 to 0.281 (the average of the correlations using the Multi-normal prior are only 33.25% of the average correlation using the Dirichlet prior). For the worst case scenarios (when considering the minimum correlation values for each simulation setups), the quality of the LDA results measured by correlation is decreased on average from 0.554 to 0.307 (the average of the correlations using the Multi-normal prior are 55.42% of the average correlation using the Dirichlet prior). For the best case scenarios (when considering the maximum correlation values for each simulation setups), the quality of the LDA results measured by correlation is decreased on average from 0.963 to 0.138 (the average of the correlation using the Multi-normal prior are only 14.33% of the average correlation using the Dirichlet prior).

We found that the quality of the LDA model appeared to have a logarithmic relationship with the document length ratio. Numerically, the relationship between the document length ratio and the quality of LDA model measured by correlation is computed based on sample of size 10 (documents). As the document length ratio changed from 1/8 to 1/4 (as measured in the real corpora in Chapter 4) the quality of the LDA results increased by 15.5% (from 0.58 to 0.67) under the Dirichlet prior situation and by 21.7% (from 0.23 to 0.28) under the Multi-normal prior situation.

We found that the quality of LDA result did not really change with the changes in the size of vocabulary.

We found the quality of the LDA results did change as document structure changed. Numerically, the quality decreased up to 55.2% from the best case scenario to the worst case scenario. From the mixture cases, we found that the quality of LDA results are not linearly related to the document structure. From the analytical mixture cases, we found that although the document structure (as measured by the difference between the proportion of the two topics) are evenly spaced (from (0.425,0.575) to (0.475,0.525)), the difference in quality of the LDA results is not evenly spaced. Also, the correlation differences due to document structure change between the Dirichlet prior situation and the Multi-normal prior situation (suggesting an interaction). We found that a 10% increase in the amount that a document is constructed based on one topic may increase the quality of the LDA results by 34.6% (from 0.25 to 0.723).

5.1.2 Details of the Analysis of Single Factors

5.1.2.1 Topic-word Distribution

As stated in Chapter 4, the true type of prior distribution can not be identified in real corpus. We note that the LDA model assumes the Dirichlet prior of the topic-word distribution and the estimation methods are developed based on this prior distribution. Hence, we would anticipate the LDA results get worse when using any prior distributions other than the Dirichlet distribution. But in real world applications there is no guarantee that the topic-word distribution is actually the Dirichlet distribution. Moreover, it is impossible to verify if the prior distribution is the Dirichlet, as assumed.

To analyze the performance of the LDA model under different prior distributions, we compared the correlation results between the simulations using Dirichlet prior and the simulations using the Multi-normal prior. Table 5.1 and Table 5.2 summarized the results of this comparison. The "Diff" column in the tables indicates the difference in the correlation values (correlation of actual and fitted topics) between the "Dirichlet" column and "Multi-normal" column. In the first column named "Simulation", each row represent a specific simulation setup. For the cases that ended with letter "D", the result is from a document length simulation; for the cases that ended with letter "V", the result is from a size of vocabulary simulation. i.e., the row of "Case 1 - D" is summarized from the simulation of document length ratio for Case 1 (Figure 4.25).

We observed that the prior distribution of the topic-word distribution does matter to the quality of the LDA results. From Table 5.1, we found that the mean correlations for the Dirichlet prior situations are higher than the Multi-normal prior situation by about 0.25. Assuming that the simulation cases represent a random sample of possible scenarios in which the priors could be compared, a paired t-test was used to compring these means. The results is listed below:

```
t = 13.585, df = 25, p-value = 4.809e-13
95 percent confidence interval:
0.2381347 0.3232408
```

Based on the result of hypothesis testing, we found that we have sufficient evidence (using $\alpha = 0.05$) to conclude that the selection of the prior distribution of the topic-word distribution does have significant impact to the mean of the correlation results of LDA model, and if the true distribution is not actually Dirichlet the mean correlation is reduced. The lower bound of the 95% confidence interval is about 0.24 and it is relatively large compared with the value of correlations.

The standard deviations of the correlation from Table 5.1 were also compared. We again conducted a paired t-test and the results are below.

```
t = -2.1702, df = 25, p-value = 0.03969
95 percent confidence interval:
-0.06798479 -0.00177934
```

Based on the result of hypothesis testing, we again found that we have sufficient evidence (using $\alpha = 0.05$) to conclude that the selection of the prior distribution of the topic-word distribution has an impact on the standard deviation of the correlation results of the LDA model. This observation indicates that the selection of the prior distribution does affect the consistency of the

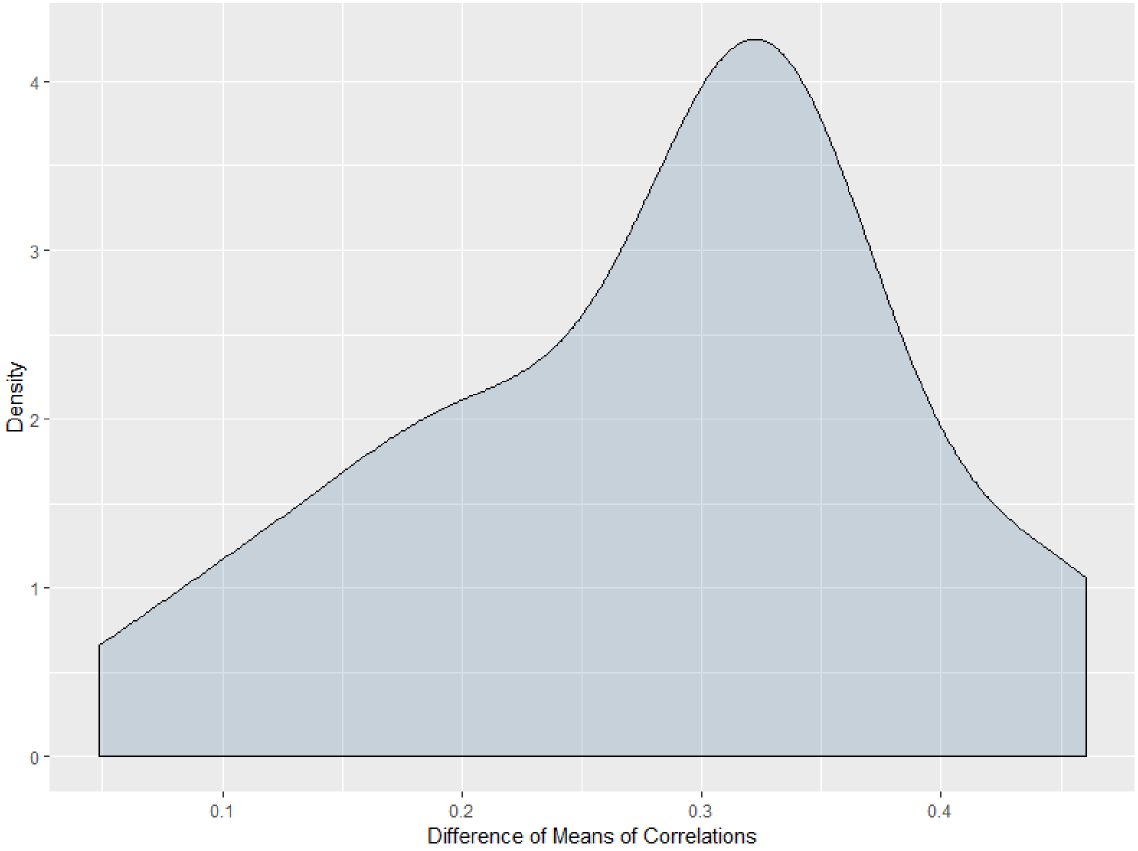

Figure 5.1: Approximate Density of the Difference between Means of Correlations

LDA correlation results, and that if the actual distribution is not actually Dirichlet, the correlations are more variable (the consistency is reduced).

Figure 5.1 and Figure 5.2 are the density plots for the differences between the two situations. The density plots support the conclusion that mean differences are greater than 0 and the standard deviation differences are slightly less than 0.

In Table 5.2, we compared the minimums and maximums of the correlations for different prior distribution selection for the different simulation setups. The minimums represent the “worst case scenario” of the agreement between the actual and fitted topics and the maximums represent the “best case scenario” of the agreement between the actual and fitted topics. We found that for both minimums and maximums, the results of the Dirichlet prior situation and the Multi-normal prior situation are significantly different. T-test results are shown in the following.

For the minimums:

Simulation	Mean			Standard Deviation		
	Dirichlet	Multi-normal	Diff	Dirichlet	Multi-normal	Diff
Document Length Ratio	0.8630495	0.5928935	0.27015601	0.173282217	0.28601742	-0.11273521
Size of Vocabulary	0.9957998	0.9470680	0.04873178	0.003635918	0.03013319	-0.02649727
Case 1 - D	0.9233224	0.6393638	0.28395857	0.085413442	0.27851440	-0.19310096
Case 2 - D	0.4885453	0.2863155	0.20222980	0.058117816	0.04120124	0.01691658
Case 3 - D	0.8412524	0.5067247	0.33452762	0.165915310	0.24602359	-0.08010828
Case 4 - D	0.8895837	0.5644057	0.32517799	0.124914198	0.28018805	-0.15527385
Case 5 - D	0.6820944	0.3395806	0.34251382	0.209001697	0.09937437	0.10962733
Case 6 - D	0.6403566	0.3049266	0.33542998	0.179568575	0.05625423	0.12331435
Case 7 - D	0.5324785	0.2979629	0.23451566	0.080277692	0.04736699	0.03291070
Case 8 - D	0.8922837	0.5673559	0.32492777	0.125655189	0.28009770	-0.15444251
Case 9 - D	0.8455099	0.5291157	0.31639411	0.189984112	0.25381480	-0.06383069
Case 10 - D	0.8203782	0.4927432	0.32763502	0.197589036	0.23606903	-0.03847999
Case 11 - D	0.7772636	0.4296286	0.34763493	0.211782389	0.19203737	0.01974502
Case 12 - D	0.6983669	0.3299725	0.36839440	0.207920111	0.11060003	0.09732008
Case 1 - V	0.9909400	0.8949841	0.09595598	0.009959151	0.07169366	-0.06173451
Case 2 - V	0.7231141	0.4102916	0.31282249	0.162365088	0.11566064	0.04670445
Case 3 - V	0.9854407	0.7808597	0.20458095	0.011647888	0.10272567	-0.09107779
Case 4 - V	0.9855505	0.8428656	0.14268483	0.028662505	0.08788960	-0.05922709
Case 5 - V	0.9293559	0.4831696	0.44618635	0.079609538	0.12551777	-0.04590823
Case 6 - V	0.8648660	0.4281191	0.43674693	0.133973932	0.11055977	0.02341417
Case 7 - V	0.7386765	0.4096311	0.32904547	0.168970412	0.11181846	0.05715196
Case 8 - V	0.9863657	0.8585832	0.12778246	0.028664616	0.06540253	-0.03673791
Case 9 - V	0.9872396	0.8080241	0.17921554	0.008795234	0.08991501	-0.08111978
Case 10 - V	0.9846817	0.7793463	0.20533547	0.010832133	0.08238965	-0.07155752
Case 11 - V	0.9727901	0.6779856	0.29480446	0.031059964	0.12863121	-0.09757125
Case 12 - V	0.9320594	0.4715663	0.46049309	0.061744126	0.12637961	-0.06463548
Average	0.845	0.564	0.281	0.106	0.141	-0.0349

Table 5.1: Comparing the Mean and the Standard Deviation of the Correlations between Fitted and True Topics for LDA using the Dirichlet Prior and LDA using the Multi-normal Prior

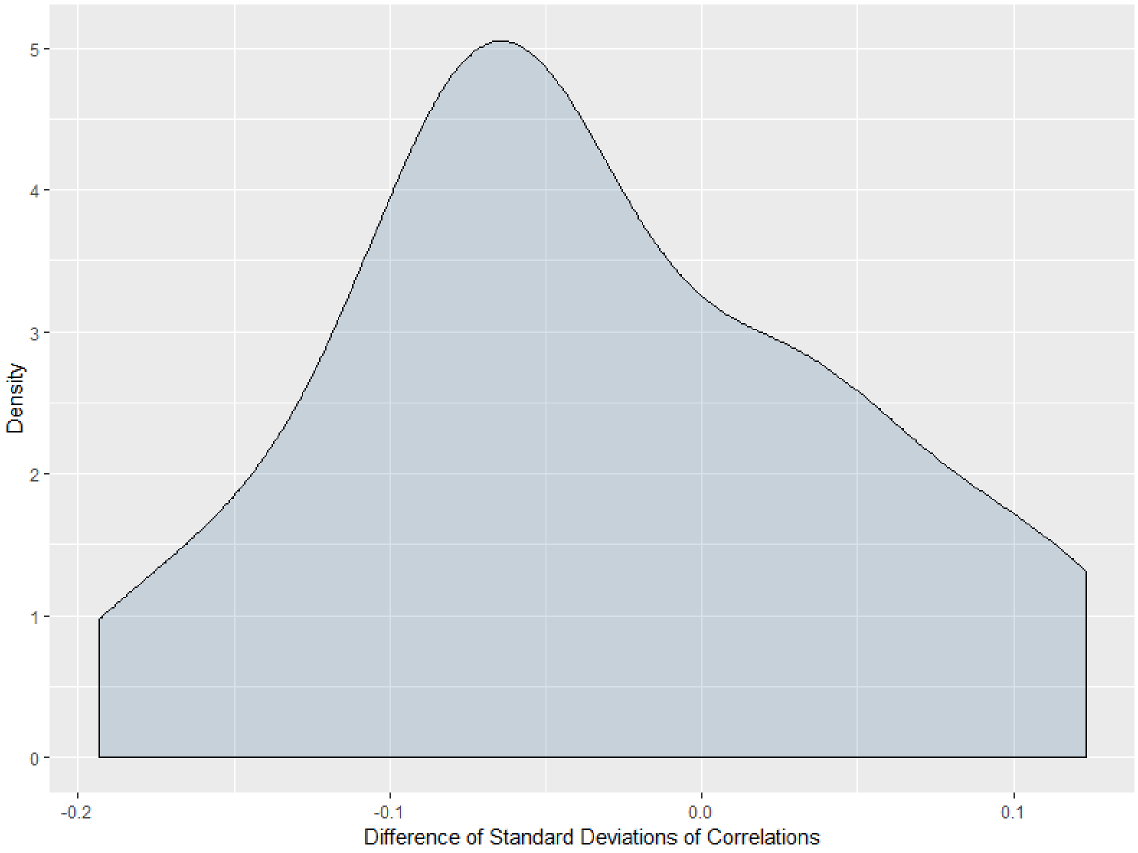

Figure 5.2: Approximate Density of the Difference between Standard Deviations of Correlations

```
t = 9.8112, df = 25, p-value = 4.704e-10
95 percent confidence interval:
0.2423131 0.3710734
```

For the maximums:

```
t = 5.2548, df = 25, p-value = 1.93e-05
95 percent confidence interval:
0.08403519 0.19236665
```

Based on the result of hypothesis testing, we found that under we have sufficient evidence (using $\alpha = 0.05$) to conclude that the selection of the prior distribution of the topic-word distribution has an impact on both the “worst case scenario” and the ”best case scenario” of the correlation results of the LDA model. This observation indicates that the selection of the prior distribution does affect on every aspects of the LDA correlation results, and that if the actual distribution is not actually Dirichlet, the correlations are decreasing.

Since the type of the prior distribution appears to have such a large impact on quality of the LDA results, the details of the analysis of the other factors will be discussed separately for each prior distribution.

5.1.2.2 Document Length Ratio

We summarized the document length ratio results in Table 5.3 (Dirichlet Prior) and Table 5.4 (Multi-normal Prior). For each table, the values are computed based on the observed correlations from all the simulation setups for the given document length ratio. The means of the correlations appears to have a logarithmic relationship with the document length ratio. The linear regression results between the correlation and the logarithm of the document length ratio is shown below:

For the Dirichlet prior situation:

```
Call:
lm(formula = m ~ log(level), data = docl_d)


Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.761114   0.008442   90.16  < 2e-16 ***
log(level)  0.082810   0.005077   16.31 3.15e-12 ***
```

Simulation	Min			Max		
	Dirichlet	Multi-normal	Diff	Dirichlet	Multi-normal	Diff
Document Length Ratio	0.4136946	0.1808482	0.2328464	0.9953444	0.9572978	0.038046576
Size of Vocabulary	0.8982850	0.4607374	0.4375476	0.9986916	0.9907327	0.007958923
Case 1 - D	0.7221651	0.18991219	0.53225296	0.9970085	0.9633320	0.033676481
Case 2 - D	0.3572045	0.19301718	0.16418727	0.6140260	0.3711413	0.242884687
Case 3 - D	0.5331328	0.19951180	0.33362104	0.9898852	0.8453508	0.144534361
Case 4 - D	0.5401770	0.20892958	0.33124738	0.9947616	0.9158051	0.078956573
Case 5 - D	0.3981657	0.20763680	0.19052892	0.9529343	0.5317158	0.421218512
Case 6 - D	0.3740322	0.21193152	0.16210072	0.9064544	0.4325858	0.473868620
Case 7 - D	0.3635195	0.22297562	0.14054384	0.7157354	0.4055980	0.310137425
Case 8 - D	0.5991129	0.17564703	0.42346591	0.9960429	0.9352530	0.060789883
Case 9 - D	0.3706171	0.18112562	0.18949151	0.9931609	0.8690341	0.124126781
Case 10 - D	0.4241140	0.20072349	0.22339047	0.9912712	0.8234239	0.167847289
Case 11 - D	0.4130986	0.17924078	0.23385787	0.9846282	0.7618011	0.222827143
Case 12 - D	0.3678986	0.18969382	0.17820478	0.9522259	0.5473254	0.404900533
Case 1 - V	0.8982850	0.46073739	0.43754756	0.9986916	0.9907327	0.007958923
Case 2 - V	0.4094030	0.12947832	0.27992466	0.9956687	0.8923595	0.103309188
Case 3 - V	0.8967249	0.26848390	0.62824103	0.9969378	0.9503589	0.046578914
Case 4 - V	0.6322713	0.46169570	0.17057562	0.9985751	0.9766019	0.021973187
Case 5 - V	0.3664223	0.22589241	0.14052992	0.9953750	0.8965963	0.098778767
Case 6 - V	0.3095097	0.22650972	0.08300001	0.9935278	0.8658685	0.127659340
Case 7 - V	0.4125841	0.09056111	0.32202298	0.9910559	0.8253392	0.165716715
Case 8 - V	0.6498209	0.52605254	0.12376840	0.9989129	0.9766573	0.022255619
Case 9 - V	0.9431696	0.34320869	0.59996096	0.9978920	0.9671153	0.030776727
Case 10 - V	0.9312748	0.43920921	0.49206559	0.9991755	0.9821331	0.017042405
Case 11 - V	0.6826331	0.25638958	0.42624352	0.9949328	0.9385297	0.056403117
Case 12 - V	0.4968585	0.00000000	0.49685852	0.9920525	0.8290552	0.162997255
Average	0.554	0.247	0.307	0.963	0.825	0.138

Table 5.2: Comparing the Min and the Max of the Correlations between Fitted and True Topics for LDA using the Dirichlet Prior and LDA using the Multi-normal Prior

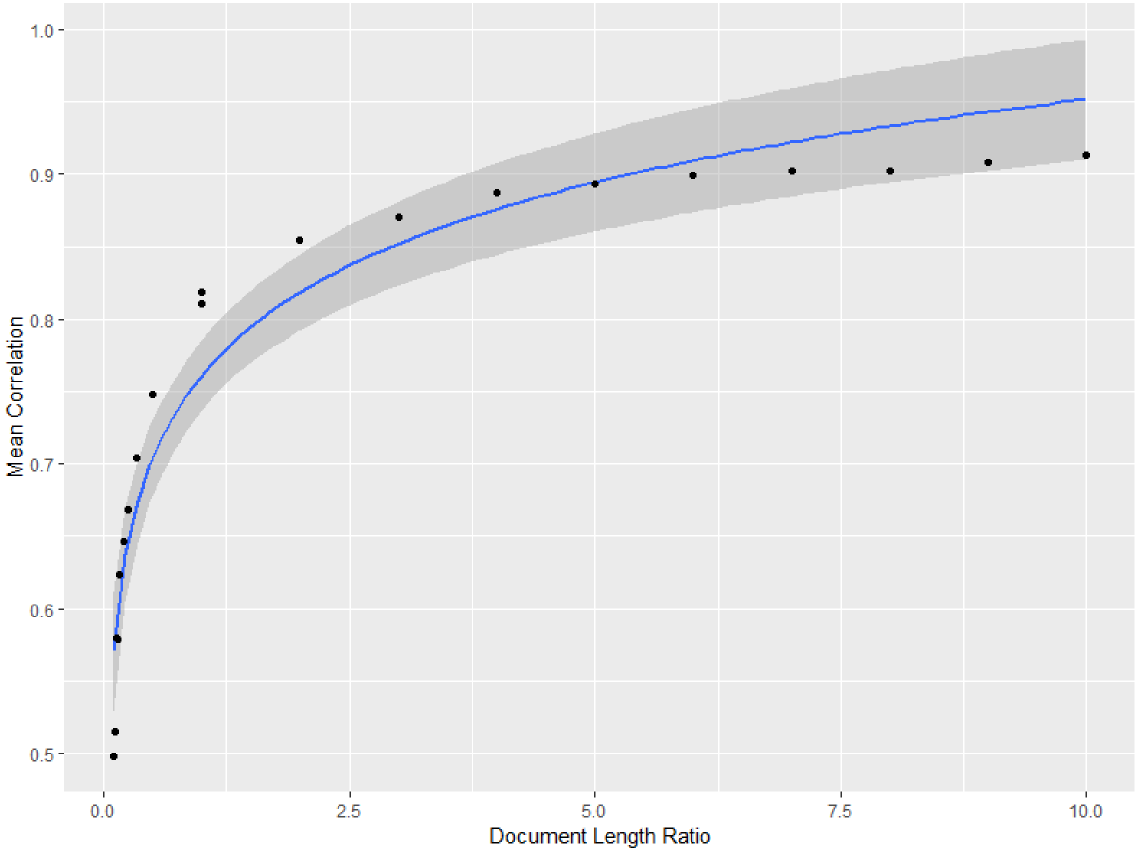

Figure 5.3: The Regression Plot of the Mean Correlation vs logarithm of the Document Length Ratio, Dirichlet prior

For the Multi-normal prior situation:

```
lm(formula = m ~ log(level), data = docl_n)


Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.452384   0.002460  183.90   <2e-16 ***

log(level)  0.106757   0.001479   72.17   <2e-16 ***
```

The regression plot for the Dirichlet prior situation is illustrated in Figure 5.3, and the regression plot for the Multi-normal prior situation is illustrated in Figure 5.4. A 99% confidence interval is also showed in the plots. The results showed that even though correlations increase for both, the quality of the LDA model is much more predictable when the multi-normal prior is used.

The standard deviation of the correlations shows a similar logarithmic relationship with

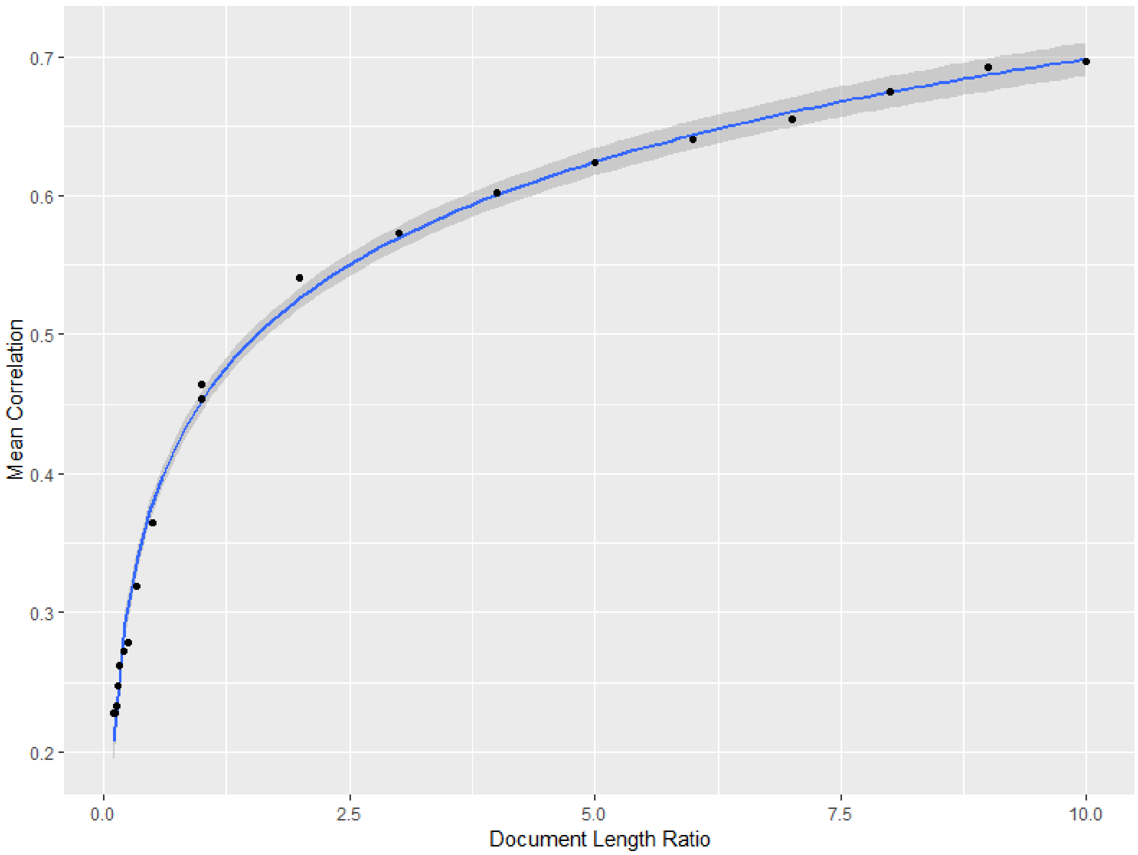

Figure 5.4: The Regression Plot of the Mean Correlation vs logarithm of the Document Length Ratio, Multi-normal prior

mean when the multi-normal prior is adopted. When the Dirichlet prior is adopted, the standard deviation gets relatively stable after the document length ratio is greater than 1. The regression results are shown below.

For the Dirichlet prior situation:

```
lm(formula = std ~ log(level), data = docl_d_std)


Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.156794   0.004611  34.003   <2e-16 ***
log(level)  0.003305   0.002773   1.192    0.249
```

For the Multi-normal prior situation:

```
lm(formula = m ~ log(level), data = docl_n_sd)


Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.138737   0.002651   52.33   <2e-16 ***
log(level)  0.050487   0.001594   31.67   <2e-16 ***
```

The regression plot for the Dirichlet prior situation is illustrated in Figure 5.5, and the regression plot for the Multi-normal prior situation is illustrated in Figure 5.6. A 99% confidence interval is also showed in the plots. We can see from the plots and Table 5.3 that the standard deviation for the Dirichlet prior situation reached to its maximum when the document length ratio is one third and then decreased and stabilized after 4. On the other hand, the

The minimum and maximum of the correlations are also tested through regression. The regression results are shown below. Figure 5.7 and Figure 5.8 illustrate the regression lines for minimum of the correlation, and Figure 5.9 and Figure 5.10 illustrate the regression lines for maximum of the correlation. We can see that although the regression coefficients for all the regression models are significant (at $\alpha = 0.05$), the regression plots show that the fitting is not ideal. Hence, even though we have not reached the best possible model of the relationship, we still can conclude that the "worst case scenario" and the "best case scenario" does have positive relationship with document length ratio and the correlations increased as the document length ratio increased.

For the minimum of the correlation, Dirichlet prior situation:

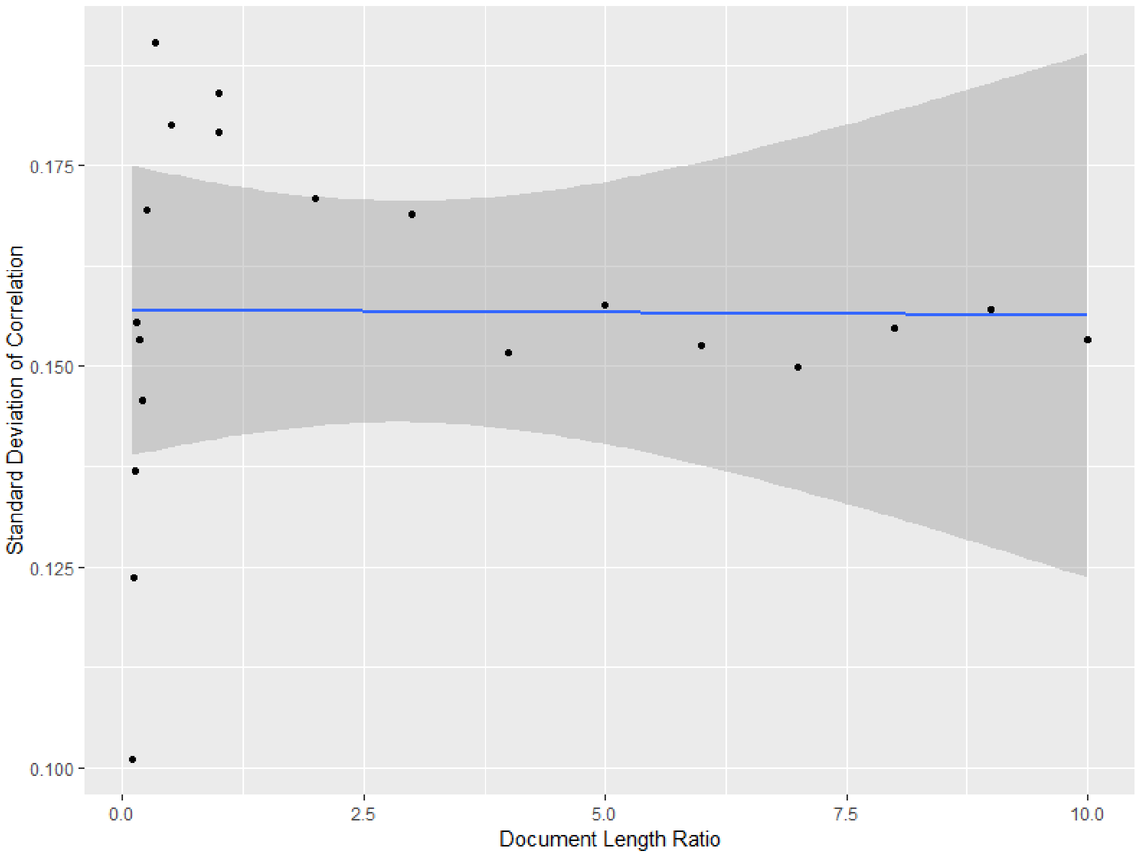

Figure 5.5: The Regression Plot of the Standard Deviation of Correlation vs logarithm of the Document Length Ratio, Dirichlet prior

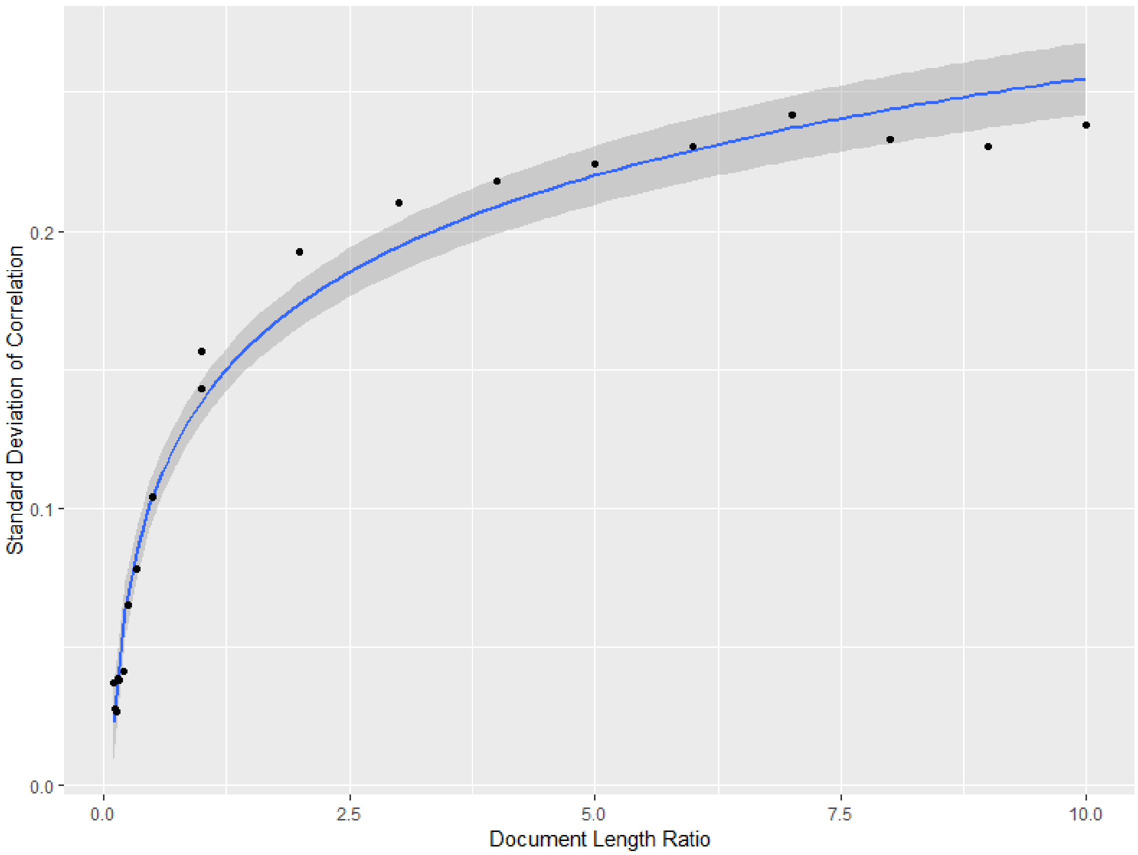

Figure 5.6: The Regression Plot of the Standard Deviation of Correlation vs logarithm of the Document Length Ratio, Multi-normal prior

```
lm(formula = min ~ log(level), data = docl_d_min)

Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.448146   0.005439  82.388  < 2e-16 ***
log(level)  0.027317   0.003271   8.351 1.32e-07 ***
```

For the minimum of the correlation, Multi-normal prior situation:

```
lm(formula = m ~ log(level), data = docl_n_min)

Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.253965   0.003192   79.55  < 2e-16 ***
log(level)  0.030070   0.001920   15.66 6.24e-12 ***
```

For the maximum of the correlation, Dirichlet prior situation:

```
lm(formula = max ~ log(level), data = docl_d_max)

Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.929325   0.008150 114.033  < 2e-16 ***
log(level)  0.043067   0.004901   8.787 6.28e-08 ***
```

For the maximum of the correlation, Multi-normal prior situation:

```
lm(formula = m ~ log(level), data = docl_n_max)

Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.669265   0.009290   72.04  < 2e-16 ***
log(level)  0.148727   0.005587   26.62 6.58e-16 ***
```

5.1.2.3 Size of Vocabulary

We found that the size of vocabulary does not appear to have much of an impact on the quality of the LDA results measured by correlation. The overall scatter plots of the size of vocabulary vs correlation are shown in Figure 5.11 (Dirichlet prior situation) and Figure 5.12 (Multi-normal

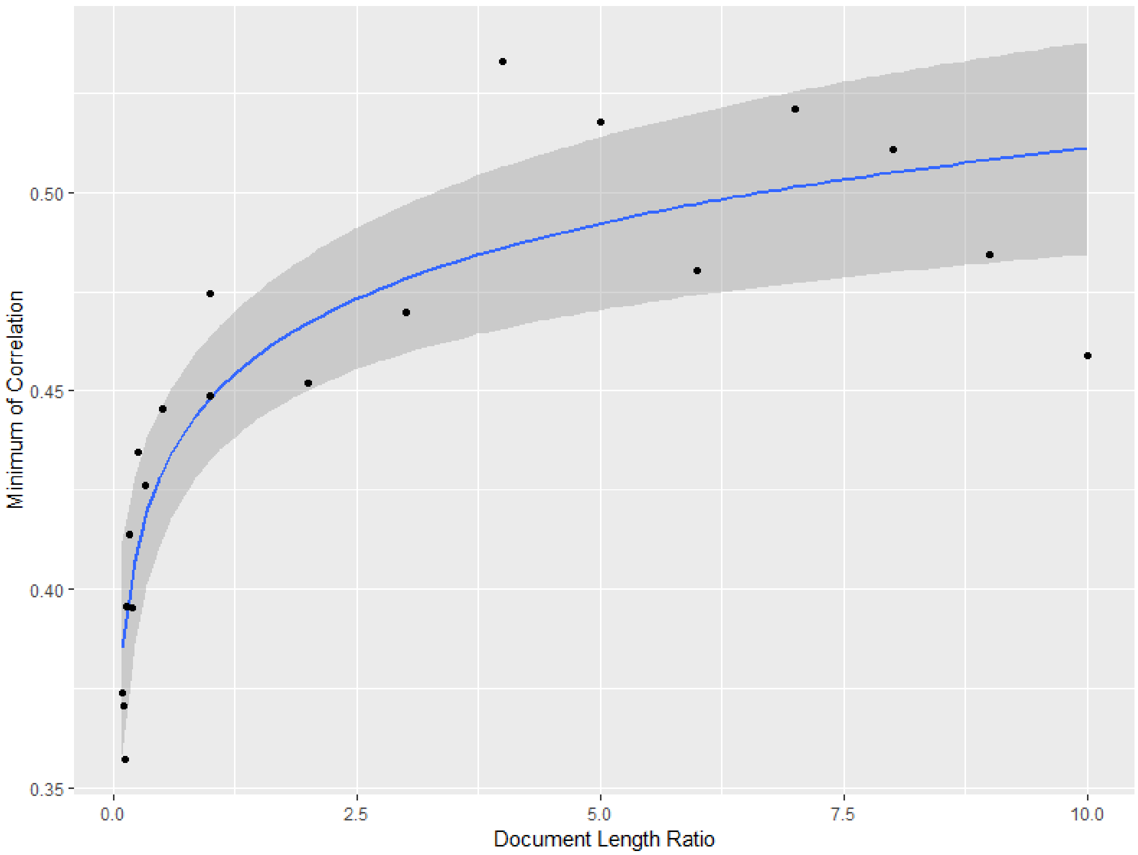

Figure 5.7: The Regression Plot of the Minimum of Correlation vs logarithm of the Document Length Ratio, Dirichlet prior

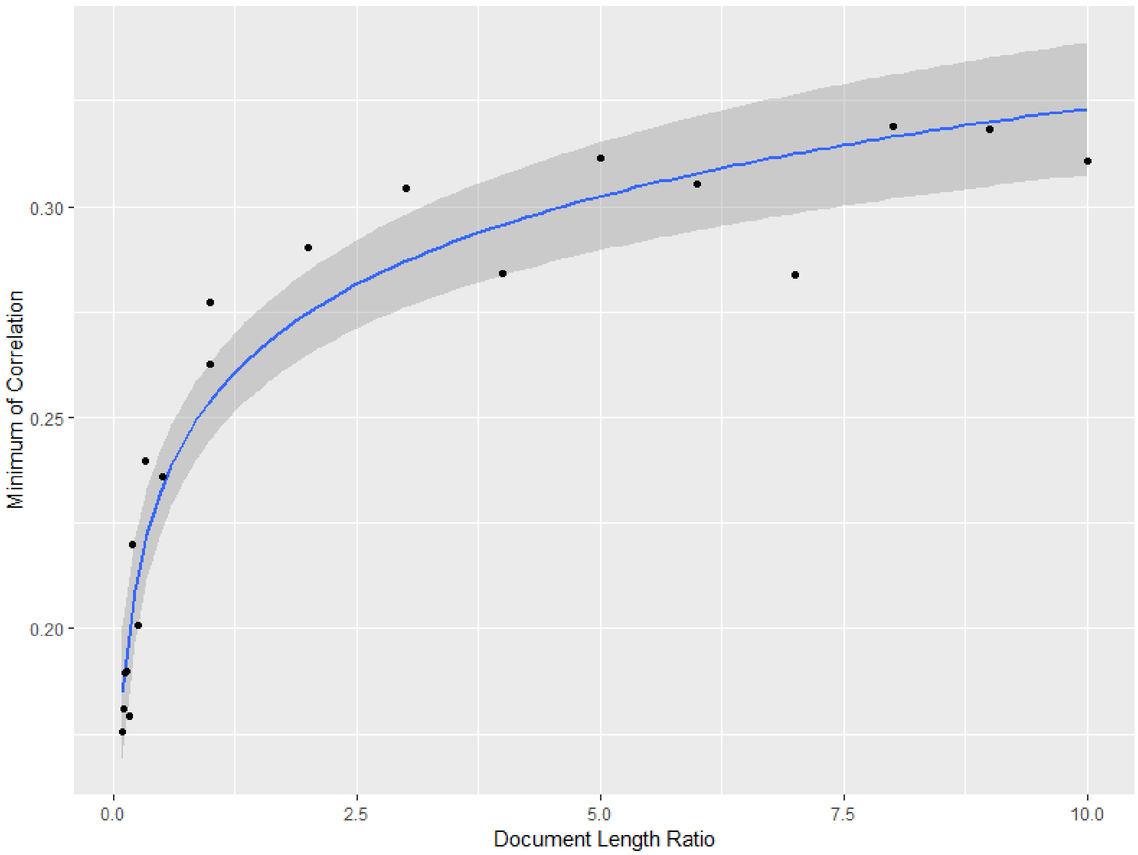

Figure 5.8: The Regression Plot of the Minimum of Correlation vs logarithm of the Document Length Ratio, Multi-normal prior

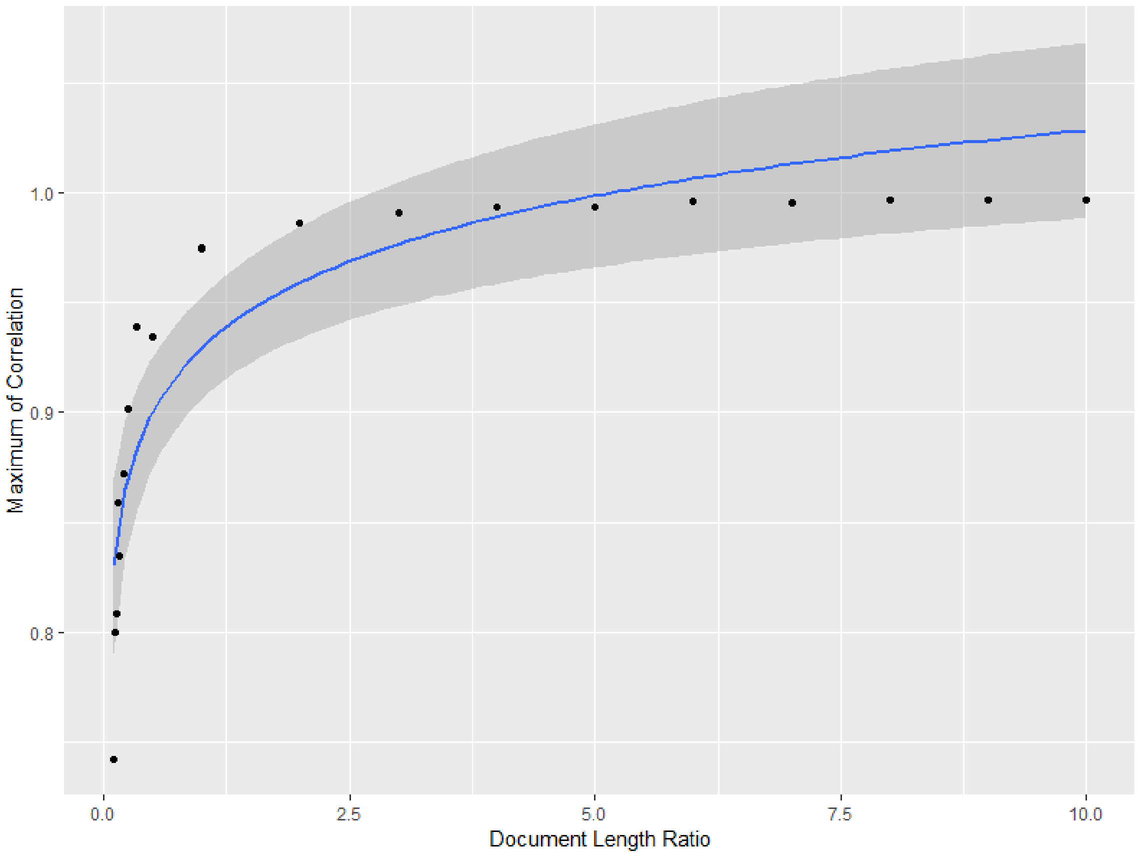

Figure 5.9: The Regression Plot of the Maximum of Correlation vs logarithm of the Document Length Ratio, Dirichlet prior

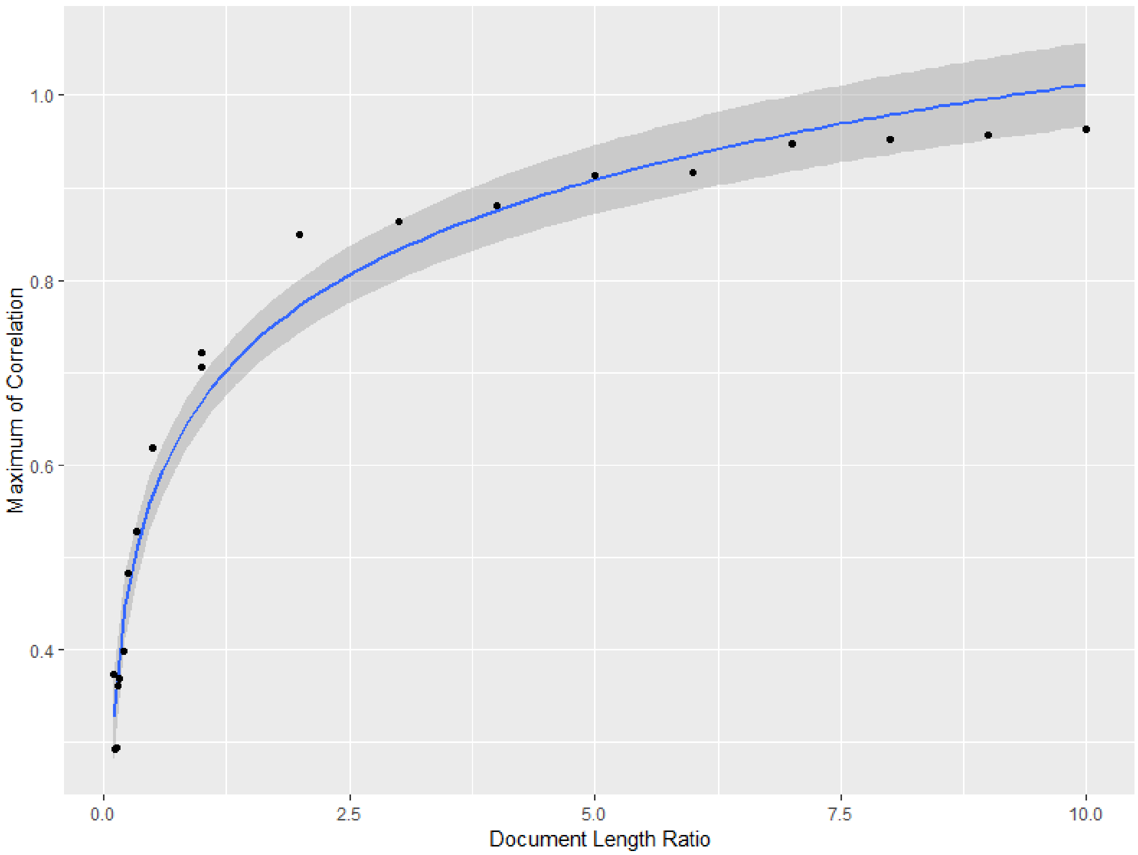

Figure 5.10: The Regression Plot of the Maximum of Correlation vs logarithm of the Document Length Ratio, Multi-normal prior

Level	Document Length Ratio	Mean	Standard Deviation	Min	Max
1	1/10	0.498	0.101	0.374	0.742
2	1/9	0.515	0.124	0.371	0.800
3	1/8	0.580	0.137	0.357	0.808
4	1/7	0.578	0.155	0.396	0.859
5	1/6	0.623	0.153	0.414	0.835
6	1/5	0.646	0.146	0.395	0.872
7	1/4	0.668	0.169	0.435	0.901
8	1/3	0.704	0.190	0.426	0.939
9	1/2	0.748	0.180	0.445	0.935
10	1	0.811	0.184	0.474	0.975
11	2	0.855	0.171	0.452	0.987
12	3	0.871	0.169	0.470	0.991
13	4	0.887	0.152	0.533	0.993
14	5	0.894	0.158	0.518	0.993
15	6	0.899	0.153	0.480	0.996
16	7	0.902	0.150	0.521	0.996
17	8	0.902	0.155	0.511	0.997
18	9	0.909	0.157	0.484	0.997
19	10	0.914	0.153	0.459	0.997

Table 5.3: Summary Descriptive Statistics for Each Level of Document Length Ratio with Dirichlet prior

Level	Document Length Ratio	Mean	Standard Deviation	Min	Max
1	1/10	0.228	0.0368	0.176	0.373
2	1/9	0.228	0.0277	0.181	0.292
3	1/8	0.232	0.0265	0.190	0.294
4	1/7	0.247	0.0384	0.190	0.360
5	1/6	0.262	0.0382	0.179	0.368
6	1/5	0.272	0.0412	0.220	0.398
7	1/4	0.279	0.0649	0.201	0.483
8	1/3	0.319	0.0781	0.240	0.528
9	1/2	0.365	0.104	0.236	0.619
10	1	0.454	0.143	0.277	0.721
11	2	0.541	0.192	0.290	0.849
12	3	0.573	0.210	0.304	0.863
13	4	0.602	0.218	0.284	0.881
14	5	0.623	0.224	0.312	0.913
15	6	0.641	0.230	0.305	0.917
16	7	0.655	0.242	0.284	0.948
17	8	0.674	0.233	0.319	0.952
18	9	0.692	0.230	0.318	0.957
19	10	0.696	0.238	0.311	0.963

Table 5.4: Summary Descriptive Statistics for Each Level of Document Length Ratio with Multinormal prior

	Estimated Coefficient	P-value
Mean, Multi-normal	0.0004238	0
Mean, Dirichlet	-0.0003801	0
Standard Deviation, Multi-normal	0.0001318	0.06
Standard Deviation, Dirichlet	0.0007077	0
Min, Multi-normal	0.0005447	0.002
Min, Dirichlet	-0.0001886	0.0003
Max, Multi-normal	-0.0000226	0.79
Max, Dirichlet	-0.0000056	0.316

Table 5.5: Summary of the Estimated regression coefficients of the simple linear regression model for Size of Vocabulary analysis

Size of Vocabulary	Mean	Standard Deviation	Min	Max
5 19	0.955593	0.085632	0.366422	0.999175
20 29	0.938193	0.099161	0.45196	0.998364
30 39	0.93041	0.119114	0.416111	0.998814
40 49	0.916357	0.14733	0.30951	0.997986
50 59	0.922785	0.122707	0.492362	0.997592
60 69	0.926149	0.11653	0.439073	0.997592
70 79	0.915361	0.137696	0.424562	0.997953
80 89	0.920834	0.132057	0.446152	0.996805
90 100	0.920839	0.135886	0.424928	0.997132

Table 5.6: Summary Descriptive Statistics for Categories of Size of Vocabulary with Dirichlet prior

prior situation). There is no observable trend in those plots. We categorized the levels of size of the vocabularies and summarized the descriptive statistics in Table 5.6 and Table 5.7. The simple linear regression analysis for the means, standard deviations, minimums, and maximums also showed that the size of vocabulary has nearly no impact to the correlation (The value of the correlation coefficient is close to zero or the p-value that testing if the coefficient is different with zero is higher than 0.05). Table 5.5 summarized the results.

5.1.2.4 Document-topic Distribution

In the previous sections, we simply used correlation as the measure of quality of the LDA model. We would like to keep using it in the document structure analysis, but there are some problems about that. For each selected document-topic distribution, it is impossible to selected one set of parameters (size of vocabulary, document length ratio, topic-word distributions, etc.) that is the "representative" of all possible combinations of factors. Hence, instead of using one single correlation to evaluate the quality of the LDA model for a specific document-topic distribution, we conducted the analysis of the size of vocabulary and document length ratio for each selected

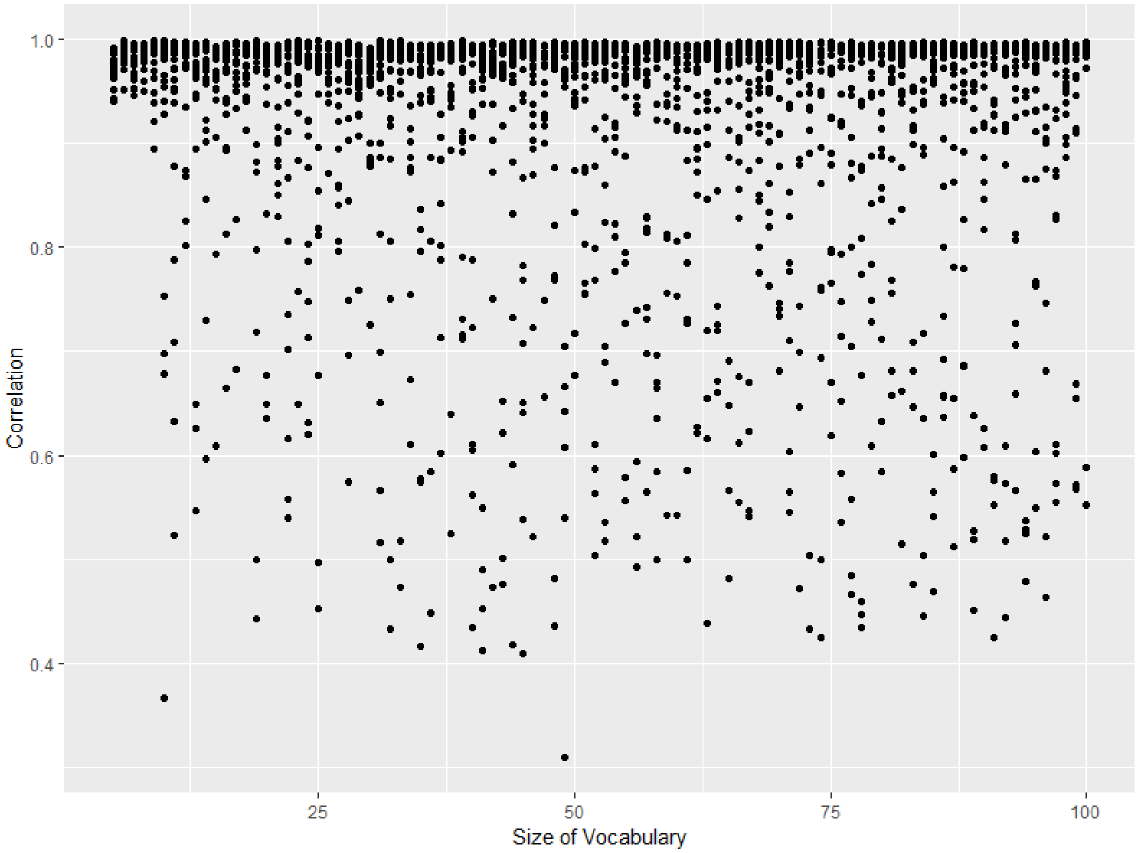

Figure 5.11: Size of Vocabulary vs Correlations between the True topic and Fitted Topic, Dirichlet Prior

Size of Vocabulary	Mean	Standard Deviation	Min	Max
5 19	0.66046	0.210222	0.090561	0.990733
20 29	0.670258	0.211709	0.22651	0.976657
30 39	0.659067	0.212596	0.243198	0.960194
40 49	0.665957	0.21917	0.245726	0.946145
50 59	0.674401	0.221061	0.237259	0.952297
60 69	0.674405	0.213673	0.228559	0.959229
70 79	0.677123	0.215745	0.281333	0.961233
80 89	0.683308	0.226531	0.26202	0.959477
90 100	0.701945	0.215614	0.269247	0.96488

Table 5.7: Summary Descriptive Statistics for Categories of Size of Vocabulary with Multi-normal prior

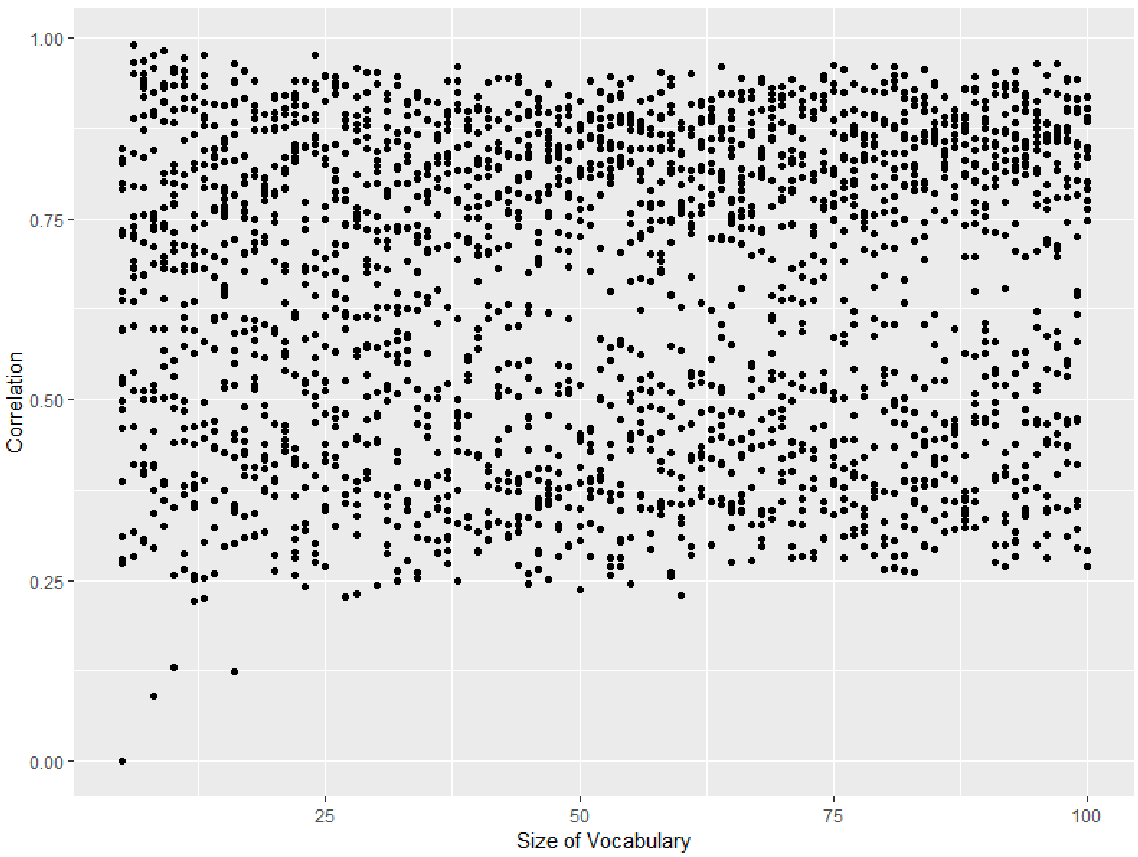

Figure 5.12: Size of Vocabulary vs Correlations between the True topic and Fitted Topic, Multinormal Prior

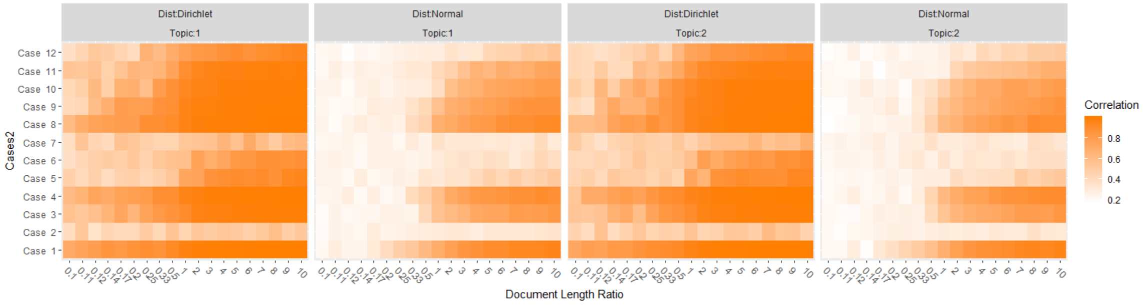

Figure 5.13: Heatmap Summary of Document Length Ratio Simulation Results

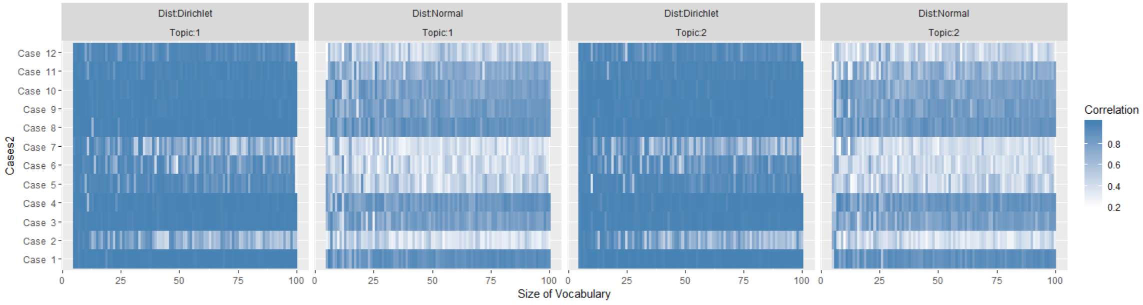

Figure 5.14: Heatmap Summary of Size of Vocabulary Simulation Results

document-topic distribution so that we can learn a lot more in details about the impact of the document structure. To avoid a large and complicated table, we use pictures so that it is easier to compare between the results from different document-topic distributions. Figure 5.13 and 5.14 are the heat-maps of the simulation results for document-topic distribution from Chapter 4. The strength of the color is related to the value of the correlation.

Table 5.8 shows the mean correlations for the two extreme cases, and the difference between the two. The percentage is found by dividing the difference by the Case 1 value. The results suggest that the document structure does affect the quality of the LDA results. Numerically, the quality measured by correlations can decrease up to 55.2% from the best case scenario to the worst case scenario.

Table 5.9 shows the mean correlations for the two mixture cases. "Diff 1-3" means the difference between Case 1 and Case 3. Although the two mixture cases are designed to be in the middle between the two extreme cases, we find that both Case 3 and Case 4 are closer to Case 1 than to Case 2. This suggests that the quality of LDA results are not changing in a linear fashion with respect to the document length ratio.

	Document Length Ratio		Size of Vocabulary	
Prior	Dirichlet	Multi-normal	Dirichlet	Multi-normal
Case 1	0.923	0.639	0.991	0.895
Case 2	0.489	0.286	0.723	0.410
Diff	0.435	0.353	0.268	0.485
Percentage	47.1	55.2	27.0	51.2

Table 5.8: Average Correlation for Extreme Cases

	Document Length Ratio		Size of Vocabulary	
Prior	Dirichlet	Multi-normal	Dirichlet	Multi-normal
Case 3	0.841	0.507	0.985	0.781
Case 4	0.890	0.564	0.986	0.843
Diff 1-3	0.082	0.132	0.006	0.114
Diff 1-4	0.033	0.075	0.005	0.052
Diff 3-2	0.352	0.221	0.262	0.371
Diff 4-2	0.401	0.278	0.263	0.433

Table 5.9: Average Correlation for Mixture Cases

Table 5.10 shows the mean correlations for the first group of the analytical mixture cases combined with the Case 2. We add the Case 2 results here because the first group of the analytical mixture cases is designed to evaluate the impact of the document-topic distribution when the distribution is close to the Case 2 situation. Table 5.11 shows the differences of the mean correlations. We found that although the document structures measured by the difference between the proportion of two topics are evenly spaced, the quality of LDA results measured by correlations are not. Also, the pattern of the correlation differences are different between the Dirichlet prior situation and the Multi-normal prior situation.

Table 5.12 shows the mean correlations for the second group of the analytical mixture cases combined with the Case 2 again, for the same reason as above. We found that from Case 8 to Case 11, these four cases are mixtures of documents from the two extreme cases. Case 8 contains 50% documents that solely depends on one topic and the other 50% documents are evenly distributed

	Document Length Ratio		Size of Vocabulary	
Prior	Dirichlet	Multi-normal	Dirichlet	Multi-normal
Case 5	0.682	0.340	0.929	0.483
Case 6	0.640	0.305	0.865	0.428
Case 7	0.532	0.298	0.739	0.410
Case 2	0.489	0.286	0.723	0.410

Table 5.10: Average Correlation for Analytical Mixture Cases, First group

	Document Length Ratio		Size of Vocabulary	
Prior	Dirichlet	Multi-normal	Dirichlet	Multi-normal
Diff 5-6	0.042	0.035	0.064	0.055
Diff 6-7	0.108	0.007	0.126	0.018
Diff 7-2	0.043	0.012	0.016	0

Table 5.11: Average Correlation Differences for Analytical Mixture Cases, First group

	Document Length Ratio		Size of Vocabulary	
Prior	Dirichlet	Multi-normal	Dirichlet	Multi-normal
Case 8	0.892	0.567	0.986	0.859
Case 9	0.856	0.529	0.987	0.808
Case 10	0.820	0.493	0.985	0.779
Case 11	0.777	0.430	0.973	0.678
Case 12	0.698	0.330	0.932	0.472
Case 2	0.489	0.286	0.723	0.410

Table 5.12: Average Correlation for Analytical Mixture Cases, Second group

mixtures of two topics. Surprisingly, the quality of the LDA results are not changed as much as what the 50% of mixture might suggest, especially when comparing Figure 4.40 and 4.26. In the left hand side of Figure 4.40, the correlation is almost always close to 1 after a vocabulary size of 25. For Figure 4.39 which presents the document length ratio analysis, the left hand side of the graph shows the correlation higher than 0.75 when the document length ratio is greater than 0.14. This is a very good value when compared with our “good” benchmark in Figure 4.25. However, the right hand side shows the correlation is around 0.25 when the document length ratio is less than 0.25. This is a bad value when compared with our “bad” benchmark in Figure 4.27. The above observation indicates that the LDA model is sensitive to the prior distribution, and the impact of the document length ratio is greater than the size of vocabulary.

We found that a 10% of document that is constructed based on one topic may increase the quality of LDA results measured by correlation for at least 34.6% (0.25 out of 0.723).

5.1.3 Summary of the Interactions between Factors

We mentioned the interaction related to the type of prior distributions and topic-word distribution in the previous sections. In this section, we will also discuss the interactions between the other factors. We primarily focus on the interaction between the document structures and the other two factors.

5.1.4 Details of the Interactions between Factors

5.1.4.1 Document Structure and Document Length Ratio Interaction

The data for comparing Case 1 in Figure 4.25 with Case 2 in Figure 4.25 have been summarized in Table 5.13. We note the following observations:

1. When the document length ratio is less than 1/10, the difference between correlations of Case 1 and Case 2 is 0.3058689 under the Dirichlet prior situation, and only 0.0134565 under the Multi-normal prior situation.

2. The correlation values start to vary after a certain level for each case and each prior distribution.

The above observations suggest that the quality of the LDA model is impacted by the interactions between the document length ratio and the type of the topic-word distribution.

From Case 5, Case 6, and Case 7, we found that the quality of the LDA results measured by correlation does not change much when document length ratio changes, when the ratio is less than 1 and topics are evenly (or close to evenly) distributed within each document. These conditions are true for many real applications.

From Cases 8 to 12, we found that the upper bound of the quality of LDA results measured by correlation changed very little as long as there exists at least one document that is constructed based on only one topic. This leads to the proposal (discussed later) of using pre-processing steps to increase the quality of LDA results.

5.1.4.2 Document Structure and Size of Vocabulary Interaction

As discussed previously, the size of vocabulary itself does not show a strong impact on the quality of LDA results. But it is interesting to find that the quality of LDA results measured in correlation has a decreasing trend as the size of vocabulary increasing under certain configurations. Figure 4.28, 4.34, 4.36, 4.38, and 4.48 all show this result.

5.1.5 Pre-processing Step

Based on the analyses above, we present a pre-processing step which is designed to increase the quality of LDA results measured by correlation.

Document Length Ratio	Dirichlet		Multi-normal	
	Case 1	Case 2	Case 1	Case 2
1/10	0.7320681	0.4261992	0.2233117	0.2098552
1/9	0.7819046	0.463527	0.2580075	0.2229907
1/8	0.7934823	0.4613425	0.2688533	0.2331432
1/7	0.8467171	0.4170367	0.2169964	0.2304878
1/6	0.8317429	0.4433219	0.3203692	0.2753751
1/5	0.8702315	0.4804467	0.3851157	0.2600644
1/4	0.8997272	0.4765887	0.4682863	0.2646225
1/3	0.8927128	0.4511822	0.5159336	0.2655088
1/2	0.9337654	0.5059134	0.6002811	0.2906777
1	0.9732089	0.4804519	0.6976041	0.3089962
2	0.9849651	0.4947157	0.7889799	0.3064896
3	0.9906245	0.4847387	0.8475498	0.3124412
4	0.9922135	0.5350713	0.8792623	0.284833
5	0.9932106	0.5199544	0.9062114	0.3199099
6	0.9952437	0.4819796	0.9168948	0.3138405
7	0.9950622	0.5674986	0.9437485	0.3157109
8	0.9963636	0.5666718	0.9454442	0.3450971
9	0.9964893	0.5188933	0.9459267	0.3282402
10	0.9963287	0.4938605	0.9604776	0.33167

Table 5.13: Compare Case 1 and Case 2 Mean Correlations for different Document Length Ratio

We have observed that when there exists a document that is constructed only from one topic, the quality of the LDA results measured by correlation was increased significantly. Hence, a possible pre-processing step is described as following: For a given corpus, if there exists one topic-word distribution that is believed to contained only of the topics, we can then generate a document based on this topic-word distribution, and then add this document to the corpus to increase the quality of the LDA results.

This pre-processing step is similar with adding an informative prior. But there is one difficulty with using informative priors we could not resolve, namely the informative prior affects all topic-word distributions at the same time. The pre-processing step was designed to impact only selected topic-word distributions. Also, it turned out to be difficult (if not impossible) to determine how to adjust the prior based on our belief of the existing of the one specific topic. We did not pursue this approach any further in this research since the pre-processing step as stated is relatively easy to adapt, and works well in increasing the quality of LDA.

The following simulation verified that the pre-processing step does increase the quality of LDA results.

Factor	Value
Prior Distribution	Multi-normal
Size of Vocabulary	1000
Document Length Ratio	0.2
Number of Documents	300
Number of Topics	2

Table 5.14: Parameter Selection for Pre-processing Corpus Construction

	LDA Topic 1	LDA Topic 2
True Topic 1	0.2817226	0.3097747
True Topic 2	0.2819809	0.3186270

Table 5.15: Correlations between the True Topics and Fitted LDA Topics for the Original Corpus

5.1.5.1 Simulation to Verify the Effectiveness of the Pre-processing Step

The pre-processing step as described above turns the unsupervised method to a semi-supervised method. We considered comparing our method with other semi-supervised methods in the quality increasing. However it is difficult to make a valid comparison because it is difficult to evaluate the amount of supervision a method requires. For now, we present a simulation study to learn if the pre-processing step actually increase the quality. An important extension of this research would be to determine exactly how the improvement found for the pre-processing step compares to the improvement for other semi-supervised methods. We will create a corpus such that each document in this corpus is constructed only from one topic. Hence, the document structure is a generalization of the Case 1. Table 5.14 shows the value we take for other necessary parameters. We take the document length ratio to be 0.2 based on the observations of the real corpus in Chapter 4. It is worth noting that it is very rare for a real corpus to contain only two topics. Hence, those corpus that contain only two topics usually does not have many documents. We choose the number of documents to be 300 so that it is large enough to mimic many real world documents applications.

Next, we perform the LDA model to find the estimated topics. Figure 5.15 shows the correlation structure of the fitted topics and the true topics. Table 5.15 illustrates the correlations between the topics. Based on the same analyses used in Chapter 4, we find the mean correlation for this result is 0.2958778.

Then we construct a single document that is only from one topic that has the same length with others and add this document to the corpus, fit LDA model again and computed the correlations

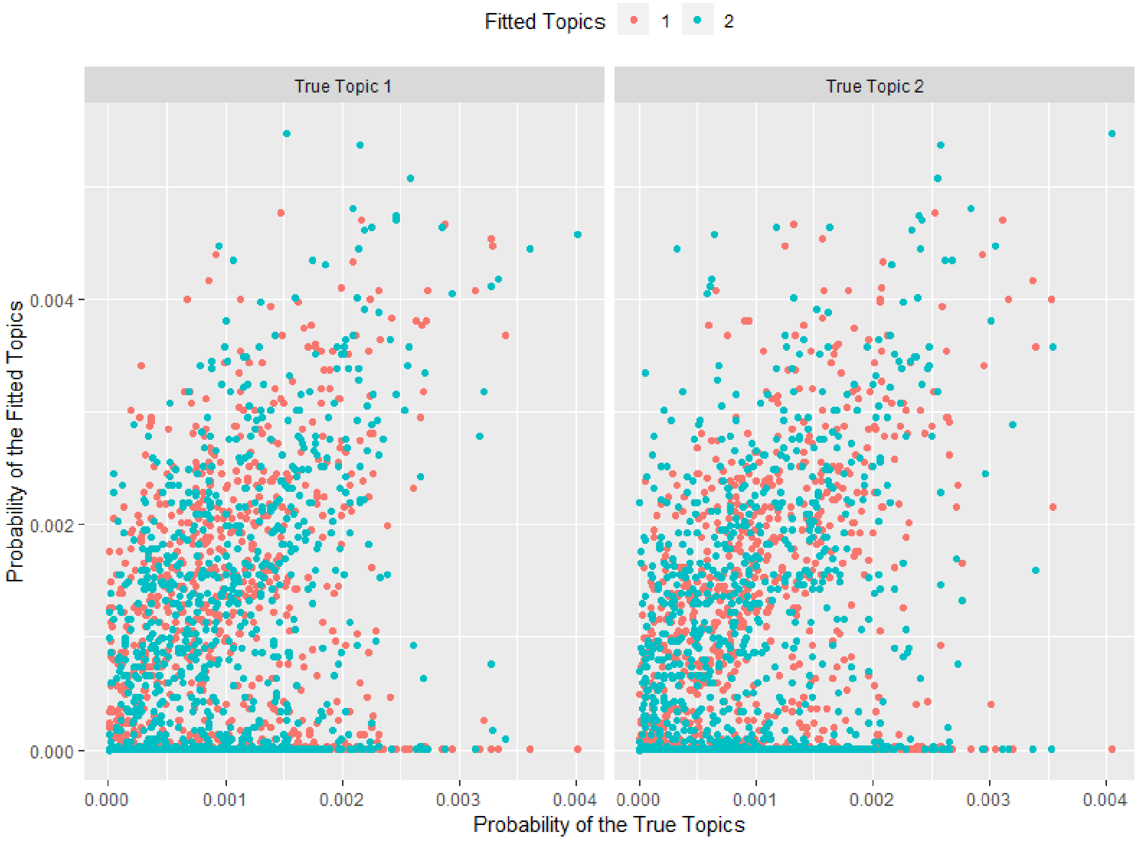

Figure 5.15: Correlation Structure of the True and Fitted Topic-word Distributions without the pre-processing step

	LDA Topic 1	LDA Topic 2
True Topic 1	0.2335280	0.3601863
True Topic 2	0.3202903	0.2802510

Table 5.16: Correlations between the True Topics and Fitted LDA Topics for the Corpus with Additional Document

for the new corpus. Table 5.16 shows the correlations. The mean correlation equals to 0.3402383 under this situation. Hence, we observed a 15.0% increase in the quality of the LDA results measured by correlation.

For the added document that is constructed from only one topic, the only characteristic one may change is the document length. Hence, we conducted an additional simulation to analyze the impact of the document length. We constructed documents from length 200 to 2000 increased by 10, and computed the correlations of the LDA results for the original corpus combined with each of the documents. Figure 5.16 shows the results. We observed that the document length does not have impact on the correlation. We fitted a linear regression model of $Correlation = \beta_0 + \beta_1$Document Length and the results are shown below.

```
Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept)        3.169e-01  2.383e-03 132.988   <2e-16 ***

‘Document Length‘ -9.409e-08  1.957e-06  -0.048    0.962
```

5.2 Future Work

There are three directions to further study the performance of LDA with different topic and document structures.

1. Add more factors to the simulation to allow combinations that are very similar to real world situations.
2. Study more detail of the interactions.
3. Create more advanced document structures that are again, very similar to real world situations.

Other characteristics discussed in Chapter 3 can also be examined through simulation studies. The number of true topics and the number of fitted topics both take on more levels to construct

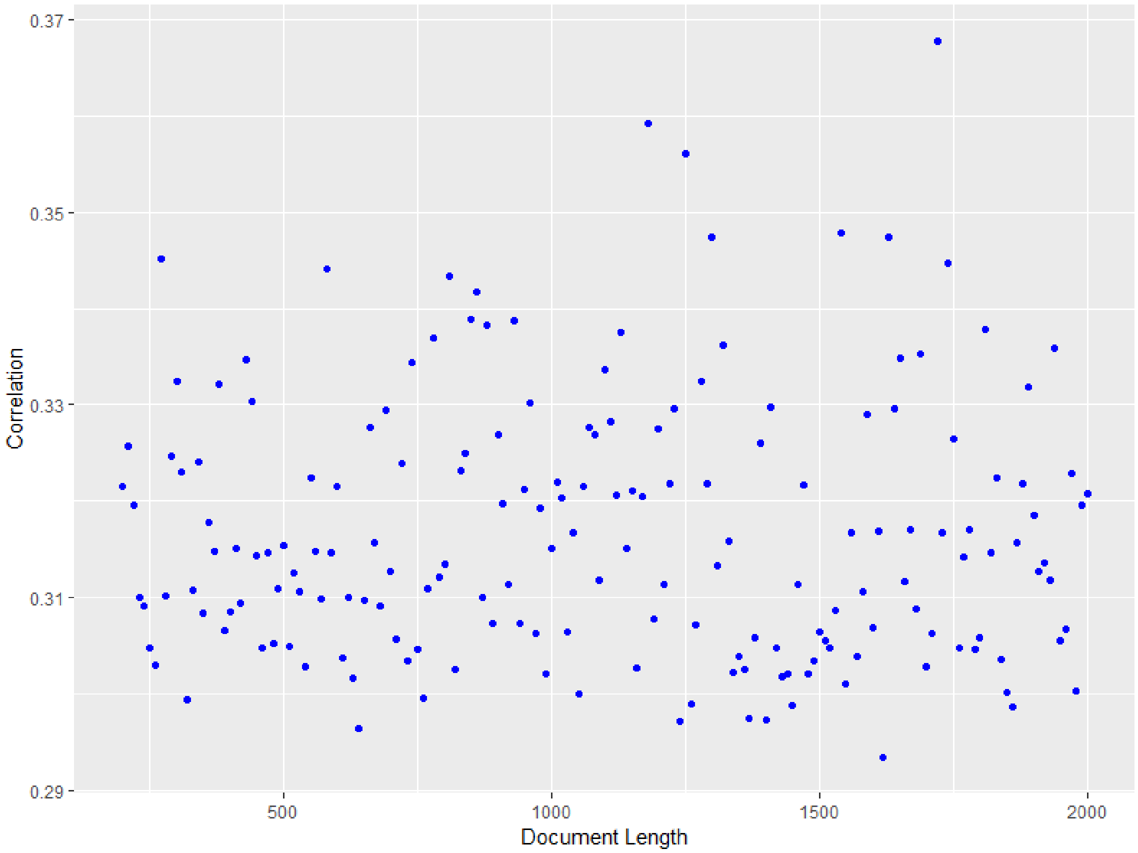

Figure 5.16: Correlation Results for Different Document Length of the Added Document

more complicated simulation configurations. The interactions between factors are the other possible direction. A more carefully designed simulations experiment should allow a more effective measure of the impact of the interactions. We may generate comparisons between the document length ratio and the vocabulary size. More advanced document structures is very helpful in learning the quality of LDA model results under real world applications. The document structure may vary even more when the number of true topics and the number of fitted topics are introduced into the simulation, the document structure may vary significantly.

In this chapter, we completed analysis on the simulation results from Chapter 4. We found that the type of the topic-word distributions significantly impact the quality of LDA model. Also, the document length ratio is positively related with the quality of the LDA model and the relationship is close to logarithmic and the size of vocabulary does not have a significant impact on the quality of the LDA model. The document structure represented by the document-topic distribution has impact on the quality of the LDA model in a complex way and there exist interactions between these factors that suggested a pre-processing step that increases the quality of the LDA results.

Appendices

Appendix A R code in this dissertation

We used the following packages:

```
library(topicmodels)

library(tidytext)

library(ggplot2)

library(tidyr)

library(dplyr)

library(stringr) #common string functions

library(tidyverse)

library(tm)

library(MCMCpack)

library(reshape2)
```

A.1 Checking Impact of the Size of Vocabulary

```
cbind(reu_voc,news_sov,ap_sov,imdb_sov)->sov

sov<-sov%>%as.tibble()%>%

dplyr::select(Reuters21578=reu_voc,Newsgroups=news_sov,AP=ap_sov,aclIMDB=imdb_sov)%>%

mutate(Documents=row_number()+4)


sov%>%filter(Documents<20|(Documents%%10==0 & Documents<100)|Documents%%100==0)%>%

dplyr::select(Documents,Reuters21578,Newsgroups,AP,aclIMDB)->sov


print(sov,n=40)


cbind(reu_voc,news_sov,ap_sov,imdb_sov)->sov_full


sov_full<-sov_full%>%as.tibble()%>%

dplyr::select(Reuters21578=reu_voc,Newsgroups=news_sov,AP=ap_sov,aclIMDB=imdb_sov)%>%

mutate(Documents=row_number()+4)
```

```
sov_full%>%melt(id=c("Documents"),varnames=c("Reuters21578","Newsgroups","AP","aclIMDB"),
value.name = "Vocabulary")%>%
mutate(Corpus=variable)->sov_full

sov_full%>%
ggplot(aes(x=Documents,y=Vocabulary,group=Corpus,color=Corpus))+
geom_path()+
theme(legend.position = "top")+
xlab("Number of Documents")+
ylab("Vocabulary Size")
```

A.2 Generate Corpus

The following functions are used to generate corpus.

```
## Function to generate corpus
simulateCorpus <- function(
M, # number of documents
docLengths, # vector of doc lengths
K,   # Number of Topics
Theta,  # Document*Topic distribution matrix M*K-Dimension
Phi  # Topic*term distribution matrix K*nTerms-Dimension
)
{
## Create corpus as a vector of strings
corp <- sapply(1:M,generateDoc,docLengths,Theta,Phi)

return(corp)
}
```

```
## generate observed document from Theta, Phi, and docLengths
generateDoc <- function(index,docLengths,Theta, Phi){
# docLengths is the length of documents in the corpus
# Theta is document-topic distribution for this document
# Phi is the topic-terms distribution matrix over all topics (term by topic)


#testing the above function
# topic_dist <- c(0.3,0.3,0.4)
# terms_topics_dist <- matrix(c(1,0,0,0,1,0,0,0,1),ncol=3)
# t<-generateWord(topic_dist,terms_topics_dist)
# t

doclengths <- docLengths[index] # specific doc length
topic_dist <- Theta[index,] # specific topic distribution for this doc
terms_topics_dist <- Phi
## create a vector of possible terms
doc_v<-sapply(1:doclengths, generateWord,topic_dist,terms_topics_dist)

## return a string format with " " as separator.
return(paste(doc_v,collapse = " "))
}

## Function to generate individual word from given Theta and Phi
generateWord <- function(index,topic_dist, terms_topics_dist){
# index is for lapply
# topic_dist is specific topic distribution for this document
# terms_topics_dist is the terms distribution matrix over all topics (term by topic)
## Choose topic for this word
topic <- rmultinom(1,1, topic_dist)
### topic is now a vector of 0s and 1. Pick the row index of 1
topic_n <- which(topic == 1)

### to test:print(topic_n)
## Choos a term from that topic’s term distribution
term <- rmultinom(1,1,terms_topics_dist[topic_n,])

### term is now a vector of 0s and 1. return the term index (integer)
return(which(term == 1))
}
```

A.3 Checking the impact of Document length ratio

```
burnin <- 4000
iter <- 2000
thin <- 500
set.seed(2345)
M=10
K=2
#constuct theta
Theta_10<-Construct_doc_topic(M, K, random=0, singleton=1, alpha=1/K, Alpha=FALSE)

Topics <- paste0("True_Topic_", seq(K))
Documents <- paste("Document", seq(M))
colnames(Theta_10) <- Topics
rownames(Theta_10) <- Documents

nTerms=100
docLengths_50=c(rep(50,M))
docLengths_100=c(rep(100,M))
Terms <- paste("Term",seq(nTerms))
```

```
corr_docLength("one_topic", min=1,max=10, nTerms=100, verbose = T,std=1
,Theta=Theta_10)->docl_sim

docl_nsim<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=100, verbose=T,std=1)



docl_sim%>%melt(value.name = "corr", varnames=c("Doclengths","True"))%>%
mutate(Doclengths=as.factor(round(rep(c(1/c(10:1),1:10),K),2)))%>%
ggplot(aes(factor(Doclengths), corr, group=True,color=factor(True)))+
geom_path()+
geom_point()+
ylim(0,1)+
xlab("Document Length Ratio")+
ylab("Correlation")+
labs(color='True Topics') +
theme(legend.position = c(0.8, 0.2))

docl_nsim%>%melt(value.name = "corr", varnames=c("Doclengths","True"))%>%
mutate(Doclengths=as.factor(round(rep(c(1/c(10:1),1:10),K),2)))%>%
ggplot(aes(factor(Doclengths), corr, group=True,color=factor(True)))+
geom_path()+
geom_point()+
ylim(0,1)+
xlab("Document Length Ratio")+
ylab("Correlation")+
labs(color='True Topics') +
theme(legend.position = c(0.8, 0.2))
```

A.4 Checking the Impact of the Document structure

```
# construct Thetas
```

```
burnin <- 4000

iter <- 2000

thin <- 500

set.seed(2345)

M=10

K=2

Topics <- paste0("True_Topic_", seq(K))

Documents <- paste("Document", seq(M))


Theta_twist_1<-rbind(matrix(0.5,ncol=2,nrow=2),matrix(c(1,0,0,1,1,0,0,1), ncol=2, byrow = T),

matrix(c(0.75,0.25,0.25,0.75, 0.75,0.25,0.25,0.75), ncol=2, byrow=T))

Theta_10<-Construct_doc_topic(M, K, random=0, singleton=1, alpha=1/K, Alpha=FALSE)

Theta_half<-matrix(0.5,ncol=2,nrow=10)

Theta_7525<-matrix(c(0.75,0.25,0.25,0.75),nrow=2, byrow=T)

[rep(1:nrow(matrix(c(0.75,0.25,0.25,0.75),nrow=2, byrow=T)),times=5),] Theta_twist_2<-

rbind(matrix(0.5,ncol=2,nrow=5),

matrix(c(1,0,0,1,1,0,0,1,1,0), ncol=2, byrow = T)

)

Theta_twist_3<-rbind(matrix(0.5,ncol=2,nrow=7),

matrix(c(1,0,0,1,1,0), ncol=2, byrow = T)

)

Theta_twist_4<-rbind(matrix(0.5,ncol=2,nrow=8),

matrix(c(1,0,0,1), ncol=2, byrow = T)

)

Theta_twist_5<-rbind(matrix(0.5,ncol=2,nrow=9),

matrix(c(1,0), ncol=2, byrow = T)

)

Theta_twist_6<-rbind(matrix(.5, ncol=2, nrow=9),

matrix(c(.75,.25), ncol=2, byrow = T))

Theta_twist_7<-rbind(matrix(c(.425,.575,.575,.425,

.425,.575,.575,.425,
.425,.575,.575,.425,
.425,.575,.575,.425,
.425,.575,.575,.425),
ncol=2)) Theta_twist_8<-
rbind(matrix(c(.45,.55,.55,.45,.45,.55,.55,.45,.45,.55,.55,.45,.45,.55,.55,.45,.45,.55,
.55,.45), ncol=2))
Theta_twist_8_1<-rbind(matrix(c(.475,.525,.525,.475,
.475,.525,.525,.475,
.475,.525,.525,.475,
.475,.525,.525,.475,
.475,.525,.525,.475),
ncol=2))

burnin <- 4000
iter <- 2000
thin <- 500
set.seed(2345)
M=10
K=3
Theta_twist_9<-Construct_doc_topic(M, K, random=0, singleton=1, alpha=1/K, Alpha=FALSE)
Theta_twist_10<-matrix(0.5,ncol=3,nrow=10)
Theta_twist_11<-rdirichlet(10,c(1,1,1))




t1nn<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_twist_1,
save.single=F,std=1)
t1dn<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_twist_1,
save.single=F,std=1)
t1nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_1)
t1dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_1)


t1nn1<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_10,
save.single=F,std=1)
t1dn1<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_10,
save.single=F,std=1)
t1nl1<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_10)
t1dl1<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_10)


t1nn2<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_half,
save.single=F,std=1)
t1dn2<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_half,
save.single=F,std=1)
t1nl2<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_half)
t1dl2<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_half)

t1nn3<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_7525,
save.single=F,std=1)
t1dn3<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_7525,
save.single=F,std=1)
t1nl3<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_7525)
t1dl3<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_7525)


t2nn<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_twist_2,
save.single=F,std=1)
t2dn<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_twist_2,
save.single=F,std=1)
t2nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_2)
t2dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_2)

t3nn<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_twist_3,
save.single=F,std=1)
t3dn<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_twist_3,
save.single=F,std=1)
t3nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_3)
t3dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_3)

t4nn<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_twist_4,
save.single=F,std=1)
t4dn<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_twist_4,
save.single=F,std=1)
t4nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_4)
t4dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_4)

t5nn<-corr_nTerms(functionname = "one_topic_norm", min=5,max=100,verbose=T, Theta=Theta_twist_5,
save.single=F,std=1)
t5dn<-corr_nTerms(functionname = "one_topic", min=5,max=100,verbose=T, Theta=Theta_twist_5,
save.single=F,std=1)
t5nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_5)
t5dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_5)



t6nn<-corr_nTerms(functionname = "one_topic_norm", min=6,max=100,verbose=T, Theta=Theta_twist_6,
save.single=F,std=1)
t6dn<-corr_nTerms(functionname = "one_topic", min=6,max=100,verbose=T, Theta=Theta_twist_6,
save.single=F,std=1)
t6nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_6)
t6dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_6)

t7nn<-corr_nTerms(functionname = "one_topic_norm", min=6,max=100,verbose=T, Theta=Theta_twist_7,
save.single=F,std=1)
t7dn<-corr_nTerms(functionname = "one_topic", min=6,max=100,verbose=T, Theta=Theta_twist_7,
save.single=F,std=1)
t7nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_7)
t7dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_7)



t8nn<-corr_nTerms(functionname = "one_topic_norm", min=6,max=100,verbose=T, Theta=Theta_twist_8,
save.single=F,std=1)
t8dn<-corr_nTerms(functionname = "one_topic", min=6,max=100,verbose=T, Theta=Theta_twist_8,
save.single=F,std=1)
t8nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_8)
t8dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_8)

t9nn<-corr_nTerms(functionname = "one_topic_norm", min=6,max=100,verbose=T, Theta=Theta_twist_8_1,
save.single=F,std=1)
t9dn<-corr_nTerms(functionname = "one_topic", min=6,max=100,verbose=T, Theta=Theta_twist_8_1,
save.single=F,std=1)
t9nl<-corr_docLength("one_topic_norm", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_8_1)
t9dl<-corr_docLength("one_topic", min=1,max=10, nTerms=1000, verbose=T,save.single=F,
std=1, Theta=Theta_twist_8_1)

list_of_corr_nn<-c("t1nn1","t1nn2","t1nn3","t1nn","t7nn",
"t8nn","t9nn","t2nn","t3nn","t4nn","t5nn","t6nn")

list_of_corr_dn<-c("t1dn1","t1dn2","t1dn3","t1dn","t7dn",
"t8dn","t9dn","t2dn","t3dn","t4dn","t5dn","t6dn")

list_of_corr_nl<-c("t1nl1","t1nl2","t1nl3","t1nl","t7nl",
"t8nl","t9nl","t2nl","t3nl","t4nl","t5nl","t6nl")

list_of_corr_dl<-c("t1dl1","t1dl2","t1dl3","t1dl","t7dl",
"t8dl","t9dl","t2dl","t3dl","t4dl","t5dl","t6dl")

list_of_names<-paste("Case ",seq(1,length(list_of_corr_nn),1))
key1<-cbind(list_of_corr_nn,list_of_names)%>%as.tibble()%>%set_names(c("From","Cases"))
key2<-cbind(list_of_corr_dn,list_of_names)%>%as.tibble()%>%set_names(c("From","Cases"))
```

```
key3<-cbind(list_of_corr_nl,list_of_names)%>%as.tibble()%>%set_names(c("From","Cases"))
key4<-cbind(list_of_corr_dl,list_of_names)%>%as.tibble()%>%set_names(c("From","Cases"))

single_wrap_n<-function(name){
return(name%>%get()%>%as.data.frame()%>%
gather(Topic,Correlation)%>%
mutate(Topic=recode(Topic,V1=1,V2=2))%>%
group_by(Topic)%>%
mutate(nTerms=row_number())%>%
ungroup()%>%
mutate(From=name))
}

single_wrap_l<-function(name){
return(name%>%get()%>%as.data.frame()%>%
gather(Topic,Correlation)%>%
mutate(Topic=recode(Topic,V1=1,V2=2))%>%
group_by(Topic)%>%
mutate(DocLength=c(1/c(10:1),c(1:10)))%>%
ungroup()%>%
mutate(From=name))
}
# single_wrap_l("t1nl")
# t1nn%>%as.data.frame()%>%
#   gather(Topic,Correlation)%>%
#   mutate(Topic=recode(Topic,V1=1,V2=2))%>%
#   group_by(Topic)%>%
#   mutate(nTerms=row_number())%>%
#   ungroup()

wrap_nn<-lapply(list_of_corr_nn,single_wrap_n)%>%bind_rows()
```

```
wrap_dn<-lapply(list_of_corr_dn,single_wrap_n)%>%bind_rows()
wrap_nl<-lapply(list_of_corr_nl,single_wrap_l)%>%bind_rows()
wrap_dl<-lapply(list_of_corr_dl,single_wrap_l)%>%bind_rows()

wrap_nn%>%bind_rows(wrap_dn)%>%
mutate(Dist=recode(substr(From,3,3),n="Normal",d="Dirichlet"))%>%
left_join(bind_rows(key1,key2))->data_n
wrap_nl%>%bind_rows(wrap_dl)%>%
mutate(Dist=recode(substr(From,3,3),n="Normal",d="Dirichlet"))%>%
left_join(bind_rows(key3,key4))->data_l

label_rev<-function(labels,multi_line=TRUE, sep=":"){
label_both(rev(labels),multi_line = multi_line, sep=sep)
}

fn = factor(list_of_names, levels=unique(list_of_names[order(1:12)]), ordered=TRUE)

data_n%>%
mutate(Cases2=factor(Cases, levels=unique(list_of_names[order(1:12)]),
ordered=TRUE))%>%
mutate(nTerms=nTerms+4)%>%
mutate(Correlation=ifelse(Correlation>.2,Correlation,.2))%>%
ggplot(aes(x=nTerms,y=Cases2,fill=Correlation))+
geom_tile()+
scale_fill_gradient(low="white", high="steelblue")+
facet_wrap(~Topic+Dist, strip.position = "top", labeller=label_rev, nrow=1)+
labs(x="Size of Vocabulary")

data_l%>%
mutate(Cases2=factor(Cases, levels=unique(list_of_names[order(1:12)]),
ordered=TRUE))%>%
mutate_at(3, funs(round(.,2)))%>%
ggplot(aes(x=factor(DocLength),y=Cases2,fill=Correlation))+
geom_tile()+
scale_fill_gradient(low="white", high="darkorange1")+
facet_wrap(~Topic+Dist, strip.position = "top", labeller=label_rev, nrow=1)+
labs(x="Document Length Ratio")+
theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

```
Theta_10%>%as_tibble()%>%
mutate(ind=row_number())%>%
melt(id.var="ind")%>%
mutate(variable=recode(variable,"True_Topic_1"='1',"True_Topic_2"='2'))%>%
mutate(ind=factor(ind))%>%
ggplot(aes(ind,variable, fill=as.factor(value)))+
geom_tile()+
scale_fill_grey(start=.8,end = .3)+
labs(x="Document",y="Topics",fill="Proportion")+
theme(panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.background = element_rect(fill="transparent",colour=NA),
plot.background = element_rect(fill="transparent",colour=NA)
)
```

A.5 Code for Pilot study

```
set.seed(2345)
M=10
K=2
nTerms=5
top_word_number=500  # useless
```

```
even=1  # useless
cluster=0  # useless
top_share=.8  # useless
docLengths<-c(rep(50,M))

# Gibbs parameters
burnin <- 4000
iter <- 2000
thin <- 500
seed <-list(2003,5,63,100001,765)
nstart <- 5
best <- TRUE

## construct Phi
Phi_5<-matrix(c(0.4,0.4,0.25,0.05,0.05,0.25,0.2,0.1,0.1,0.2), ncol=2, byrow = T)%>%
as_tibble()%>%t() Terms <- paste("Term",seq(nTerms))
Topics <- paste0("True_Topic_", seq(K))
Documents <- paste("Document", seq(M))
colnames(Phi_5) <- Terms
rownames(Phi_5) <- Topics

#see the true topic graph
t(Phi_5)%>%as_tibble()%>%
ggplot(aes(1:length(True_Topic_1),True_Topic_1))+
geom_bar(stat = "identity")

rowSums(Phi_5)

#constuct theta
Theta_5<-Construct_doc_topic(M, K, random=0, singleton=1, alpha=1/K, Alpha=FALSE)
colnames(Theta_5) <- Topics
```

```
rownames(Theta_5) <- Documents

# Create corpus
corpus_5<-simulateCorpus(M,docLengths,K,Theta_5,Phi_5)
corpus_5[1]

# Generate dtm
corpus_5_dtm <-Doc2DTM(corpus_5)

# Fit LDA with Gibbs
# Number of topics to be fitted (k) and hyperparameter (alpha) are
# the only stuff that can be changed. People usually only change k.
# I still don’t know how to adjust beta.
LDAresult_5<-LDA(corpus_5_dtm,k=K, method="Gibbs", control=list(
nstart=1, seed = 1234, best=best, burnin = burnin, iter = iter, thin=thin, verbose=0))

# Generate the comparision table
rank_table_5<-TruethCompare(LDAresult_5,Phi_5, Theta_5)

# Generate graph. Number ’5’ can be customized
Graphics_LDA_TopicTerm(rank_table_5,5)

# Pecentage graph and table. Number ’10’ can be customized
True_percentage(rank_table_5,3,graph=TRUE)

# find all true_percentage based on rank
trend_5<-rank2trend(rank_table_5)

# Generate graph
trend2graph(trend_5)
```

```
# fetching maximum for each color
rankmax(trend_5)
rankmax(trend_5[which(trend_5$level<=4),])
trend_5[which(trend_5$True_topic=="True 1"),]

# get beta
LDAresult_5%>%tidy(matrix="beta")%>%arrange(topic)

# get gamma
LDAresult_5%>%tidy(matrix="gamma")%>%dcast(document~topic)

# graphics
pilot_names<-list('1'="Fitted Topic 1",
'2'="Fitted Topic 2"
)
pilot_labeller<-function(variable,value){
return(pilot_names[value])
}
LDAresult_5%>%tidy(matrix="beta")%>%arrange(topic)%>%
ggplot(aes(factor(term),beta))+
geom_col()+
facet_wrap(~topic,scales = "free_y", labeller=pilot_labeller,ncol=1)+
labs(x="Terms",y="Probability")

LDAresult_5%>%tidy(matrix="gamma")%>%
mutate(document=factor(document, levels = c(1,2,3,4,5,6,7,8,9,10)))%>%
ggplot(aes(document,factor(topic), fill=gamma))+
geom_tile()+
scale_fill_gradient(low="#CCCCCC",high = "#666666")+
labs(x="Document",y="Topics",fill="Proportion")+
theme(panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.background = element_rect(fill="transparent",colour=NA),
plot.background = element_rect(fill="transparent",colour=NA)
)

#change seed
LDAresult_5_1<-LDA(corpus_5_dtm,k=K, method="Gibbs", control=list(
nstart=1, seed = 12345, best=best, burnin = burnin, iter = iter, thin=thin, verbose=0))
# get beta
LDAresult_5_1%>%tidy(matrix="beta")%>%arrange(topic)

# get gamma
LDAresult_5_1%>%tidy(matrix="gamma")%>%dcast(document~topic)

# prediction job
corpus_5_1<-simulateCorpus(M,docLengths,K,Theta_5,Phi_5)
corpus_5_1_dtm <-Doc2DTM(corpus_5_1)
perplexity(LDAresult_5,corpus_5_1_dtm)
```

A.6 Real Corpus Analysis

```
####################################################
##
##  Reuters21578
##
####################################################

library(tm.corpus.Reuters21578)
data(Reuters21578)
data(Reuters21578_DTM)
Reuters21578_DTM
tidy(Reuters21578_DTM)->reu
```

```
reu%>%distinct(document)%>%
summarise(n())
# number of doc 19042

reu%>%distinct(term)%>%
summarise(n())
# size of vocabulary 33255

reu%>%group_by(document)%>%
summarise(wordcounts=sum(count))%>%
summarise(avg_wc=mean(wordcounts))

# avg_wc=79.8

reu%>%summarise(sum(count))
# number of words 1520283

re_ind<-reu%>%distinct(document)%>%
mutate(id=row_number())

reu_id<-reu%>%left_join(re_ind)

doc_ratio<-function(ind=1,C=reu_id,N=19042){
#sample 10 from N
s=sample.int(N,10)
C%>%filter(id %in% s)->d
voc<-d%>%distinct(term)%>%summarise(voc=n())%>%pull(voc)
avg<-d%>%group_by(id)%>%summarise(n=sum(count))%>%
summarise(avg=mean(n))%>%pull(avg)
return(avg/voc)
}

voca_n<-function(ind=1,C=reu_id,N=19042,n=10){
s=sample.int(N,n)
C%>%filter(id %in% s)->d
voc<-d%>%distinct(term)%>%summarise(voc=n())%>%pull(voc)
return(voc)
}

voca_rep<-function(n=10,C=reu_id,N=19042){
return(mean(sapply(1:100, voca_n, C=C, N=N, n=n)))
}

doc_ratio()

set.seed(2345)
sapply(1:10000, doc_ratio)->reu_docl

reu_docl%>%tibble::enframe(name=NULL)%>%
ggplot(aes(value,fill=1))+
geom_density(alpha=.2)+
xlab("Document Length Ratio")+
ylab("Density")+ theme(legend.position="none")

mean(reu_docl)
# mean 1918609

summary(reu_docl)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.1418  0.1797  0.1909  0.1919  0.2024  0.3194
```

```
set.seed(2345)
sapply(5:1000, voca_rep)->reu_voc

reu_voc%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(n,value))+
geom_line()+
xlab("Number of Documents")+
ylab("Vocabulary Size")

reu_voc%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(sqrt(n),value))+
geom_line()+
xlab("Square root of Number of Documents")+
ylab("Vocabulary Size")+
geom_abline(slope = 242.01432, intercept = -373.21016,size=1,color="blue")

reu_voc%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
mutate(value_sq=valueˆ2)%>%
mutate(n_sqrt=sqrt(n))->tar_reu

fit<-lm(value˜n_sqrt,tar_reu)

fit$fitted.values
summary(fit)
plot(fit)

ggplot(fit,aes(fit$fitted.values,fit$residuals))+
geom_point()+
geom_abline(slope = 0,intercept = 0, color="red")+

xlab("Fitted Values")+

ylab("Residuals")


predict(fit, n_sqrt=10)

###################################################

##

##   20 News

##

###################################################

News20 <- read.csv("http://ssc.wisc.edu/~ahanna/20_newsgroups.csv", stringsAsFactors = FALSE)

News20<-News20[-1]


## make tbl_df for nicer behavior on output

News20 <- tbl_df(News20)


## peak at the data

names(News20)

head(News20$text, 2)

nrow(News20)


sample_n(News20,size=10)

unique(News20$target)


News20%>%

filter(cumsum(text=="")>0, cumsum(str_detect(text, "^--"))==0)


train_folder<-"20_newsgroup/"


read_folder <- function(infolder) {

tibble(file = dir(infolder, full.names = TRUE)) %>%
mutate(text = map(file, read_lines)) %>%
transmute(id = basename(file), text) %>%
unnest(text)
}

raw_text <- tibble(folder = dir(train_folder, full.names = TRUE)) %>%
unnest(map(folder, read_folder)) %>%
transmute(newsgroup = basename(folder), id, text)

glimpse(raw_text)

cleaned_text <- raw_text %>%
group_by(newsgroup, id) %>%
filter(cumsum(text == "") > 0,
cumsum(str_detect(text, "^--")) == 0) %>%
ungroup()

cleaned_text <- cleaned_text %>%
filter(str_detect(text, "^[^>]+[A-Za-z\\d]") | text == "",
!str_detect(text, "writes(:|\\.\\.\\.)$"),
!str_detect(text, "^In article <"),
!id %in% c(9704, 9985))

usenet_words <- cleaned_text %>%
unnest_tokens(word, text) %>%
filter(str_detect(word, "[a-z’]$"),
!word %in% stop_words$word)

usenet_words %>%
count(word, sort = TRUE)
```

```
head(usenet_words)

usenet_words%>%group_by(newsgroup,id)%>%
mutate(len=n())%>%
ungroup()%>%
mutate(index=paste(newsgroup,id))%>%
distinct(index, .keep_all = T)->twenty_news

twenty_news%>%summarise(n=n())

# number of doc 19791

twenty_news%>%summarise(avg_wc=mean(len))

# avg number of words per doc 62.5

twenty_news%>%summarise(voc=sum(len))

# number of words 1237217

usenet_words%>%distinct(word)%>%
summarise(voc=n())

# size of vocabulary 96509

news_tidy<-usenet_words%>%
mutate(term=word)%>%
mutate(index=paste(newsgroup,id))%>%
left_join(
twenty_news%>%mutate(id=row_number()),
by="index"
)%>%
dplyr::select(id=id.y,term)%>%
group_by(id)%>%
count(term)%>%
ungroup()%>%
mutate(count=n)

set.seed(2345)
sapply(1:10000, doc_ratio,news_tidy,19791)->news_docl

news_docl%>%tibble::enframe(name=NULL)%>%
ggplot(aes(value,fill=1))+
geom_density(alpha=.2)+
xlab("Document Length Ratio")+
ylab("Density")+ theme(legend.position="none")



mean(news_docl)
# mean 0.1345293

summary(news_docl)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.1045  0.1231  0.1297  0.1345  0.1392  0.4384

set.seed(2345)
sapply(5:1000, voca_rep,news_tidy,19791)->news_sov

news_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(n,value))+
geom_line()+
xlab("Number of Documents")+
ylab("Vocabulary Size")

news_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
mutate(x=sqrt(n)+nˆ(1/3))%>%
ggplot(aes(x,value))+
geom_line()+
xlab("Square root of Number of Documents")+
ylab("Vocabulary Size")
#geom_abline(slope = 613.183, intercept = -2746.087,size=2,color="blue")

news_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
mutate(value_sq=valueˆ2)%>%
mutate(n_sqrt=sqrt(n))%>%
mutate(n_log=log(n))%>%
mutate(n_cubic=nˆ(1/3))->tar_news

fit<-lm(value˜n_sqrt+n_cubic,tar_news)
fit1<-nlme::gls(value˜n_sqrt+n_cubic,data=tar_news)

fit$fitted.values
summary(fit)
plot(fit1)

ggplot(fit,aes(fit$fitted,fit$residuals))+
geom_point()+
geom_abline(slope = 0,intercept = 0, color="red")+
xlab("Fitted Values")+
ylab("Residuals")

ggplot(tar_news, aes(n,value))+
geom_point(size=1, alpha=.2)+
geom_smooth(method=lm, formula = y ~ sqrt(x)+I(x**(1/3)))+
xlab("Number of Documents")+
ylab("Vocabulary Size")

predict(fit, data.frame(n_sqrt=sqrt(10),n_cubic=10**(1/3)))

#############################################
##
##  AP
##
#############################################

data("AssociatedPress",package = "topicmodels")
ap_td <- tidy(AssociatedPress)
ap_td%>%
group_by(document)%>%
summarise(doclen=sum(count))%>%
summarise(avg_wc=mean(doclen))

ap_td%>%distinct(term)%>%
summarise(n=n())
# size of vocabulary 10473

ap_td%>%distinct(document)%>%
summarise(n=n())
# number of documents 2246
```

```
ap_td%>%group_by(document)%>%
summarise(voca=sum(count))%>%
summarise(avg_len=mean(voca))
# avg number of words per doc 194

ap_td%>%summarise(n=sum(count))
# number of words 435838

ap_td%>%left_join(
ap_td%>%distinct(document)%>%
mutate(id=row_number())
)->ap_tidy

set.seed(2345)
sapply(1:10000, doc_ratio,ap_tidy,2246)->ap_docl

ap_docl%>%tibble::enframe(name=NULL)%>%
ggplot(aes(value,fill=1))+
geom_density(alpha=.2)+
xlab("Document Length Ratio")+
ylab("Density")+ theme(legend.position="none")

mean(ap_docl)
# mean 0.1775743

summary(ap_docl)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.1421  0.1710  0.1774  0.1776  0.1840  0.2186

set.seed(2345)
sapply(5:1000, voca_rep,news_tidy,2246)->ap_sov
```

```
ap_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(n,value))+
geom_line()+
xlab("Number of Documents")+
ylab("Vocabulary Size")

ap_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(sqrt(n),value))+
geom_line()+
xlab("Square root of Number of Documents")+
ylab("Vocabulary Size")+
geom_abline(slope = 606.95, intercept = -3059.71,size=2,color="blue")

ap_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
mutate(value_sq=value^2)%>%
mutate(n_sqrt=sqrt(n))%>%
mutate(n_cubic=n^(1/3))->tar_ap

fit<-lm(value~n_sqrt+n_cubic,tar_ap)

summary(fit)

ggplot(fit,aes(fit$fitted.values,fit$residuals))+
geom_point()+
geom_abline(slope = 0,intercept = 0, color="red")+
xlab("Fitted Values")+
ylab("Residuals")
```

```
ggplot(tar_ap, aes(n,value))+
geom_point(size=1, alpha=.2)+
geom_smooth(method=lm, formula = y ~ sqrt(x)+I(x**(1/3)))+
xlab("Number of Documents")+
ylab("Vocabulary Size")

predict(fit, data.frame(n_sqrt=sqrt(10),n_cubic=10**(1/3)))

############################################
##
##  IMDB
##
############################################

train_folder<-"imdb/"

read_folder <- function(infolder) {
tibble(file = dir(infolder, full.names = TRUE)) %>%
mutate(text = map(file, read_lines)) %>%
transmute(id = basename(file), text) %>%
unnest(text)
}

raw_text <- tibble(folder = dir(train_folder, full.names = TRUE)) %>%
unnest(map(folder, read_folder)) %>%
transmute(newsgroup = basename(folder), id, text)
folder<-tibble(folder = dir(train_folder, full.names = TRUE))
read_folder(train_folder)->raw_text
raw_text%>%unnest_tokens(word,text)%>%
group_by(id)%>%
count(word)->word_count

word_count%>%distinct(id)%>%ungroup()%>%
summarise(n=n())
# number of doc 100000

word_count%>%ungroup()%>%distinct(word)%>%
summarise(voca=n())

# size of vocabulary 171770

word_count%>%ungroup()%>%
summarise(voca=sum(n))

# number of words 23645581

word_count%>%
summarise(voca=sum(n))%>%
summarise(avg_len=mean(voca))

# avg number of words per doc 236

head(word_count)

word_count%>%
ungroup()%>%
mutate(term=word,count=n)%>%
left_join(
word_count%>%
ungroup()%>%
distinct(id)%>%
mutate(index=row_number())
)%>%

dplyr::select(id=index,term,count)->imdb_tidy



imdb_tidy%>%distinct(id)%>%
summarise(n())

set.seed(2345)
sapply(1:10000, doc_ratio,imdb_tidy,100000)->imdb_docl

imdb_docl%>%tibble::enframe(name=NULL)%>%
ggplot(aes(value,fill=1))+
geom_density(alpha=.2)+
xlab("Document Length Ratio")+
ylab("Density")+ theme(legend.position="none")

mean(imdb_docl)
# mean 0.2528772

summary(imdb_docl)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#0.2015  0.2398  0.2517  0.2529  0.2650  0.3590

set.seed(2345)
sapply(5:1000, voca_rep,imdb_tidy,100000)->imdb_sov

imdb_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(n,value))+
geom_line()+
xlab("Number of Documents")+
ylab("Vocabulary Size")

imdb_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
ggplot(aes(sqrt(n),value))+
geom_line()+
xlab("Square root of Number of Documents")+
ylab("Vocabulary Size")+
geom_abline(slope = 653.4839, intercept = -1671.4089,size=1,color="blue")

imdb_sov%>%as.tibble()%>%
mutate(n=row_number()+4)%>%
mutate(value_sq=value^2)%>%
mutate(n_sqrt=sqrt(n))%>%
mutate(n_cubic=n**(1/3))->tar_imdb

fit<-lm(value~n_sqrt+n_cubic+n,tar_imdb)

summary(fit)

ggplot(fit,aes(fit$fitted,fit$residuals))+
geom_point()+
geom_abline(slope = 0,intercept = 0, color="red")+
xlab("Fitted Values")+
ylab("Residuals")

ggplot(tar_imdb, aes(n,value))+
geom_point(size=1, alpha=.2)+
geom_smooth(method=lm, formula = y ~ sqrt(x)+I(x**(1/3))+I(x))+
xlab("Number of Documents")+
ylab("Vocabulary Size")


predict(fit, data.frame(n_sqrt=sqrt(10),n_cubic=10**(1/3),n=10))
```

Bibliography

[1] Mark C Baker. *The atoms of language: The mind's hidden rules of grammar.* Basic books, 2008.

[2] Bo-Christer Bjork, Annikki Roos, and Mari Lauri. Scientific journal publishing: yearly volume and open access availability. *Information Research: An International Electronic Journal*, 14(1), 2009.

[3] David M Blei and John D Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Norman E Breslow. Generalized linear models: checking assumptions and strengthening conclusions. *Statistica Applicata*, 8(1):23–41, 1996.

[6] Stefan Büttcher, Charles LA Clarke, and Gordon V Cormack. *Information retrieval: Implementing and evaluating search engines.* Mit Press, 2016.

[7] Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Nips*, volume 31, pages 1–9, 2009.

[8] Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. Evaluation metrics for language models. 1998.

[9] Noam Chomsky. Lectures on government and binding, foris, dordrecht. *ChomskyLectures on Government and Binding1981*, 1981.

[10] Shorter Oxford English Dictionary. Shorter oxford english dictionary, 2007.

[11] empty. 20 newsgroups dataset, empty.

[12] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. 2002.

[13] Chris Gropp, Alexander Herzog, Ilya Safro, Paul W Wilson, and Amy W Apon. Scalable dynamic topic modeling with clustered latent dirichlet allocation (clda). *arXiv preprint arXiv:1610.07703*, 2016.

[14] Bettina Grün and Kurt Hornik. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011.

[15] Gregor Heinrich. Parameter estimation for text analysis. *University of Leipzig, Tech. Rep*, 2008.

[16] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[17] Kriste Krstovski. Efficient inference, search and evaluation for latent variable models of text with applications to information retrieval and machine translation. 2016.

[18] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004.

[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[20] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499. ACM, 2007.

[21] David Mimno, Hanna M Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 880–889. Association for Computational Linguistics, 2009.

[22] Thomas Minka. Estimating a dirichlet distribution, 2000.

[23] Calvin S. Mooers. Editor's corner: "coding, information retrieval, and the rapid selector". *American Documentation*, 1(4):225, Oct 01 1950. Last updated - 2013-02-24.

[24] James W. Perry, Allen Kent, and Madeline M. Berry. Machine literature searching x. machine language; factors underlying its design and development. *American Documentation*, 6(4):242–254, 1955.

[25] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[26] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[27] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[28] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[29] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112. ACM, 2009.

[30] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. ACM, 2010.

[31] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[32] Wikipedia contributors. Stemming — Wikipedia, the free encyclopedia, 2019. [Online; accessed 13-June-2019].

[33] George Kingsley Zipf. The psychology of language. In *Encyclopedia of psychology*, pages 332–341. Philosophical Library, 1946.