A DOCUMENT EXPLORING SYSTEM ON LDA TOPIC MODEL FOR WIKIPEDIA ARTICLES

by

Zhou Tong

Thesis
submitted in partial fulfillment of the requirements for
the Degree of Master of Science (Computer Science)

Acadia University
Fall Convocation 2016

© by Zhou Tong, 2016
This thesis by Zhou Tong was defended successfully in an oral examination on June 28th, 2016.

The examining committee for the thesis was:

______________________________________________
Dr. Paul Stephenson, Chair

______________________________________________
Dr. Pawan Lingras, External Reader

______________________________________________
Dr. André Trudel, Internal Reader

______________________________________________
Dr. Haiyi Zhang, Supervisor

______________________________________________
Dr. Darcy Benoit, Director

This thesis is accepted in its present form by the Division of Research and Graduate Studies as satisfying the thesis requirements for the degree Master of Science (Computer Science).
I, Zhou Tong hereby grant permission to the University Librarian at Acadia University to provide copies of the thesis, upon request, on a non-profit basis.

______________________________  
Author

______________________________  
Supervisor

______________________________  
Date
# Contents

Abstract xiii

Acknowledgments xv

1 Introduction 1
   1.1 Motivation and Thesis Scope ............................. 2
   1.2 Thesis Structure ....................................... 3

2 Background 5
   2.1 Machine Learning ........................................ 5
      2.1.1 Supervised Learning ................................. 7
      2.1.2 Unsupervised Learning ............................... 8
   2.2 Information Retrieval .................................... 12
      2.2.1 Information Retrieval Tasks .......................... 12
      2.2.2 Information Retrieval Models ........................ 13
         Vector Space Model ..................................... 14
         Probabilistic Model .................................... 15
         Inference Network Model ................................. 16
   2.3 Text Mining ............................................... 16
      2.3.1 The Landscape of Text Mining ........................ 18
      2.3.2 Text Mining Tasks and Algorithms ................... 20
   2.4 Topic Models ............................................. 21
      2.4.1 Bag of Words Assumption ............................. 21
      2.4.2 Latent Dirichlet Allocation .......................... 23
4.3 Experiment 3: Exploring Corpus ......................................... 61
  4.3.1 Objective ................................................................. 61
  4.3.2 Method ................................................................. 61
  4.3.3 Results ................................................................. 63
  4.3.4 Summary ............................................................... 67

5 Conclusions and Future Work ........................................... 69
  5.1 Conclusions ............................................................. 69
  5.2 Limitations ............................................................. 70
  5.3 Future Work ............................................................. 70

Bibliography ................................................................. 73
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Real Estate Prices and the Sizes of Houses in Wolfville</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>A Sample of Men and Women’s Height and Weight</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>A Sample of <em>iris</em> data frame</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>The Cluster Means of 3 Clusters after K-means Process</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>The Comparison between the Actual Species and Clusters</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Document-Term Matrix of the Example Collection of Documents</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>EC2 Server Configuration</td>
<td>34</td>
</tr>
<tr>
<td>3.2</td>
<td>Model Log-likelihood per Number of Topics</td>
<td>42</td>
</tr>
<tr>
<td>3.3</td>
<td>7 of 50 Topics with Top 10 Terms</td>
<td>44</td>
</tr>
<tr>
<td>3.4</td>
<td>Top 3 Topics for Document Biology</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>A Sample of Documents Topic Clustering Results</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Top 30 Terms in Topic 3</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Clustering with Two Options</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Clustering Proportions of Two Options</td>
<td>54</td>
</tr>
<tr>
<td>4.5</td>
<td>Top 8 Documents with Closest JS Distance to Document Constitution</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>Top 8 Documents Titles to Document Constitution</td>
<td>59</td>
</tr>
<tr>
<td>4.7</td>
<td>Questionnaire Results of Document Constitution</td>
<td>59</td>
</tr>
<tr>
<td>4.8</td>
<td>The Scores of Questionnaire Documents</td>
<td>60</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 Applying Linear and Quadratic Regression on House Price Problem . 8
2.2 Men and Women’s Height and Weight Scatter Plot . . . . . . . . . . 9
2.3 Two K-means Clusters Plots by Sepal and Petal . . . . . . . . . . . 11
2.4 Text Mining Components [1] . . . . . . . . . . . . . . . . . . . . . . . 18
2.5 Text Mining Process [27] . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.6 An Example of LDA Topic Model [6] . . . . . . . . . . . . . . . . . . 24
2.7 A Graphic Model Representation of LDA [6] . . . . . . . . . . . . . . 26

3.1 The Workflow of the Experiment . . . . . . . . . . . . . . . . . . . . 33
3.2 The Order of the Pre-processing Steps . . . . . . . . . . . . . . . . 41
3.3 Log-likelihood per Number of Topics . . . . . . . . . . . . . . . . . 43
3.4 Document Biology and Topic Distribution Barplot . . . . . . . . . . . 45

4.1 Histogram of Documents Topic Clustering Result . . . . . . . . . . . 52
4.2 Questionnaire Structure . . . . . . . . . . . . . . . . . . . . . . . . . 57
4.3 Barplot of the Topic Distribution of Document Constituion . . . . . 58
4.4 System Structure of the Documents Exploring System . . . . . . . . 62
4.5 3 Layers of System Webpage Design . . . . . . . . . . . . . . . . . . 63
4.6 Screenshot of System Index Webpage (1) . . . . . . . . . . . . . . . 64
4.7 Screenshot of System Index Webpage (2) . . . . . . . . . . . . . . . 65
4.8 Screenshot of the Webpage of Topic 33 . . . . . . . . . . . . . . . . . 66
4.9 Screenshot of the Webpage of Document Mars . . . . . . . . . . . . 67
4.10 Click the Title Mars linked to the Simple Wikipedia Page of Mars . 68
Abstract

Organizing and exploring millions of documents, papers and other text information becomes a challenge for researchers and publishers. As machine learning techniques are quickly developed and widely used, a new text mining method called topic model was proposed in 2003. The topic model is based on Latent Dirichlet allocation (LDA) and has drawn much attention since it was introduced. LDA topic model is a probabilistic model, which can process text documents and exhibit hidden topics. Compared to other document processing methods working on content directly, the LDA topic model processes documents to topic distributions. The results are easier to understand, categorize and compare. Most importantly, topics make more sense to humans than structured machine formats.

In the thesis, we briefly introduce the background knowledge of LDA topic model and its working principles. Then we deeply explain how to apply LDA topic model to a text corpus by doing experiments on Simple Wikipedia documents. The experiments include all necessary steps of data retrieving, pre-processing, fitting the model and evaluations. The result of the experiments shows the LDA topic model working effectively on document clustering and finding similar documents. Meanwhile, based on LDA topic model, we propose a document exploring system which allows users to organize and explore the documents by topic where related documents are easier to find and access.
Acknowledgments

First and foremost, I would like to thank my parents Mr. and Mrs. Guoguang Tong and Yanxin Sun who financially supported my master’s studies in Canada. Thank you for your faith and unconditional love.

Then I would offer my sincere gratitude to my supervisor Dr. Haiyi Zhang, who provided me all necessary resources and his profound knowledge. Without your support and patience, this thesis may never be finished.

I also want to thank Ms. Erma Stultz and her reading club Jackets. They helped me finish the questionnaire, which is used as human judgement compared with LDA topic model results. Erma also did the proofreading for the thesis.

My dear and beloved friend Xin Liu, who gave me support and motivated me when the thesis was stuck at one point.

Last, but certainly not the least, I am thankful to all the staff of Jodrey School of Computer Science and Acadia University. They provided me with a perfect environment to study and excellent professors to answer my questions.
Chapter 1

Introduction

We say that we are living in the “Information Age”, which is defined as a period in human history characterized by the shift from traditional industry to an economy based on information computerization [11]. Large collections of data, such as news, books, mail, and other content is stored online that enables email services, cellphones, online shopping, streaming music or movies and many other new services that come into the new era. We benefit from the fast, convenient information service every day. However, the information has exploded astronomically and it is hard for humans to handle and understand easily. A new study shows that in 1986 we received around 40 newspapers full of information every day but this had rocketed to 174 in 2007. In 1986 we sent out — mainly by post, telephone and fax — around two and a half pages of newspaper each day. But it had increased to six newspapers thanks to email, digital photography, Twitter and social network sites by 2007 and it is still growing [3]. We are suffering from information overload, which is one of the big problems we have to face in the “Information Age”.

Information retrieval (IR) is to obtain information resources relevant to an information need from a collection of information resources. Searching and archiving information can be traced back to around 3000 BC, when the Sumerians designated special areas to store clay tablets with cuneiform inscriptions [35]. With the advent of computers and the information overload issue, the field of Information Retrieval was born and fast developed. IR systems are used on an everyday basis in order to
reduce information overload.

As researchers or university students, we are looking through tons of papers and magazines to read before we actually start to do research. Facing these mountains of articles, with thousands of new ones published each year, it is impossible for humans to read through. Obviously, for most documents we only want to know what are these documents talking about, and plenty of them might be irrelevant to our studies. What are their topics? Is there a way to efficiently and quickly organize, manage and explore their contents?

A new approach to extract information called topic models was proposed more than a decade ago. As the name suggests, it is a text model based on topics enabling us to feed in a collection of text data and assign the data into topics, and manage the documents by topics which are more meaningful to humans. Topic models can be a useful tool for the statistical analysis of document collections and other discrete data [7]. It is an unsupervised learning model that can process the text data without any labels required. Its results can be used for browsing the documents by topic, managing the documents by content and categorising the documents by their similarities. In the last couple of years, it has been constructed in different models to explore its functioning and performance. It also has been applied to many other areas like image processing and natural language processing.

In this thesis, we focus on Latent Dirichlet allocation (LDA) on topic models applied with a large collection of documents. The goal is to explore and understand the LDA topic model and apply the model to a practical dataset to explore its functioning including documents clustering and finding similar documents. Meanwhile, we proposed a documents exploring system assembling all those features. In the end, we conclude a better way of understanding and applying the model and some thoughts on the direction of future development in topic models as well.

1.1 Motivation and Thesis Scope

The LDA topic model was first proposed by Dr. David Blei in 2003, which is still relatively new. It has drawn much attention in the past couple of years and offered
many directions to explore. Topic models have previously been used for a variety of applications, including geographical information retrieval and the analysis of the development of ideas over time in the field of computational linguistics.

The LDA topic model has an impressive result on text data. In the thesis, one of the goals is to apply the model to a new document collection and build up the tools to explore and search on the documents. Furthermore, there is no paper or thesis that fully introduces the steps of applying LDA topic model to a dataset and also evaluates the results with human judgement. I decided to fill the void of this area and apply the LDA topic model to the Simple Wikipedia dataset following with a set of evaluation and comparison experiments. Wikipedia is a free access encyclopaedia, which can be treated as our human culture knowledge base. It contains millions of entries from thousands of areas, covering almost every corner of our life. Studying such a knowledge base will be a good start and a useful reference for future artificial intelligence robots or virtual assistants studies. Topic models not only contributes in computer science, it is also related to the area of math, statistics and history. The conclusions in this thesis could be a useful tool in social science and business. I hope my thesis and research studies will be a minor paving stone for future students and researchers.

In the experiments, I use the framework of the statistical programming language R. R is a open-source programming language and software environment for statistical computing. R language is widely used among statisticians and data miners for developing statistical software and data analysis. The R packages `topicmodels` and `tm` are the main tool that I use in the thesis research. They are open-sourced and implement the topic model and also many other useful text mining techniques.

### 1.2 Thesis Structure

This thesis is divided into five chapters, and their contents are shown as follows:

- Chapter 1: Introduction. It mainly talks about the introduction of the thesis, the motivation of the thesis and its scope.
• Chapter 2: Background. It deeply explains the background knowledge for this topic including machine learning, information retrieval, text mining, topic models and Jensen-Shannon (JS) divergence.

• Chapter 3: Experiment Design. It illustrates the experimental steps in this thesis including the data retrieving and precessing, fitting model and evaluation methods.

• Chapter 4: Experiments. It shows the experiments on LDA topic model with documents clustering, finding similar documents and exploring the corpus.

• Chapter 5: Conclusions and Future Work. It concludes the thesis and discusses the limitations and future directions of this topic.
Chapter 2

Background

In this chapter, I give a short introduction to some computer science techniques or concepts associated with this thesis, including information retrieval, machine learning, text mining, topic models and Jensen-Shannon divergence. Understanding these concepts is fundamental when reading the thesis. The definitions and some conclusions in this chapter are largely referenced from the original authors’ works.

2.1 Machine Learning

Machine learning is inherently a multidisciplinary field. It draws on results from artificial intelligence, probability and statistics, computational complexity theory, control theory, information theory, philosophy, psychology, neurobiology, and other fields [28]. It is capable of adapting to new circumstances, detecting and extrapolating patterns. Here is a more formal definition by Dr. Tom Mitchell:

Definition: A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$ [28].

Simply speaking, machine learning is a program that can learn by itself. For example, to design a learning system to play chess, the task $T$ is playing chess, the performance measure $P$ is the results - win or lose, and the experience $E$ is to play thousands of games with humans or even with itself. So, the playing chess program
learns from thousands of games and is measured by the results to improve the ability to win. Generally speaking, machine learning builds up a model from data examples by a certain algorithm and makes decisions or predictions based on the pattern. The result is dynamic as the data examples increase, rather than following a strict static instruction.

Machine learning has been rapidly developed in the last decade throughout computer science and beyond. Machine learning is used in web search, spam filters, recommender system, fraud detection, stock trading, human-face processing, auto-drive cars and many more. Researchers believe that machine learning will be the next big wave of innovation. As a result, more and more people are taking part in machine learning research. There are literally thousands of papers available and hundreds of more are published each year, but it always consists of combinations of three components [15]:

- Representation. A classifier must be represented in some formal language the computer can handle.

- Evaluation. An evaluation function is needed to distinguish good classifiers from bad ones.

- Optimization. A method is able to search among classifiers in the language for the highest-scoring one.

Machine learning starts with an appropriate representation algorithm, followed by an evaluation function to evaluate the performance, and finally an optimization method to find the optimal solution. The representation is the key part of machine learning, but it’s easy to overlook the fact that evaluation and optimization are equally important.

By the form of the training data, machine learning can be roughly divided into two types - supervised learning and unsupervised learning. In the following sections, I will briefly talk about supervised and unsupervised learning.
2.1.1 Supervised Learning

A supervised learning algorithm takes some example input-output pairs and learns a function that maps from input to output [32], meaning each example in the training set consists of an input object and a desired output value.

For example, Table 2.1 shows real estate prices and the sizes of houses in Wolfville (data from http://www.royallepage.ca/en/ns/wolfville/properties, retrieved on Feb 17 of 2016):

Table 2.1: Real Estate Prices and the Sizes of Houses in Wolfville

<table>
<thead>
<tr>
<th>House No.</th>
<th>Size (square feet)</th>
<th>Price (thousand dollars)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1907.00</td>
<td>274.90</td>
<td>46 Carriageway Court</td>
</tr>
<tr>
<td>2</td>
<td>1080.00</td>
<td>150.00</td>
<td>104 Maple Avenue</td>
</tr>
<tr>
<td>3</td>
<td>3433.00</td>
<td>479.50</td>
<td>49 Bishop Avenue</td>
</tr>
<tr>
<td>4</td>
<td>2346.00</td>
<td>349.90</td>
<td>38 Minas View Drive</td>
</tr>
<tr>
<td>5</td>
<td>4109.00</td>
<td>475.00</td>
<td>118 Woodman Road</td>
</tr>
<tr>
<td>6</td>
<td>4099.00</td>
<td>364.90</td>
<td>14 Acadia Street</td>
</tr>
</tbody>
</table>

To simplify the question, we ignore the location, the age of the house and other features, but only explore the relation between the size of the house and the house price. This is a typical supervised learning with a paired data set \((x, y)\), where \(x\) is the size of the house and \(y\) is the price. Since \(x\) and \(y\) are numbers, this type of learning problem is also called regression. Particularly in this case, we can apply linear regression or quadratic regression to find a fit function. To be noted, there is no absolutely correct answer on prediction function because the dataset is always introducing new data. The optimal solution is based on the current training set and finding the highest scoring function in evaluation.

In Figure 2.1, we plot two fit functions (suggested as a red line and a yellow curve in the figure) using linear regression and quadratic regression, respectively.

Another typical supervised learning problem is classification. Similarly with regression, the data for a classification problem is still a paired data set \((x, y)\), but \(y\) is more of a category than numbers. Here is an example of Self-Reports of Height and Weight dataset from car package of R (retrieved from https://vincentarelbundock.github.io/Rdatasets/csv/car/Davis.csv). Table 2.2 gives a peek of the data.
Figure 2.1: Applying Linear and Quadratic Regression on House Price Problem

Table 2.2: A Sample of Men and Women’s Height and Weight

<table>
<thead>
<tr>
<th>No.</th>
<th>Weight (kg)</th>
<th>Height (cm)</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77</td>
<td>182</td>
<td>Male</td>
</tr>
<tr>
<td>2</td>
<td>58</td>
<td>161</td>
<td>Female</td>
</tr>
<tr>
<td>3</td>
<td>53</td>
<td>161</td>
<td>Female</td>
</tr>
<tr>
<td>4</td>
<td>68</td>
<td>177</td>
<td>Male</td>
</tr>
<tr>
<td>5</td>
<td>59</td>
<td>157</td>
<td>Female</td>
</tr>
<tr>
<td>6</td>
<td>76</td>
<td>170</td>
<td>Male</td>
</tr>
</tbody>
</table>

Table 2.2 lists first six entries of the sample data. If we treat the height and weight as input data and their gender as output data, we can set up a classifier to distinguish a person’s gender based on his or her height and weight. When we plot 20 entries in Figure 2.2, it is easy to see that the data tends to be in two categories.

There are a couple of classification algorithms, such as k-NN (k nearest neighbours) and logistic regression (it is a regression, but often used in classification problem). The key feature in supervised learning is that the dataset always has a output label, either a real number or a category. What about those data without a output label?

### 2.1.2 Unsupervised Learning

An unsupervised learning is a task to discover the hidden structure and patterns even when no explicit feedback is supplied [32]. As we mentioned before, simply speaking,
unsupervised learning is dealing with the data with no output given. In a word, unsupervised learning received no error or reward for further training or evaluation. Therefore, the training data is not as restricted as supervised learning in practical problems and unsupervised learning is more widely used in machine learning. One of the typical problems is clustering.

Clustering is a task of grouping a set of data according to some similar features. Generally speaking, there is no “correct” way to group data, so they do not have a output label. There is no single specific standard in clustering either, as it is achieved by a variety of algorithms depending on how we measure each feature. K-means clustering is one of the most widely used unsupervised learning algorithms. The procedure follows a simple and easy way to classify a given data set through a certain number $k$ of clusters. The main idea is to define $k$ centers, one for each cluster. Initially, we will randomly assign $k$ centroids to $k$ points. For each point, we will find the nearest centroid and assign it to that cluster. After all the points are assigned to a cluster, we will recalculate the centroid for each cluster. With the new centroids, we will repeat the previous steps until there are no cluster assignments change. Here is an example of using k-means to solve clustering problem.

In R, there is a sample dataset preloaded in the environment called *iris*. Iris is a type of flower. It took the name from the Greek word for a rainbow (Wikipedia).
The sample dataset *iris* is a data frame with 150 cases and five variables named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width and Species. There are three types in Species: setosa, versicolor and virginica. In this problem, I will discard the variable Species when training, so we can use k-means to cluster data as an unsupervised learning problem. After the training, we will use these three types as the evaluation for this model. The data were collected by Edgar Anderson in 1935, which is now used as a demonstration data sample in R. Table 2.3 shows the top 10 data entries in the data frame.

Table 2.3: A Sample of *iris* data frame

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>4.9</td>
<td>3.0</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>4.7</td>
<td>3.2</td>
<td>1.3</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>4.6</td>
<td>3.1</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>5.0</td>
<td>3.6</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>5.4</td>
<td>3.9</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>7</td>
<td>4.6</td>
<td>3.4</td>
<td>1.4</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>5.0</td>
<td>3.4</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>9</td>
<td>4.4</td>
<td>2.9</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>4.9</td>
<td>3.1</td>
<td>1.5</td>
<td>0.1</td>
</tr>
</tbody>
</table>

We use k-means algorithm to apply the 150 observations with 4 variables with 3 clusters. Here is the summary for this experiment:

- K-means clustering with three cluster sizes of 62, 50, 38.
- Table 2.4 shows cluster means.

Table 2.4: The Cluster Means of 3 Clusters after K-means Process

<table>
<thead>
<tr>
<th>Clusters</th>
<th>Sepal.Length</th>
<th>Sepal.Width</th>
<th>Petal.Length</th>
<th>Petal.Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.901613</td>
<td>2.748387</td>
<td>4.393548</td>
<td>1.433871</td>
</tr>
<tr>
<td>2</td>
<td>5.006000</td>
<td>3.428000</td>
<td>1.462000</td>
<td>0.246000</td>
</tr>
<tr>
<td>3</td>
<td>6.850000</td>
<td>3.073684</td>
<td>5.742105</td>
<td>2.071053</td>
</tr>
</tbody>
</table>
• Here is a comparison between the actual species and the clusters in Table 2.5. The species - setosa, versicolor and virginica - has 50 observations each. The clusters generated by k-means is shown in this table as well. We can calculate the accuracy, which are 100%, 96% and 72% for each species respectively. The total accuracy for this problem is 89.3%.

Table 2.5: The Comparison between the Actual Species and Clusters

<table>
<thead>
<tr>
<th></th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>setosa</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>versicolor</td>
<td>48</td>
<td>0</td>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>virginica</td>
<td>14</td>
<td>0</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>62</td>
<td>50</td>
<td>38</td>
<td>150</td>
</tr>
</tbody>
</table>

At last, we plot the data into three clusters to better illustrate, which are shown in Figure 2.3. However, each observation has four variables and it cannot be plotted on two-dimensional space. So we split the variables into two sets, one is with the length and width of sepal, the other is the length and width of petal. Those two plots are showing the two sets respectively.

![Figure 2.3: Two K-means Clusters Plots by Sepal and Petal](image)
2.2 Information Retrieval

Information retrieval (IR), as we mentioned in Chapter 1, can be traced back to 3000 BC, when the Sumerians designated special areas to store clay tablets with cuneiform inscriptions [35]. Since the concept of information retrieval has been for thousands of years, the meaning of the term is literally very broad. If you shop online, pick up a credit card, and type in the credit card number, this would simply be a form of information retrieval. If you open a phone book and get a specific number you need, this will also be a form of information retrieval. However, in machine learning and where this thesis is focused, information retrieval is often used in a much larger data set and finding something more interesting and meaningful than a number from a phone book. As an academic field of study, Dr. Christopher Manning defined information retrieval as thus:

Information retrieval is finding material of an unstructured nature that satisfies an information need from within large collections [26].

Information Retrieval used to be an activity for the people who work with “searching”, such as librarians and professional researchers. But along with the Information Age, information retrieval is widely used by millions of people every day, as in search engines, social media, weather forecast and even in sports analysis. It has become the most dominant form of information access.

2.2.1 Information Retrieval Tasks

Information retrieval is such a large task that it almost contains every tasks associated with information. But roughly speaking, there are three typical tasks on perspective of information: information representation, information ranking, and document similarity [22].

Information Representation. From the perspective of information, at the early stages of information retrieval, we often work on the information itself, such as indexing, encoding, breaking into segments, parsing and so on. This is because information is inherently concrete, definable and encodable. In the late 1980s, some researchers started to suggest that the context, task, or situation is also important. With the
fast development of machine learning, lots of representation methods, and more forms of information such as images, audio and videos enhanced the study of information retrieval.

**Information Ranking.** Ranking is the ordering of retrieved documents in response to users’ input, typically a query. In the early ages, especially for relational database system, the result could be a set of data which are all correct. In current information retrieval systems, the results of data can be algorithmically ranked to the degree that match the query. It becomes very clear and important when it comes to search engines. One query can lead to millions of results where the ranking is extremely significant. The results are ranked by decreasing probability of relevance to the users’ query. However, there is no one ranking algorithm or evaluation method for all information retrieval systems. For this research, we will cover the information ranking and documents similarity method in Chapter 3.

**Document Similarity.** Based on the Rule of Document Similarity, if one can associate a query with a relevant document, other documents that have similar characteristics also will be relevant [22]. Simply speaking, if you are reading one particular document from a certain query, and other documents have some similar characteristics from that same query, they are relevant. Document similarity can be measured by some certain algorithm between documents or even simply use the information ranking. However, similarity is an ambiguous concept, which can be affected by various features other than the content itself, such as time, place, language, and even more personal preference. Thus, plenty of information retrieval systems based on personal interests are quite popular these years. According to the information such as browsing history or users’ manual settings, the similarity calculating algorithms are becoming more personal and more accurate.

### 2.2.2 Information Retrieval Models

In order to rank documents by their estimation of the usefulness of a document, most information retrieval systems assign a numeric score to every document and rank them in a certain order. Several models have been proposed for this purpose
and the most used ones are the vector space model, the probabilistic model and the inference network model [35]. I will give a brief introduction of the three methods in this section.

**Vector Space Model**

In the vector space model, each document is represented by a vector of terms. The definition of a term is not inherent in the model, but terms are typically single words, keywords, or longer phrases. If words are chosen to be the terms, the dimension of the vector is the number of words in the vocabulary. In this way, any document can be represented by a vector in a relatively high dimensional space. Since a query is also text, it can then be converted to a vector as well. Documents and queries can be represented as follows:

\[ \vec{d}_j = (w_{1,j}, w_{2,j}, ..., w_{i,j}) \]
\[ \vec{q} = (w_{1,q}, w_{2,q}, ..., w_{i,q}) \]

where \( w_{i,j} \) is the value of the \( i \)th term in the vector of the \( j \)th document and \( w_{i,q} \) is the value of the \( i \)th term in the query vector \( \vec{q} \). The terms may be weighted according to their importance by \( w \) [33]. To assign a numeric score to a document from a query, the model just simply measures the similarity between the query vector and the document vector. Typically, the angle between two vectors is used as a measure of divergence between two vectors, and the cosine of the angel is used as the numeric similarity, as shown in Equation 2.1. To simplify the calculation, the inner-product between two vectors are often used for similarity measure as an alternative, which are shown in Equation 2.2.

\[ \cos \theta = \frac{\vec{d}_j \cdot \vec{q}}{\| \vec{d}_j \| \| \vec{q} \|} \] (2.1)
\[ Sim(\vec{d}_j, \vec{q}) = \sum_{i=1}^{n} w_{i,j} \cdot w_{i,q} \] (2.2)
Probabilistic Model

The probabilistic model is based on the probabilistic ranking principle (PRP) purposed by Drs. Cooper and Robertson in 1977.

The probability ranking principle (PRP): “If a reference retrieval system’s response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data” [31].

The principle indicates that documents in a collection should be ranked by decreasing probability of their relevance to a query. The true probabilities are impossible to get since the queries are from users. Even with the same query, different users hold different opinions on the document relevance. Therefore, probabilistic IR models estimate the probability of relevance of documents for a query and make the estimation as accurate as possible. Most of current probabilistic IR systems are based on this principle.

We denote the probability of relevance for a given document $D$ by $P(R|D)$. Under log-odds transformation, we can rank documents by $\log \frac{P(R|D)}{P(\bar{R}|D)}$, where $P(\bar{R}|D)$ is the probability that document is non-relevant. Applying Bayes’ Rule, it can be transformed to $\log \frac{P(D|R)P(\bar{R})}{P(D|\bar{R})P(\bar{R})}$. Considering $P(R)$ is independent of the document and thus is a constant, the formula can be further simplified to $\log \frac{P(D|R)}{P(D|\bar{R})}$. We assume the terms in the model are mutually independent, so $P(D|R)$ can be rewritten as [31]:

$$P(D|R) = \prod_{t_i \in Q,D} P(t_i|R) \cdot \prod_{t_i \in Q,\bar{D}} (1 - P(t_i|R))$$ (2.3)

If $p_i$ denotes $P(t_i|R)$, and $q_i$ denotes $P(t_i|\bar{R})$, the ranking formula reduces to:

$$\log \frac{\prod_{t_i \in Q,D} p_i \cdot \prod_{t_j \in Q,\bar{D}} (1 - p_j)}{\prod_{t_i \in Q,D} q_i \cdot \prod_{t_j \in Q,\bar{D}} (1 - q_j)}$$ (2.4)
CHAPTER 2. BACKGROUND

For a given query, we can add to this a constant \( \log(\prod_{t_i \in Q} \frac{1-q_i}{1-p_i}) \) to transform the ranking formula to use only the terms present in a document:

\[
\log \prod_{t_i \in Q,D} \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)}
\]  

or

\[
\sum_{t_i \in Q,D} \log \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)}
\]

Different probabilistic models diverges in some points but the assumption is basically the same. The LDA topic model is also based on probabilistic model, and we will cover more about this in Section 2.4.

Inference Network Model

In this model, information retrieval is an inference or evidential reasoning process in which we estimate the probability of a user’s needs. A document instantiates a term with a certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document. The strength of the document can be considered as the weight of the document terms. The document ranking is also quite similar to the vector space model and probabilistic model that we mentioned previously. The strength is not defined by the general model, it depends on the particular case [38].

2.3 Text Mining

Data mining, also called knowledge discovery in database (KDD), is the process of discovering interesting and useful patterns and relationships in large volumes of data [13]. Generally speaking, data mining discovers all types of data, including text, images, audio, videos and etc. Text mining is a subfield of data mining focusing on text. It is one of the most important tasks of data mining. Here is a definition of text mining from the UK Government archives:

“Text mining is the process of deriving information from machine-read material.
It works by copying large quantities of material, extracting the data, and recombining it to identify patterns” [19].

Simply speaking, the value of text mining to scientific research is that text and data mining uses existing knowledge to discover new hidden relationships [19]. Why do we need text mining? Computers have a magnificent power of computing. Even a standard PC, available in every family, can do 2 billion processes in a second. It is beyond human capability. However, it took several super computers training years with low accuracies to do some tasks that can be done easily by humans, such as recognizing an object in a certain environment, understanding an article, or even acting like a human. These are the problems that the artificial intelligence scientists are trying to solve and the drive to push the field forward. In text mining, the main problem is that we are trying to let the computers “understand” the text. The word understand is quoted, because it is impossible for computers to actually understand the text (maybe that could be done in the future). We are applying algorithms to take advantage of the fast computation to analyse the text under some predefined text customs and rules, so that the computers can discover the deeper meaning of the text or even respond like human.

So far, we have achieved great progress on text mining. One achievement is natural language processing (NLP). Basically speaking, NLP is the process for computers to understand human language. Computers can understand the human language query and respond with human language as well. It started in the early 1950s, when Alan Turing published an article titled “Computing Machinery and Intelligence” which proposed what is now called the Turing Test as a criterion of intelligence. The Turing Test was designed to provide a satisfactory operational definition of intelligence [32]. No single program has passed the Turing Test yet, but we are getting there. On our smart phones, there are some virtual assistants, such as Siri from Apple, Google Now from Google and Cortana from Microsoft. These assistants are programs, capable of processing natural language to understand human requests, using cloud computing and data mining to find the answer, and then respond with actions or language. Besides that, we are also making text mining algorithms for classification, extraction, and prediction.
In this section, we will first talk about the landscape of text mining, and describe the components in text mining. Next, we will briefly introduce some major tasks in text mining.

2.3.1 The Landscape of Text Mining

For a researcher, there is too much literature to read. The scholar publication base consists of 11,550 journals, to which 1.5 million articles are added per year [27]. Furthermore, online research resources such as social networking communications are too numerous to read for a single researcher. Traditional keywords search could reduce and refine the number of documents, however the searching accuracy is not good enough. We need a way to efficiently group documents and find information we actually care about.

Text mining offers a solution by drawing on techniques from information retrieval, natural language processing, information extraction and data mining, which is shown in Figure 2.4.

There are four stages in text mining. Stage 1 is enhanced information retrieval, which retrieves all the related documents based on the user’s query. Stage 2 is linguistic analysis, also called natural language processing. It processes the language into machine readable structured text data, and useful information can then be extracted in Stage 3. The final stage is data mining. The text data can be mined to discover
new knowledge, meaningful patterns and identify relations.

Figure 2.5: Text Mining Process [27]

The process of text mining is very similar with researchers doing research. Researchers first come up with a topic and search through the whole collection looking for the related documents. Then they deeply read these documents, make annotations and take notes. Lastly, they analyse the materials and draw conclusions based on the documents and what they have learned [39]. Text mining is based on this process with some minor changes. As shown in Figure 2.5, text mining starts with a whole collection of documents. Using information retrieval techniques, it will retrieve all of the related documents according to the query. Then it applies some basic text mining techniques to clean the data to a certain format for later processing. This process is also called normalising. The next step is textual analysis, including lexical, entity identification. This step will analyse and process the normalised documents to a derived dataset, some of which are capable of semantic interrogation. This dataset is the result of text mining. It could be as simple as a collection of refined information, a system that organizes the documents, or even a smart knowledge dataset that can
be queried for information. Last step, this dataset can provide certain information or discover new knowledge and patterns.

2.3.2 Text Mining Tasks and Algorithms

With the great efforts from text mining researchers, so far we have plenty of algorithms solving lots of text mining tasks. Some we are using every day, and some others we are taking advantage of the techniques and do not even realize. In this section, we will introduce some major text mining tasks and related algorithms.

Classification. Text classification is to assign a known label to text documents. It is a supervised learning algorithm to classify documents into certain groups. One of the applications is email spam filtering system. By learning users spamming emails, the system can classify emails to spam and normal. The most widely used algorithm is Naive Bayes Classifier.

Clustering. Clustering is to group similar documents. It is often used to organize large amounts of documents for easier access and index. Clustering is an unsupervised learning problem, meaning the groups are unknown. The basic clustering algorithm creates a vector of terms to represent the document. The terms could be keywords, a certain combination of words that summarize the document, or simply including every word in the document if the length is reasonable. K-means, which was introduced in Section 2.1.2, can be also used in text clustering.

Keyword Extraction. Keyword extraction (KE) is tasked with automatic identification of a set of terms that best describe the subject of a document [5]. It is an important problem in text mining, information retrieval and natural language processing. There are several methods for keyword extraction, which can be roughly divided into statistical approach, linguistic approach, and graph-based approach.

Summarization. Text summarization is to retain the useful information and main points in the documents and largely reduce the length [16]. When humans write a summarization of a document, we often read through all the documents and list the main idea and refine a conclusion. While computers are good at large computation, it is still a big challenge to interpret the meaning of a document. So text summarization
uses a different strategy to do the job. It extracts original sentences by statistically weighing the sentences. When coming to certain keywords or phrases, such as “in conclusion”, “to sum up”, the algorithm will extract those sections behind them.

2.4 Topic Models

In the previous section of Text Mining Tasks and Algorithms, we did not talk about one of the most important algorithms on purpose, which is the topic model. It is the core technique in this thesis, we will deeply introduce topic models in this section.

A topic model is a statistic model to discover hidden structure in a collection of documents. The topics are what a document mainly talks about. Intuitively, a document is about a particular topic, and those topic words tend to appear more often than others [6]. For example, in sports news, words like “players”, “games”, and “scores” will appear more often, while in tech news, words like “tech”, “artificial”, and “smart” will appear more often. Words like “the”, “a”, and “if” will appear almost the same in all the documents. A topic model is based on these topic proportions. A topic is a set of terms. Each word in the document is drawn from a particular topic. So a document is just about several topics but with different proportions. There are several models for topic modelling. The most commonly used is Latent Dirichlet Allocation (LDA), which we will talk about in Section 2.4.2.

The topic model has been proposed only for a little more than a decade. However, it has drawn so much attention that more and more papers are published to explore lots of directions of this model. It is now widely used in document management. Other than this, topic models are also introduced to solve image recognition and make progress in other fields such as bioinformatics.

2.4.1 Bag of Words Assumption

There is an important assumption, which is not only fundamental to topic models, but also to many other text mining models, called “bag of words assumption” (or “bag of words model”). Bag of words assumption (BoW) treats each document to
be a bag of words disregarding grammar and even word order but keeping multiplicity [40]. It breaks the documents into words, which is also refereed to tokenization. A set of words is extracted to represent the document. This technique ignores the grammar and the order of words. What we get is a set of words that best describe the document in an unordered way. The bag of words assumption is widely used in document classification. Recently, the assumption was modified and introduced to image processing, where each image is treated as a bag of image features.

How does the bag of words assumption represent a document? Here is an example of two very short documents represented by this model. To simplify the problem, we use two extremely short documents which contains only one sentence.

Document 1: Joe likes basketball.

Document 2: Alex likes Korean food, but Joe likes Chinese food.

The first step is to extract vocabulary for the documents. Vocabulary is a set of distinct words which has appeared in the documents. A list of vocabulary can be conducted as:

[“Joe”, “likes”, “basketball”, “Alex”, “Korean”, “food”, “but”, “Chinese”]

Vocabulary is a base vector of all the documents. To interpret each document, we need to denote 0 or 1 to each word in the vocabulary to show whether the attendance of the word. Here is how we represent these two documents in bag of words assumption:

Document 1: [1, 1, 1, 0, 0, 0, 0, 0]

Document 2: [1, 2, 0, 1, 1, 2, 1, 1]

As we can see in the Document 1, if the word shows up, we denote to 1, otherwise 0, while in Document 2, there are two words represented by 2, which means the frequency of the word is 2. Besides the 0 and 1 for appearance or not, we can also tell the frequency of the word. This model of representation is like a histogram over each word in the vocabulary. However, the order of the word for the original document is not preserved in this model.

If we transfer the collection of documents via this format to be a matrix as shown in Table 2.6, it is also called Document-Term Matrix (DTM). The words in each document are weighted by their frequency. There are many other methods for word weighting. The most common one is TF-IDF schema. But since LDA topic model is
not required for TF-IDF, we will not cover this method in the thesis.

Table 2.6: Document-Term Matrix of the Example Collection of Documents

<table>
<thead>
<tr>
<th></th>
<th>Joe</th>
<th>likes</th>
<th>basketball</th>
<th>Alex</th>
<th>Korean</th>
<th>food</th>
<th>but</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Document 2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2.4.2 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) was first introduced by Drs. David Blei, Andrew Ng, and Michael Jordan. They described LDA “a generative probabilistic model of discrete data such as text corpora” [9]. The intuition behind LDA is that documents exhibit topics. For example, when we start to write a document, we need to identify what topics we want to talk about. If I am writing a document about the town of Wolfville, I may involve Acadia University, a good vineyard, beautiful sea scenery and some others. So I would probably use topic words about academic studies such as “university”, “students”, the topic words about vineyard like “wine”, “grapes”, and also the topic words about scenery as “waterfront”, “breeze”. Generally speaking, a topic is a distribution over terms; a document is a mixture of words from each topic [6]. LDA is trying to catch this idea in a statistical model. Here is a generative process for LDA [30]:

1. For each topic, choose a distribution over terms.

2. For each document,

   (a) Choose a proportion of each topic, meaning what topics this document is talking about.

   (b) For each word in this document,

      i. Choose a topic

      ii. Given this topic, choose a likely word (generated in Step 1)
CHAPTER 2. BACKGROUND

To better explain the process, we use an example from Dr. Blei’s paper [6]. Figure 2.6 shows an illustration of LDA topic model for an article from Science magazine called “Seeking Life’s Bare (Genetic) Necessities”. LDA topic model is based on the bag of words assumption, so the document is a mixture of words. It ignores the order of the words and English grammar. On the left hand side, there is a set of topics. In each topic, it is a distribution over each term in this topic, which is called term distributions per topic. On the right side, there is a little histogram figure, it is called topics distribution per document. It shows the proportion of each topic in the document. The set of circles point to highlighted words, which is called topic to words assignments. It represents given this topic, what the probability of word is in this document.

In my own words, the procedure is trying to figure out the structure of the documents. In the example of Figure 2.6, the model will first highlight the words (or terms) that are actually meaningful. We will ignore the stopwords and punctuation in the pre-processing. These highlighted words are hidden topics but they are in a bag. The model will extract the terms into topics by their frequency and associations. For each topic, we will calculate the probability of each term, because they are weighted
differently in different topics. Given a document, we can get the distribution over topics. For each word in this document, we can get the probability for each word given what topics they are from. It is a little bit similar to reversing the steps of an author writing the article. If I am writing this article, first I will think about what topics I am going to use, perhaps about genes, life, or brain. Then I will combine the words together with some grammar and other common sense to make it readable. But the model is doing it from the other way. It will first remove all those words and punctuation only affecting the grammar, and then generate the topics for the corpus. It will assign each word to a topic with a probability. The difference is that the LDA model is trying to follow this process in a statistical method.

2.4.3 The Dirichlet Distribution

The LDA model draws samples from the Dirichlet distribution and from the multinomial distribution and makes central use of the Dirichlet distribution. The Dirichlet distribution’s probability density function is defined as:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i-1}$$

with $\alpha$ being a positive K-vector and $\Gamma$ denoting the Gamma function, which is a generalization of the factorial function to real values [8]. The Dirichlet is a convenient distribution on the simplex - it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. It tells us the probability of observing a count of two or more independent events, given the number of draws and fixed probabilities per outcome that sums to one [9].

Recall the generative process for LDA in the last section (Section 2.4.3), we can also summarize the probabilistic generative process [8]:

1. For each topic $k$, draw a distribution over words $\beta_k \sim Dir(\eta)$.

2. For each document $d$,

(a) Draw a vector of topic proportions $\theta_d \sim Dir(\alpha)$. 
(b) For each word $i$ in this document,

i. Draw a topic assignment $z_{d,i} \sim \text{Mult}(\theta_d)$, $z_{d,n} \in \{1, 2, ..., K\}$,

ii. Draw a word $w_{d,i} \sim Mult(\beta_{z_{d,n}})$, $w_{d,n} \in \{1, 2, ...V\}$

This process can also be illustrated as a graphic model in plates notation shown in Figure 2.7.

![Figure 2.7: A Graphic Model Representation of LDA [6]](image)

Now we can make a summary of the LDA generative process with the following notation. The topics are $\beta_{1:K}$, where each $\beta_k$ is the distribution over the vocabulary. The topic proportions for the $d$th document are $\theta_d$, where $\theta_{d,k}$ is the topic proportion for topic $k$ in document $d$. The topic assignments for the $d$th document are $z_d$ where $z_{d,n}$ is the topic assignment for the $n$th word in document $d$. The observed words for document $d$ are $w_d$, where $w_{d,n}$ is the $n$th word in document $d$, which is an element from the fixed vocabulary. In the graphic model, $W_{d,n}$ is the observed variable and $\beta_k$, $\theta_d$, $Z_{d,n}$ are hidden variables. With the notations and the generative process, we can conclude the joint distribution of the hidden and observed variables as $[6]$

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d)(\prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n}))$$

We can also compute the conditional distribution of the topic structure given the observed documents, which is also called posterior computation. With the same
notation it is shown as:

\[ p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \]  

(2.9)

2.4.4 Gibbs Sampling

LDA topic modelling is a sampling-based algorithm that attempts to collect samples from the posterior to approximate it with an empirical distribution. The most commonly used sampling algorithm is Gibbs sampling. But before we step into Gibbs sampling, I would like to first give a quick introduction to its source algorithm - Markov Chain Monte Carlo (MCMC).

In the January/February 2000 issue of Computing in Science & Engineering, a joint publication of the American Institute of Physics and the IEEE Computer Society, highlighted the top 10 best algorithms that have had the greatest influence of the 20th century. The metropolis algorithm, which is an instance of a large class of sampling algorithms, known as Markov Chain Monte Carlo was among those 10 algorithms [12]. It has played a significant role in statistics, econometrics, physics and computing science over the last two decades. The motivation behind MCMC method is to solve integration and optimisation problems in large dimensional spaces. The MCMC method attempts to draw samples from a complex distribution of interest based on constructing a Markov chain in order to approximate the desired distribution [4].

Gibbs sampling is one of the MCMC algorithms, which obtains a sequence of observed data samples from a joint distribution of multiple random variables. It constructs a Markov chain, which is a sequence of random variables, each dependent of the previous. Its limiting distribution is the posterior. The Markov chain is defined by the hidden topic variables for a particular corpus and the algorithm is to run the chain for a long time, collect samples from the limiting distribution, and then approximate the distribution with the collected samples [20].

For example, given a sample each of the random variables \( x_i \) is sampled in turn conditioned on all other variables, which can be expressed as \( p(x_i|x_{-i}) \) where \( x_{-i} \) is denoted for \( x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \) that all the other variables but \( x_i \). Suppose
there are only two random variables $x$ and $y$. In order to compute the joint distribution of $(x, y)$, during each iteration, the Gibbs sampler will start with sampling $x_1$ from the conditional distribution $p(x|y = y_1)$ given the initial value of $y_1$. Then the sampler will generate the value of $y_2$ with the previous value $x_1$ and the conditional distribution $p(y|x = x_1)$. So in general, the process for sampling $x_i$ and $y_i$ is:

$$x_i \leftarrow p(x|y = y_{i-1}) \tag{2.10}$$

$$y_i \leftarrow p(y|x = x_i) \tag{2.11}$$

If we run the iterations sufficient times, the dataset $(x_i, y_i)$ is able to estimate the full joint distribution [23].

In LDA topic modelling, we represent the collection of documents by a set of word indices $w_i$ and document indices $d_i$, for each word token $i$. The Gibbs sampling procedure considers each word token in the text collection in turn, and estimates the probability of assigning the current word token to each topic, conditioned on the topic assignments to all other word tokens. From this conditional distribution, a topic is sampled and stored as the new topic assignment for this word token.

### 2.5 Jensen-Shannon Divergence

With the LDA topic modelling, we can get the set of topics distribution over each document $\theta_d$, which is a vector of proportions of each topic for a document $d$. This can help to answer questions about the similarity of words and documents. In this section, we will talk about methods to calculate the similarity - information gain, Kullback-Leibler divergence and Jensen-Shannon divergence, which are actually closely related.

Information theory was originally developed by Claude E. Shannon to find fundamental limits on signal processing and communication operations such as data compression. He proposed the concept of entropy (or Shannon Entropy) which is the expected value (average) of the information contained in each message [25]. The higher the entropy, the more information it contains. For a set of variable $X$, the probability for a variable $x$ in $X$ is $p(x)$, then the entropy $H(X)$ of the set of variables
can explicitly be written as:

\[ H(X) = -\sum_{x \in X} p(x) \log p(x) \]  

(2.12)

In many applications of information theory, ranking the importance of predictive variables is one of the most key tasks. Information gain can tell us the importance of a given variable for discriminating between classes to be learned. In other words, in a vector of training variables, which feature is to most gain the information. Equally speaking, the entropy that it produce is the least. The information gain can be written as:

\[ IG(T, a) = H(T) - H(T|a) \]  

(2.13)

where \( H(T) \) is the entropy of the whole training vector \( T \), \( H(T|a) \) is the entropy of the vector given variable \( a \).

Kullback-Leibler divergence (also KL divergence) follows this idea. It is a measure of the difference between two probability distributions \( P \) and \( Q \). In applications, \( P \) typically represents the actual distribution of data while \( Q \) typically represents a theory, model, description, or approximation of \( P \). KL divergence is denoted as \( D_{KL}(P||Q) \), is a measure of the information gained when one revises ones beliefs from the prior probability distribution \( Q \) to the posterior probability distribution \( P \). In other words, it is the amount of information lost when \( Q \) is used to approximate \( P \) [10]. For discrete probability distributions \( P \) and \( Q \), the Kullback-Leibler divergence of \( Q \) from \( P \) is defined as:

\[ D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \]  

(2.14)

Jensen-Shannon divergence (JS divergence) could be treated as an improvement version of KL divergence. It is mostly based on KL divergence but with some notable and useful differences, including that it is symmetric and it is always a finite value. The square root of the Jensen-Shannon divergence is a metric often referred to as Jensen-Shannon distance [29]. JS divergence is a symmetrized and smoothed version
of KL divergence, JS divergence is defined as:

\[ JSD(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M), \quad (2.15) \]

where \( M = \frac{1}{2}(P + Q). \)

In this experiment, we will apply JS divergence to calculate the similarity between the words and documents. Particularly, we will calculate JS distance between every two documents and that is the metric for finding similar documents.
Chapter 3

Experiment Design

In this chapter, I will deeply go through the Wikipedia experiment for this thesis. In order to implement LDA topic model and explore its performance, I will propose a series of experiments from scratch, which consists of data retrieving, pre-processing, fitting the model and evaluation. The experiment data is simple Wikipedia documents collection. By applying these series of experiments, we can get a clear look at how LDA topic model works under practical circumstance, and the fitted model will be applied to implement a document exploring system.

3.1 Overview

LDA topic modelling is an unsupervised machine learning algorithm based on a probabilistic model. It is widely used in documents management and analysis. For a large collection of documents, LDA topic model can efficiently process documents and extract hidden topics. The topics imply what the document is mainly talking about. LDA topic model can process the documents to a vector of topics distribution, which is the key structure of the document contents. We can then apply distribution divergence to calculate document similarity and expand to many directions. The model provides the base to explore and analyse documents on the contents level and it has attracted lots of attention since it was proposed. LDA topic model was also introduced to image and video processing, but in this thesis, I will only cover the text
In order to get better performance and reduce the processing time, the running environment of this thesis is the Elastic Compute Cloud (EC2) from Amazon Web Service (AWS). It allows users to rent virtual computers to run applications. It provides many options to customize and meet your needs. This service can enhance the calculating power much beyond my laptop.

The coding environment is R programming language, which is widely used for statistical computing. R is a open source tool and is provided for various operating systems. One of the good features in R is the capability to extend by installing user-created packages. These packages largely enhance the functionality of R language. In this experiment, I will use many R packages to process the data, such as \texttt{tm}, \texttt{topicmodels}.

The experiment data is Simple Wikipedia. It provides a lot of entries of term explanation. Wikipedia is a knowledge base consists of a large amounts of essential human knowledge, and it is worth analysing and organizing. It is also a free collection so it is easier to get access.

The experiment basically consists of 4 parts - data retrieving, data pre-processing, model fitting and evaluation. The workflow is shown in Figure 3.1. The first step is retrieving the data from Wikipedia, but the format can not be processed directly by the algorithm. We need a series of pre-processing steps to make the raw text data clean and usable. It includes parsing the text data, formatting to the right style, tokenization, stopwords removing, and stemming. The pre-processing can transfer the raw text data into a document-term matrix which is required by the bag of words assumption. Then we will apply the text data to fit the LDA topic model and the performance will be measured by a set of evaluation algorithms.

In this chapter, I will first introduce the running environment and coding language of the experiment and then I will go through every step from data retrieving to evaluation. The results and performance comparison will be illustrated in Chapter 4.
3.2. Environment

3.2.1 Running Environment

The running environment is the Elastic Compute Cloud (EC2) from Amazon Web Service (AWS), which is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers [34]. It is a cloud service with virtual computers. The users are allowed to rent customized servers to run their own applications. You have the choice of multiple instance types,
operating systems, and software packages. Amazon EC2 allows users to select a configuration of memory, CPU, instance storage, and the boot partition size that is optimal for your choice of operating system and application.

Here is the basic information of the EC2 instance for this experiment shown in Table 3.1.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Amazon Linux AMI (64 bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS version</td>
<td>Ubuntu 14.04.3 LTS</td>
</tr>
<tr>
<td>vCPU</td>
<td>2.5GHz, Intel Xeon Family</td>
</tr>
<tr>
<td>Number of vCPUs</td>
<td>1</td>
</tr>
<tr>
<td>Memory</td>
<td>1G</td>
</tr>
<tr>
<td>Storage</td>
<td>EBS 30G</td>
</tr>
<tr>
<td>R version</td>
<td>3.2.3</td>
</tr>
<tr>
<td>IDE</td>
<td>RStudio Server</td>
</tr>
</tbody>
</table>

### 3.2.2 Coding Language

The coding language is R programming language. It is an open source software environment for statistical computing and graphics. R is widely used by statisticians and data mining scientists to do data analysis. The home page of R is [https://www.r-project.org](https://www.r-project.org) where you can download the source code and other documents.

R is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity [18].
3.2.3 R Packages

One of the standard and most important capabilities of R is extensible by so-called packages which can be installed by the user and can be loaded from within an R session. R packages can be installed by the users to allow specialized statistical techniques. The packages are defined by other users as well. R packages enlarge the scope to solve a variety of problems. A central repository of such additional packages by third parties is the Comprehensive R Archive Network (CRAN). Cran mirrors are available around the world and currently host more than 3,400 packages from a wide range of categories, such as econometrics, graphics or machine learning.

In this section, I will introduce two of the packages that I used most often in the experiment - \texttt{tm} and \texttt{topicmodels}.

\texttt{tm} is the R package for solving text mining tasks, which was written by Ingo Feinerer in 2008, and it is still updating. \texttt{tm} is mostly used for pre-processing text data. The main structure for managing documents in \texttt{tm} is called a corpus, representing a collection of text documents. A corpus is an abstract concept, and there can exist several implementations in parallel [17]. \texttt{tm} provides a set of reader functions to read raw data from different kinds of sources. After reading, the data will be stored in the corpus. The corpus can be then transferred or filtered by the function \texttt{tm.map()} and \texttt{tm.filter()}. The pre-processing transformations includes:

- Removal of punctuations, numbers and spaces;
- Removal of stopwords;
- Conversion to lower case;
- Stemming.

\texttt{topicmodels} is the R package for implementing topic models, which is written by Bettina Grun and Kurt Hornik in 2011. \texttt{topicmodels} provides the interface with David Blei’s open source implementations of LDA which is written in C programming language [21]. The input data depends upon the pre-processing from package \texttt{tm}. \texttt{topicmodels} provides an interface for implementing topic models in LDA and inclusion of other estimation methods of topic models.
CHAPTER 3. EXPERIMENT DESIGN

3.3 Data Retrieving and Pre-processing

The text data for this experiment is from Wikipedia. Wikipedia is a free-access, free-content Internet encyclopaedia, supported by non-profit Wikimedia Foundation. It has millions of articles for people to search, explore or even edit. However, many Wikipedia entries are very long and rarely used, and it is not an ideal dataset for this experiment. I chose to use the simplified version of Wikipedia called Simple English Wikipedia. The Simple English Wikipedia was started as a response to the needs of English learners and English teachers. The project began to see major contributions after 150,000 articles in the original English Wikipedia and articles in each of seven other languages had been created. Its articles are intended to be less complex, and can be used for free readings in the classroom, computer activities, and so on. Simple Wikipedia has been shortened in the length and written in more understandable language.

3.3.1 Legal Clearance

Before we started to download data from Wikipedia, I had checked for legal clearance that in this experiment the raw text data is free to download from Wikipedia Foundation and free of use under Terms of Use (https://wikimediafoundation.org/wiki/Terms_of_Use). Here is a quote:

“[…]The Wikimedia Foundation, Inc. ("we" or "us"), is a nonprofit charitable organization whose mission is to empower and engage people around the world to collect and develop content under a free license or in the public domain, and to disseminate it effectively and globally, free of charge.[…]”

Here is a more human readable summary of the terms in this very page:

“[…]You are free to:

• Read and Print our articles and other media free of charge.

• Share and Reuse our articles and other media under free and open licenses.

• Contribute To and Edit our various sites or Projects.[…]”
3.3.2 Data Retrieving

Wikipedia is an open source Internet encyclopaedia, which also provides free access for users to download the backups for offline reading. Currently, there is a few Windows desktop softwares that can parse the backups and display the documents. In this experiment, we need a raw data format of a file, so that we can access the data in R and do pre-processing. In order to get our preferred format, I started from scratch to parse the data rather than using the current parsing software.

The first step is to visit the Wikipedia backup page (https://dumps.wikimedia.org/backup-index.html), where you can find all the backups for Wikipedia. They are stored in four types - database backup dumps, static HTML dumps, DVD distributions and image tarballs. The page indicates that database backup dumps is a complete copy of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML, which is the one we are going to use. HTML dumps is also usable, but it requires to parsing HTML files from different directories. XML file is one large file, which is easier to handle by R language.

My copy of Simple Wikipedia XML file was downloaded on January 14th of 2016, and the timestamps of this backup file is “20160111”. The experiment data is all based on this version of Simple Wikipedia backup.

To help with parsing this large XML file (size of 527MB), I installed an R package called XML, which is used for parsing all kinds of XML and HTML files. Wikipedia backup XML file has some unique attributes in the root label for their own display and archive purposes, which we have to delete in order to work with XML package. The removal of these attributes will not affect the actual contents in the file. After the XML package parsed the XML file, we will get a huge list with 430,750 articles. We will only keep the titles and the documents, and discard other attributes like author, update date that are not going to be used in the experiment, thus we can reduce the data size smaller.

In the Simple Wikipedia, there are many documents linked to other documents because they are similar entries, for example soccer, English football and association football are one entry in Simple Wikipedia. Besides that, we do not want a document that is too short like only a few sentences. So we will discard any document which
has less than 100 characters, which reduces the number of documents to 141,915. Finally, considering the space, the computation time and the power of the current EC2 instance, I will set a seed and randomly sample 2000 articles as our experiment data.

3.3.3 Pre-processing

Text pre-processing, also called text cleaning, is one of the most important parts of this experiment. The purpose of pre-processing is to simplify the text data, eliminating as much as possible language dependent factors [24]. Articles are written in natural language for humans to understand. But in text mining, those data are not always easy for computers to process. In order to reduce the computation time, we always slim down the size of the text data by removing redundant data such as stopwords or numbers. Pre-processing is picky, tedious and time-consuming, but it will pay off in the end for the high quality data to fit the model and get a better result.

Here are the steps in pre-processing that could be necessary in all kinds of text mining algorithms:

- **Removal of punctuations, numbers and white spaces.** The computer cannot actually read, so for human, punctuations and white spaces are helpful to understand in terms of grammar. However to computers, they are just the same with other characters. So we can actually remove the punctuations and white spaces to simplify the document. Numbers are similar, though they have some meanings. In the LDA topic model, the basic unit is a word and the numbers will be treated as words. So it will become wired and meaningless when the numbers are separated from its context. In some cases such as history, 1776 can be treated as a meaningful word since it is the year United States declared independence. But in most cases, for example, there are 20 students in the library. The more meaningful words would be student and library, and number 20 is irrelevant. So we will remove numbers as well in this experiment.

- **Tokenization.** A document is treated as a string, removing all the punctuations and then partitioned into a list of tokens. The removal of punctuations,
numbers and white spaces will make the documents unreadable. But you can sense what they are trying to talk about because the core contents, or the topics are still there. It is based on bag of words assumption that the document is a bag of words. The basic unit is a word, which is drawn from the topics.

- **Conversion to lowercases.** This step is necessary because the frequencies of each word is useful in the probabilistic model. In the documents, there will be some letters in uppercase such as it is the first word in the sentence. This will cause the computer to treat “water” and “Water” to be different words. To avoid this situation, we can just simply convert all the letters to lowercases.

- **Removal of stopwords.** Stopwords are the words used to help the sentence be more readable but with no actual meanings. For example, “the”, “and”, “if”, they are necessary in the sentence but the meaning does not change if you remove them. The frequency of the stop words in most English documents are somehow similar. Removal of the stopwords will not change the meaning but can largely reduce the size of the document and increase the text mining quality. In the **tm** package, the common stopwords have been pre-defined. We can always take a look and decide to add or remove some of the stopwords if necessary.

- **Stemming.** As we just mentioned, the frequencies of each word is useful in the probabilistic model. Some of the words express the same meaning but they are in different participles or plural. For example, “interested” and “interesting”, “apple” and “apples”, “happy” and “happiness”. It is easier for human to recognize but for computers we need to stem the words. Stemming is the process to transfer the word to its root form. This process can also reduce the computation time and increase the text mining quality.

Those are the steps that are necessary in all text mining pre-processing. Fortunately, the **tm** package provides most of the functions and some others will only need some minor changes to work. However, for the particular experiment data, we will always make some unique pre-processing steps to clean the data [41]. Here are some
steps that need to be done for this experiment:

- **Removal of URLs.** In Wikipedia documents, URL are commonly used to link to another document or webpage. However in the LDA topic model, it is impossible to make use of a URL to exhibit topics. We will delete all the URLs in the documents.

- **Removal of attached files.** In Wikipedia documents, there are many attached files to illustrate the entry. They are either a figure or a file, and our model can process neither. Images and file processing are not considered in this thesis, so we will remove the attached files too. The attached file is expressed in a format of “[[File:filename]]”, which we can match and delete.

- **Removal of labels.** The text data we retrieved from XML file is with some labels for formatting. These labels will not affect the meanings and we will delete them as well.

- **Removal of other special words.** Besides stopwords, there are some particular special words in the documents. For example, “ndash” represents the dash in the document; “basestyle” is used in format of XML; “de” is a Spanish stopwords that sometimes can be found in the documents.

Those steps are unique in this experiment and no direct functions can be used. However, we can always make some minor changes on the current functions to finish these unique pre-processing steps. We have shown several steps of pre-processing and the order needs to be carefully arranged because it may affect the result. Some of the steps need to be done twice. In the pre-processing for this particular experiment, we will first transfer the documents to lower case and then remove the URLs, attached files and labels. Next we remove the stopwords and special words with punctuations. After that we remove the punctuation and numbers. Then we remove special words without punctuations. Finally we will remove the white spaces and start stemming. The order of these steps is also shown in Figure 3.2.

After words removal and transformation, the text data is clean and slim. Then it requires formatting as a document term matrix to train and fit the model.
3.4 Fitting Model

After data retrieving and pre-processing, now we have a document term matrix with clean data to fit the LDA topic model. The coding language is R and mainly use the R package \texttt{topicmodels} with its dependency package \texttt{tm}.

In the LDA topic model, the input data is the collection of documents and the number of topics $K$. Similar with k-means algorithm, as an unsupervised learning algorithm, the topics number $K$ has to be manually decided. In the early years of topic modelling, researchers empirically decided how many topics are going to be used in the model. In some certain projects, the number of topics is decided by the limitation of the application. However, as the algorithm having been developed, researchers have concluded a way to choose the number of topics in the model.

3.4.1 Model Selection

One of the most widely used metrics to measure the model performance is maximum likelihood function. In statistics, maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given data [36].

For example, if Acadia University would like to know the heights and weights of their male and female students, it is unable to measure the height of every single student in the whole university due to cost or time constraints. But we can assume the heights and weights are normally distributed with some unknown mean and variance,
the mean and variance can be estimated with MLE while only sampling a small amount of students’ heights and weights.

In general, for a fixed set of data and underlying statistical model, the method of maximum likelihood selects the set of values of the model parameters that maximizes the likelihood function as:

\[
L(\theta, D) = p(D|\theta) = \prod_{i=1}^{M} p(x_i|\theta)
\] (3.1)

where \(\theta\) is a vector of parameters and \(D = x_1, x_2, \ldots, x_m\) is the dataset whose data are independent and identically distributed. Generally, it is often more convenient to work with the logarithm of the likelihood function, called the log-likelihood:

\[
\log(D|\theta) = \sum_{i=1}^{M} p(x_i|\theta)
\] (3.2)

Estimation using Gibbs sampling requires specification of values for the parameters of the prior distributions. Drs. Griffiths and Steyvers suggest a value of \(50/k\) for \(\alpha\) and 0.1 for \(\beta\). Because the number of topics is in general not known, models with several different numbers of topics are fitted and the optimal number is determined in a data-driven way [20].

<table>
<thead>
<tr>
<th>Number of Topics</th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log-likelihood</td>
<td>-5596814</td>
<td>-5487432</td>
<td>-5502309</td>
<td>-5513969</td>
<td>-5554038</td>
<td>-5564250</td>
</tr>
</tbody>
</table>

The estimation of log-likelihood is embedded in the \texttt{LDA()} function of package \texttt{topicmodels}. When the \texttt{LDA()} function is called, the return value and its log-likelihood are calculated as well. So we run a set of topic numbers of 25, 50, 75, 100, 150, 200 to fit the LDA topic model and record each of the log-likelihood in the Table 3.2. Figure 3.3 shows the plot of the log-likelihood for each topic. Given the number of topics ranging from 25 to 200, we can see the figure hits the highest log-likelihood at the 50 topics and then slowly starts to go down. According to the log-likelihood, the best number of topics to fit this corpus is 50.
3.4. FITTING MODEL

3.4.2 Result

An LDA model of Simple Wikipedia with 2000 documents, returned after 2000 iterations of Gibbs sampling, with $K = 50$ topics, and Dirichlet hyperparameters $\beta = 0.1$ and $\alpha = 50/K$, which are also the default parameter settings in package topicmodels. The algorithm generated the 50 topics for this corpus, in Table 3.3, we can have a peek of 7 of 50 topics with their top 10 terms.

The 50 topics are generated by LDA topic model and the terms tend to have a similar meaning. The terms are clustering into 50 topics and the algorithm does not give the topics a label, but we can see there is a category to each topic. Sometimes, researchers or application developers would give a label to these topics. For example in Table 3.3, Topic 2 with terms like “music”, “vienna” and “beethoven”, can be labelled as “music”; Topic 17 with terms like “internet”, “program”, “web”, can be labelled as “IT”; Topic 33 with terms like “planet”, “earth”, “moon”, can be labelled as “outer space”. We can leave the topics as just numbers since it does not affect the algorithm and they are actually hidden in the structure.
Then we can apply posterior computation to calculate the topic distribution over each document. The distribution can be treated as topic proportions over documents. For example in Figure 3.4, the document from Simple Wikipedia called Biology, which is an introduction to the term of Biology. After posterior computation, the document can be processed to be a distribution of topics and barplot is shown in the figure. We can easily get Topic 24 with the highest proportion, and we can infer that Topic 24 is the one associated with biology or nature. The top 3 topics for this document are Topic 24, Topic 15, Topic 22. To testify the results, we list those 3 topics in Table 3.4.
3.5 Evaluations and Performance

Topic model is a relatively new trending field in machine learning. As Dr. Blei admitted, there is a disconnect between how topic models are evaluated and why we expect topic models to be useful. This is the model checking problem [6]. For unsupervised learning, there is always a problem for evaluating. Unlike supervised learning, the dataset has a label of either a real value or a category. The label is the standard to train the model. The evaluation of the model is to reach the label value or get as close as possible.

However in unsupervised learning, sometimes the dataset has an external indices, that works like supervised learning. The algorithm runs the dataset without labels as before, but the external indices will help the evaluation of model. For example in Section 2.1.2, the example with the clustering of iris species with sepals and petals is actually an unsupervised learning plus external indices evaluation. If there are no external indices, we have to check on the internal indices. Simply speaking, it usually assigns the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. These evaluation methods differ from algorithms and the actual dataset.
In the last section to fit the LDA topic model, I took advantage of the relatively small sample size of 2000 documents to test a sequence of topic numbers from 30 to 200 and calculate the log-likelihood to pick the best number of 50 topics. It is an efficient method but not for a large dataset, which is considered to have more than 10,000 documents. Typically topic models are evaluated in this way. First hold out a subset of the corpus as test set. Then apply a variety of topic models to the rest of the corpus and find the fit of the model to check on the test set. Finally choose the model to best fit the test set.

Rather than solving a difficult task for topic models, I prefer to look into a new angle for the problem. The ultimate goal for evaluation is to find the fit for the model and more importantly for the actual problem. So in this thesis, I will focus on the topic models performance for particular problems not on the model itself. On designing the experiments, I want the model to meet the real problems and measure the performance with human judgement. There are three experiments for three aspects for topic models - clustering documents, finding similar documents, and exploring corpus. Here is a description to the experiments design.

### 3.5.1 Clustering Documents

Clustering is a division of data into groups of similar objects [2]. LDA topic model can extract the hidden topics and analyse the document to a distribution over each topic. We can use this feature to cluster the documents. If we label the documents with the highest proportion topic, we can cluster the documents into 50 topics. For example, in the previous section, we have seen a barplot for document Biology to understand the structure into a distribution of topics rather than having to read it through. The highest proportion topic is Topic 24, which is a topic of biology. So we can actually label the document Biology to Topic 24, and this is a method for LDA topic model labelling or clustering document.
3.5.2 Finding Similar Documents

LDA topic model gives us a distribution over each topic, which can be treated as the proportions of each topic in this document. This is a view of topics that can be used to find the similar documents. Different than previous methods, which compare every word in the document, LDA topic model will compare the distribution on the topic level.

In this experiment, I use Jensen Shannon divergence to calculate the distance between two documents and find the most similar documents. To compare the performance, I also ask some volunteers to find the most similar documents by their own standard. In this way, we can compare the model with human judgement.

3.5.3 Exploring Corpus

In this experiment, I design a small application to explore the collection of documents with topic models. Traditionally, if we are looking for a particular document, we will type in some keywords, title words, or author names to find the documents. However, that is a result orientated search, which means you absolutely know what you are looking for, and the clues you type in can shrink the range and you will pick out the correct document. But in some cases, we are just looking for a direction or a topic. We do not have a particular answer to the search and more importantly, we can explore through the results. This often happens to researchers or students who are researching topics as reference or looking for some inspiration.

This application will use the results from LDA topic models to provide a topic explore tool. It will start with a set of topics, and you can dig into the topics to find more documents. For each document, it is also linked with similar topics and similar documents. It is a tool to explore through topics which take advantages of LDA topic model results and other algorithms are unable to do.
Chapter 4

Experiments

In this chapter, we will present three experiments using LDA topic model to cluster documents, find similar documents and explore corpus in a new application.

Clustering documents by traditional clustering method is only based on the terms in the documents which does not make sense to humans. LDA topic modelling can process the documents into topics and provide a better performance. Document similarity is a very useful attribute in practical projects. LDA topic model combined with Jensen-Shannon (JS) divergence can give a better performance to calculate the documents similarity which is the closest to human judgement. Finally, we will assemble the LDA topic model with the original documents and the clustering and similarity results to present an application which allows us to explore through topics and related documents.

4.1 Experiment 1: Clustering Documents

4.1.1 Objective

The objective of this experiment is to use LDA topic model to present a topic level clustering.
4.1.2 Method

The experiment data is the 2000 documents from Simple Wikipedia. The goal is to find the highest topic proportion in each document and mark the document belonging to that topic.

This method is not a traditional clustering method. Typical clustering algorithms, like k-means (see Section 2.1.2) or hierarchical clustering, measure the data by distance where the data are treated as a multi-dimensional point. The dataset is a matrix where the rows are the observations and the columns are the attributes. The close distance points tend to be clustered. The method is often used in clustering biological values such as fungus or leaves. If applied to documents clustering, we will treat each of the term as an attribute and transfer the documents to a term-document matrix. The clustering results are based on the terms used in the documents. But in real cases, lots of the words in different documents frequently appeared. Simply applying k-means or hierarchical clustering is not good enough if the documents share a lot of similar terms, and also their efficiency is low. For example, there are two documents titled “Chinese food” and “Basketball”, and they should be labelled in different clusters. However, if the two documents are focused more on how much they love the sports and food like “love”, “great”, but not many words actually about the sports and food, the traditional clustering method will tend to label them in one cluster.

However in LDA topic model, the method is different. After the LDA model is fitted, each document becomes a distribution over topics. The topics exhibits what the document is talking about and the topics can be treated as a label of a cluster. We can take advantage of this feature and choose the highest proportion topic to label this document. Compared to the traditional methods, rather than simply throwing in all the terms, LDA topic model first processes the documents to be topics, and then uses the topics to cluster. Still using the example of Chinese food and Basketball, even they express more love than actual sports and food terms, the LDA topic model will first label in a topic about love and second label them in a topic of food and topic of sports. The two option method gives us a more flexible way to fit in documents. Also it is a new way to cluster documents on the topic level which makes more sense
4.1. EXPERIMENT 1: CLUSTERING DOCUMENTS

4.1.3 Results

The results are based on the fitted model (from Section 3.4) with 50 topics. We assigned the highest proportion topic to the document and list all the documents into a table. Table 4.1 shows some of the results.

Table 4.1: A Sample of Documents Topic Clustering Results

<table>
<thead>
<tr>
<th>Titles</th>
<th>Topic(cluster)</th>
<th>Titles</th>
<th>Topic(cluster)</th>
</tr>
</thead>
<tbody>
<tr>
<td>April</td>
<td>Topic 13</td>
<td>Computer Science</td>
<td>Topic 17</td>
</tr>
<tr>
<td>August</td>
<td>Topic 13</td>
<td>Google</td>
<td>Topic 17</td>
</tr>
<tr>
<td>December</td>
<td>Topic 13</td>
<td>Apple Macintosh</td>
<td>Topic 17</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>Topic 14</td>
<td>Network</td>
<td>Topic 17</td>
</tr>
<tr>
<td>Addition</td>
<td>Topic 14</td>
<td>Galaxy</td>
<td>Topic 33</td>
</tr>
<tr>
<td>Brazil</td>
<td>Topic 47</td>
<td>Mars</td>
<td>Topic 33</td>
</tr>
<tr>
<td>China</td>
<td>Topic 47</td>
<td>Earth</td>
<td>Topic 33</td>
</tr>
<tr>
<td>Farm</td>
<td>Topic 10</td>
<td>Chinese</td>
<td>Topic 50</td>
</tr>
<tr>
<td>Breakfast Sausage</td>
<td>Topic 10</td>
<td>Grammar</td>
<td>Topic 50</td>
</tr>
<tr>
<td>Berry</td>
<td>Topic 10</td>
<td>Hydrogen</td>
<td>Topic 28</td>
</tr>
<tr>
<td>Fruit</td>
<td>Topic 10</td>
<td>Helium</td>
<td>Topic 28</td>
</tr>
</tbody>
</table>

If we take a look at Table 4.1, compared with the titles and the topics, we can find the clustering algorithm works quite well. There are two ways to evaluate the clustering algorithm - external evaluation and internal evaluation. The most commonly used is external evaluation, which is to compare the label of the dataset with the actual results. When collecting the data, it may have a given category. When fitting the model, the labels are not used. So it is still unsupervised learning, but the label will be used to evaluate the algorithm. Unfortunately, the text data that I downloaded from Wikipedia, does not have a label of their categories. So we can not compare the clustering results with the original data categories. Another way is internal evaluation, which uses the current results and values to judge the performance. This is not an ideal method and it differs from models. For example in k-means, the internal evaluation metrics includes connectivity, Dunn index and others. Basically speaking, the internal evaluation methods minimize the distance inside the cluster.
and maximize the distance among clusters. As I mentioned in Section 3.5, evaluation is always a task for unsupervised learning especially for topic models, so far there is no perfect solution for LDA topic model clustering evaluation. But according to the concept of the internal evaluation, LDA topic model clustering method is indeed based on their similarity because the cluster is actually the highest proportion topic. In terms of similarity evaluation, LDA topic model clustering is a good performance method. Other than that, I will not discuss further clustering evaluation problem in this thesis.

Figure 4.1 is a histogram of the documents topic clustering, which shows the number of documents in each cluster. In this figure, we can tell the number of documents in Topic 3 is way beyond other topics. This is because the terms in this topic is a little neutral among the topics. We list the top 30 terms in Topic 3, which is shown in Table 4.2. These terms have an ambiguous boundaries to belong to an area,
and meantime they are not stop words. For examples, the terms “people”, “like”, “make”, they could appear in any documents. The LDA will put them into a topic because in human language customs, they tend to have similar probabilities in some documents (similar as stopwords, but stopwords probabilities are more stable in every documents, and that is the reason why we can remove them). In pre-processing, it is reasonable to keep these terms because they actually have a meaning and removing them will affect the quality of some particular documents (for example, a document in Wikipedia titled “Like”, where the words are basically neutral to any topics).

It is fine to leave the clustering results as it is now. But in order to have a better result, we can work around the clustering to reduce the size of Topic 3 and spread the documents to other topics. The advantage of topic model is that it is a probabilistic model, and all the documents have multiple values on topic assignments. For all the documents belonging to Topic 3, we can add the second highest proportion as the second option or simply substitute all of them with the second option to be the cluster labels. Here in Table 4.3 listed with some of the documents in Topic 3 with their second option clusters.

The examples in Table 4.3 show that if we use the second option to be the label, it makes better sense than the first one. LDA topic model provides the proportions of 50 topics for each document. If we list the actual proportions of first two options, the clustering philosophy will be clearer.

Table 4.4 lists the highest three proportions and the rest proportion of the document from Table 4.3. We can tell the highest two topics (50 topics in total) are quite dominating in the proportion, which are at least 30% and a few are over 50%. So
Table 4.3: Clustering with Two Options

<table>
<thead>
<tr>
<th>Title</th>
<th>First Option</th>
<th>Second Option</th>
<th>Top Terms in Second Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviation</td>
<td>Topic 3</td>
<td>Topic 50</td>
<td>language, english, word</td>
</tr>
<tr>
<td>Application</td>
<td>Topic 3</td>
<td>Topic 17</td>
<td>computability, internet, system</td>
</tr>
<tr>
<td>Coin</td>
<td>Topic 3</td>
<td>Topic 34</td>
<td>money, econometrics, companies</td>
</tr>
<tr>
<td>Human Body</td>
<td>Topic 3</td>
<td>Topic 24</td>
<td>cell, body, disease, organ</td>
</tr>
<tr>
<td>Harbour</td>
<td>Topic 3</td>
<td>Topic 27</td>
<td>engine, ship, boot, make</td>
</tr>
<tr>
<td>IELTS</td>
<td>Topic 3</td>
<td>Topic 50</td>
<td>language, english, word</td>
</tr>
<tr>
<td>Nature</td>
<td>Topic 3</td>
<td>Topic 15</td>
<td>universe, science, study, book</td>
</tr>
<tr>
<td>Statistics</td>
<td>Topic 3</td>
<td>Topic 14</td>
<td>number, mathematic, line</td>
</tr>
</tbody>
</table>

Table 4.4: Clustering Proportions of Two Options

<table>
<thead>
<tr>
<th>Title</th>
<th>First Option</th>
<th>Second Option</th>
<th>Third Option</th>
<th>The Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviation</td>
<td>0.299</td>
<td>0.229</td>
<td>0.045</td>
<td>0.427</td>
</tr>
<tr>
<td>Application</td>
<td>0.175</td>
<td>0.143</td>
<td>0.071</td>
<td>0.611</td>
</tr>
<tr>
<td>Coin</td>
<td>0.243</td>
<td>0.126</td>
<td>0.045</td>
<td>0.586</td>
</tr>
<tr>
<td>Human Body</td>
<td>0.332</td>
<td>0.297</td>
<td>0.123</td>
<td>0.248</td>
</tr>
<tr>
<td>Harbour</td>
<td>0.166</td>
<td>0.146</td>
<td>0.119</td>
<td>0.569</td>
</tr>
<tr>
<td>IELTS</td>
<td>0.205</td>
<td>0.098</td>
<td>0.066</td>
<td>0.631</td>
</tr>
<tr>
<td>Nature</td>
<td>0.329</td>
<td>0.185</td>
<td>0.068</td>
<td>0.418</td>
</tr>
<tr>
<td>Statistics</td>
<td>0.477</td>
<td>0.323</td>
<td>0.035</td>
<td>0.165</td>
</tr>
</tbody>
</table>

the highest two proportions can represent the document topic. In this case, for some documents originally labelled in Topic 3, we can use the second option to make a sub-label or replace as the first option. Also, the document in the table titled “Harbour” with highest three dominated topics, which is also reasonable to be labelled with 3 options. Please note that, theoretically, we can divide two or three options as we want, but practically speaking, the number of options has to depend on the particular application and the its requirements.

In the end, no matter which way we cluster, the LDA topic modelling can provide all 50 proportions that scale the document in terms of topics and the performance of the clustering is quite convincing.
4.2 Experiment 2: Finding Similar Documents

4.2.1 Objective

The objective of this experiment is to use the LDA topic model to find the most similar documents given a particular document.

4.2.2 Method

The experiment data is still the 2000 documents from Simple Wikipedia. After fitted to the LDA topic model, the documents are processed to be a vector of 50 topic proportions, which sum to be one. In terms of probabilistic model, it can also be treated as a distribution over topics. So we can get the similarity between two documents via calculating the distance between the two distributions.

Similarity is the state or fact of being similar while similar is referring to a resemblance in appearance, character or quantity, without being identical [14]. However in LDA topic modelling, the similarity is more of topic similarity rather than the words only. Before topic models, the document similarity was more based on titles, keywords or abstracts. First of all, processing the titles and keywords runs more quickly than processing the whole document. Secondly, processing the whole document involves a lot of neural terms that can distract the results from the actual topics. In the light level text mining, processing titles and keywords instead of the whole documents is quite efficient and easy. But in some cases such as the title which contains very less information or the contents which covers several topics, these methods perform poorly in text mining.

Topic models, which may come in different mathematical methods, process the documents into topics first. Even with lots of neural terms, the actual contents can still be presented in a proportion of topics. After that, we can use a mathematical expression to measure the proportion and calculate the similarity. In this way, we process the whole documents with no information discarded (we actually discarded stopwords and numbers, but they contain very little useful information), and it also provides results on the topic perspective which makes more sense to people.
In this case, the mathematical method to calculate the similarity is JS divergence (detail see Section 2.5), which can calculate the JS distance between distributions. Given a document, we will simply calculate the JS distance between any other documents and put the results into a vector. Sorting this vector by increasing order, we can get the most similar to the least similar ones for this given document.

4.2.3 Evaluation

Evaluation for finding similar documents is a tricky task, because there is no absolute answer to the question. Every person may hold their unique standard for which document is the most similar to another. Even in a general view of this question, where a majority of people may have the same answer, their reasons differ from each other. Because after all, it is a subjective question. But in my opinion, there is a fuzzy answer to this question - topics.

If the documents are talking about the same topics, they might be similar documents. This is one of the reasons why topic model can provide better results on finding similar documents. But in terms of evaluation by a certain metric, it is still impossible to find a way to compare the performance where given a document finding the most similar ones, or whether document A is a better answer than document B. Most papers will stop here, with the results listed or use the words like “good” and “better” to describe the subjective performance.

I would like to do something interesting in this part. Since there is no standard, how about using human judgement to get a majority answer? So I made a questionnaire to ask a small group of people to rate the most similar documents for a given one, and compare the results with the algorithm generated. This way we can give a performance evaluation of how the LDA topic model matches human judgement.

The questionnaire has a structure similar shown in Figure 4.2. The readers are asked to read the given document and then identify the first and second most similar documents from following documents A, B, C and D. The four documents in the chosen group consists of two of the most similar documents from the LDA topic model, one from vector space model, and one of the human picked distraction option. The
4.2. EXPERIMENT 2: FINDING SIMILAR DOCUMENTS

distractive document is still topic related but not in the top five similar documents in either of the algorithm. There are five groups of these documents like this in Figure 4.2.

Figure 4.2: Questionnaire Structure

For example, the given document is titled Constitution, from which comes four following documents - (A) Head of State, (B) Regime, (C) Freedom of Speech and (D) Citizenship. The readers are asked to read the given document Constitution first, then choose the first and second most similar documents from the four options.

4.2.4 Results

The results are generated by calculating the JS distance between every two documents. So in this section, we will take the document Constitution as an example to explain.
First, Figure 4.3 is the barplot of the topic distribution of Document Constitution, where we have a distribution of topic proportions of this document and they sum to be 1. Next, we calculate the JS distance (see Section 2.5) between Document Constitution and every other document. Then we sort the results to be a list of the increasing order of the documents with the closest JS distance. The top 8 documents from the result are shown in Table 4.5, and their titles are listed in Table 4.6.

Table 4.5: Top 8 Documents with Closest JS Distance to Document Constitution

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JS Distance</td>
<td>0.0000</td>
<td>0.1404</td>
<td>0.1823</td>
<td>0.1828</td>
</tr>
<tr>
<td>JS Distance</td>
<td>0.2018</td>
<td>0.2043</td>
<td>0.2044</td>
<td>0.2070</td>
</tr>
</tbody>
</table>

In those tables we can find that the most related document to Constitution, whose closest JS distance is 0, which is Constitution itself. Besides that, the most related documents are Freedom of Speech, Head of State and so on.
4.2. EXPERIMENT 2: FINDING SIMILAR DOCUMENTS

Table 4.6: Top 8 Documents Titles to Document Constitution

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Title</th>
<th>Document ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document 99</td>
<td>Constitution</td>
<td>Document 1016</td>
<td>Election</td>
</tr>
<tr>
<td>Document 983</td>
<td>Freedom of Speech</td>
<td>Document 1512</td>
<td>Republic</td>
</tr>
<tr>
<td>Document 1473</td>
<td>Head of State</td>
<td>Document 422</td>
<td>Time limit</td>
</tr>
<tr>
<td>Document 897</td>
<td>Citizenship</td>
<td>Document 360</td>
<td>Regime</td>
</tr>
</tbody>
</table>

So how are the results? I did a questionnaire documents of five groups of documents on a reading club called Jackets. Since the experiment requires readers to read and understand the documents, then make a decision on which are most related, the participants have to be willing to read 25 documents and able to make a relatively correct answer. So I asked my dear friend Ms. Erma Stultz and her reading club Jackets for help. Jackets is a reading club of 10 reading lovers who each holds a meeting once a month to read a book and share opinions. They love reading and are able to well understand the contents. So they are the ideal participants for this questionnaire. In addition to Constitution, the other four documents in the chosen group are Freedom of Speech, Head of State, Citizenship and Regime. In table 4.7 shows the results from the questionnaires.

Table 4.7: Questionnaire Results of Document Constitution

<table>
<thead>
<tr>
<th></th>
<th>Most Related</th>
<th>Second Most Related</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>Freedom of Speech</td>
<td>Head of State</td>
</tr>
<tr>
<td>Reader 1</td>
<td>Freedom of Speech</td>
<td>Head of State</td>
</tr>
<tr>
<td>Reader 2</td>
<td>Head of State</td>
<td>Regime</td>
</tr>
<tr>
<td>Reader 3</td>
<td>Freedom of Speech</td>
<td>Regime</td>
</tr>
<tr>
<td>Reader 4</td>
<td>Freedom of Speech</td>
<td>Head of State</td>
</tr>
<tr>
<td>Reader 5</td>
<td>Citizenship</td>
<td>Freedom of Speech</td>
</tr>
<tr>
<td>Reader 6</td>
<td>Freedom of Speech</td>
<td>Regime</td>
</tr>
<tr>
<td>Reader 7</td>
<td>Freedom of Speech</td>
<td>Citizenship</td>
</tr>
<tr>
<td>Reader 8</td>
<td>Head of State</td>
<td>Regime</td>
</tr>
<tr>
<td>Reader 9</td>
<td>Freedom of Speech</td>
<td>Head of State</td>
</tr>
<tr>
<td>Reader 10</td>
<td>Citizenship</td>
<td>Head of State</td>
</tr>
</tbody>
</table>

In table 4.8, the most related documents ranked by LDA are Freedom of Speech and Head of State. The table also shows the results from topic model are quite similar
with the 10 readers’. Here we list the first and second choice from the readers. In the questionnaire we actually ask them to sort all four documents if possible. Then we set up a scoring method that a first choice document earns 4 points, a second choice earns 3 point, a third choice earns 2 points and a fourth choice earns 1 point, which is similar to MVP voting method in sports. It is hard to find an order from 10 readers, so we use the voting method to get a trend from all the answers. For example, there are three people sorting documents as ACBD, ABCD and ACBD, then A gets $4 + 4 + 4 = 12$ points, B gets $2 + 3 + 2 = 7$ points, C gets $3 + 2 + 3 = 8$ points and D gets $1 + 1 + 1 = 3$ points. If we follow this method, the most and second most similar documents by human judgement is Freedom of Speech and Head of State, which matches the LDA topic model. If we apply the method to all the five documents in the questionnaire, the results are shown in Table 4.8.

Table 4.8: The Scores of Questionnaire Documents

<table>
<thead>
<tr>
<th>Titles</th>
<th>Health</th>
<th>Molecule</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA Topic Model</td>
<td>Physical Exercise, Medicine</td>
<td>Organic Compound, Polymer</td>
</tr>
<tr>
<td>Human Judgement</td>
<td>Physical Exercise, Medicine</td>
<td>Polymer, Organic Compound</td>
</tr>
<tr>
<td>Titles</td>
<td>Roman</td>
<td>Euro</td>
</tr>
<tr>
<td>LDA Topic Model</td>
<td>9(year of 9 AD), Naples</td>
<td>Dollar, GDP</td>
</tr>
<tr>
<td>Human Judgement</td>
<td>9(year of 9 AD), Denarius</td>
<td>Dollar, Continent</td>
</tr>
<tr>
<td>Titles</td>
<td>Constitution</td>
<td></td>
</tr>
<tr>
<td>LDA Topic Model</td>
<td>Freedom of Speech, Head of State</td>
<td></td>
</tr>
<tr>
<td>Human Judgement</td>
<td>Freedom of Speech, Head of State</td>
<td></td>
</tr>
</tbody>
</table>

From the results of the questionnaire, we can see the LDA topic model got four groups correct on the first choice and two both correct groups (Constitution and Health). Generally speaking, the results can provide a good estimation of a list of similar documents, and it has a good probability on getting the most similar documents from human judgement. If we provide a list of similar documents for a given one, the LDA topic model will work better on predicting people’s choice. Normally there is no correct answer, the “human judgement” is just an average on the weights of people’s choice so we can get a general trend on the most similar document.
Admittedly, due to the limitations, we could not collect a larger group of people on the questionnaire. But the results are still convincing and the method is transferable. If we can have a larger group of people and more documents in LDA topic model, the comparison will be more valuable.

4.3 Experiment 3: Exploring Corpus

4.3.1 Objective

The objective of this experiment is to propose a document exploring system based on LDA topic modelling where documents are categorized by topics and linked with related ones. It is designed for researchers and students who would prefer to explore and scan for documents by certain topics rather than search for a particular document by title or keywords.

4.3.2 Method

To design such an exploring system, we have to use the fitted model with related analysis results including clustering and similar documents lists. Figure 4.4 is the structure of the system, which consists of 4 steps:

- Fitting the model. The first step is to train the raw text data into the LDA topic model, including pre-processing the raw data to clean format, fitting the model with 50 topics and posterior calculation to get documents topic proportions. It basically contains all the steps in Chapter 3.

- Documents clustering. The second step is to cluster the documents. All the documents will be clustered by the first topic option to 50 groups. Then we will apply second topic option to some of the large clusters like Topic 3.

- Similar documents lists. The third step is to calculate the JS distance between every two documents. Each document will be linked with the increasing order of the similar documents.
• Assemble in HTML files. Final step is to combine all the results to a set of HTML files as the system. It will start with the topic groups that users are able to scan the topics to enter. When a certain topic is selected, it will show all the documents in this cluster. After selecting a particular document in the cluster, the page will display the contents, followed by its related topics and documents.
4.3. EXPERIMENT 3: EXPLORING CORPUS

4.3.3 Results

The documents exploring system is based on HTML. In terms of displaying the function of exploring the Wikipedia corpus and a prototype of applying LDA topic models, I will not put too much effort on the appearance of the web pages, but focus more on the idea of exploring documents through topics.

![Figure 4.5: 3 Layers of System Webpage Design](image)

The system is mainly divided into 3 layers shown in Figure 4.5. The first layer is the index page, which is a list of all the topics; the second layer is the topic page, which shows a particular topic and its related documents; the third layer is the document page, which shows a particular document and its related documents. The first and second layer are linked by topics, the second and the third layer are linked by documents.

The screenshot of first layer index page are shown in Figure 4.6, where the system starts with a list of 50 topics. Each topic contains the top five terms to let
the users know what they are mainly about. The length of the screenshot only shows the top 16 topics. But in the actual webpage, it contains all 50 topics. Users can scan every topic and find what they are interested in and these topics are the entries to each document.

To better explain how to use the system and its advantages, let us start with an example. Suppose now we are going to do a project in the university about astronomy, and we are going to research some articles about a star, or our solar system on Wikipedia. We do not have a particular target on which star or what astronomic phenomenon to illustrate in this project. We just want to research and read through some options and find some inspiration. First we will read through the
4.3. EXPERIMENT 3: EXPLORING CORPUS

...topic list, and we will find Topic 33 with terms of “planet”, “earth” and “moon”, which is the one we are looking for (see Figure 4.7). So we can click the link to open the webpage of Topic 33.

Figure 4.7: Screenshot of System Index Webpage (2)

This leads us to the second layer – topic page. The topic page is shown with the documents that clustered into this topic. In this case, all the documents shown in Figure 4.8 are clustered in Topic 33. On the left side, it also provides the top 12 terms in Topic 33. In other words, we can find all the documents that related to Topic 33 which are mainly about astronomy or space. If we take a look at the related documents list on the right, we will find plenty of documents that are potentially related to our project such as Asteroid, Comet, Earth, Jupiter and so on. Now we can pick one of the document to read. Here I pick Mars, which is in the middle of
the list, right below Mercury (planet).

![Screenshot of the Webpage of Topic 33](image)

<table>
<thead>
<tr>
<th>Topic 33</th>
<th>Related Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>planet</td>
<td>Astronomy</td>
</tr>
<tr>
<td>earth</td>
<td>Asteroid</td>
</tr>
<tr>
<td>moon</td>
<td>Comet</td>
</tr>
<tr>
<td>star</td>
<td>Earth</td>
</tr>
<tr>
<td>sun</td>
<td>Galaxy</td>
</tr>
<tr>
<td>system</td>
<td>Jupiter</td>
</tr>
<tr>
<td>galaxia</td>
<td>Milky Way</td>
</tr>
<tr>
<td>solar</td>
<td>Mercury (planet)</td>
</tr>
<tr>
<td>ring</td>
<td>Mars</td>
</tr>
<tr>
<td>saturn</td>
<td>Neptune</td>
</tr>
<tr>
<td>jupiter</td>
<td>Planet</td>
</tr>
<tr>
<td>mar</td>
<td>Solar System</td>
</tr>
<tr>
<td></td>
<td>Saturn</td>
</tr>
<tr>
<td></td>
<td>Uranus</td>
</tr>
<tr>
<td></td>
<td>Venus</td>
</tr>
<tr>
<td></td>
<td>Mercury</td>
</tr>
</tbody>
</table>

Figure 4.8: Screenshot of the Webpage of Topic 33

After clicking “Mars”, we are at the third layer - document page. Figure 4.9 shows the screenshot of the document webpage of Mars. On the left side, there are the three most related topics to Mars which you can expand to other areas. On the right side, there are the top 20 most related articles to Mars including Venus, Earth and so on. To read the contents of Mars, we can click the title “Mars” that links to the Simple Wikipedia webpage (see Figure 4.10). The related topics and documents enable us to explore through the corpus and switch between the topics. We can find and collect all sorts of information between Mars, Earth and other planets to enrich our project contents. This is only built on a 2000 documents corpus. If we have a
4.3. EXPERIMENT 3: EXPLORING CORPUS

much larger knowledge base, such as a publisher documents set, the system can make more powerful impact on exploring and researching documents.

![Image of Document Mars webpage]

**Figure 4.9: Screenshot of the Webpage of Document Mars**

### Related Topics

<table>
<thead>
<tr>
<th>Topic 33</th>
<th>Topic 3</th>
<th>Topic 36</th>
</tr>
</thead>
<tbody>
<tr>
<td>earth</td>
<td>use</td>
<td>lake</td>
</tr>
<tr>
<td>moon</td>
<td>can</td>
<td>water</td>
</tr>
<tr>
<td>star</td>
<td>call</td>
<td>ocean</td>
</tr>
<tr>
<td>sun</td>
<td>also</td>
<td>mountain</td>
</tr>
<tr>
<td>system</td>
<td>mania</td>
<td>state</td>
</tr>
<tr>
<td>galaxia</td>
<td>one</td>
<td>north</td>
</tr>
<tr>
<td>solar</td>
<td>make</td>
<td>sea</td>
</tr>
<tr>
<td>ring</td>
<td>differ</td>
<td>categorical</td>
</tr>
<tr>
<td>saturn</td>
<td>like</td>
<td>south</td>
</tr>
<tr>
<td>jupiter</td>
<td>time</td>
<td>ice</td>
</tr>
<tr>
<td>mar</td>
<td>way</td>
<td>fall</td>
</tr>
<tr>
<td>space</td>
<td>mean</td>
<td>air</td>
</tr>
</tbody>
</table>

### Related Documents

- Venus
- Earth
- Greenhouse effect
- Atmosphere of Earth
- Outer space
- Solar System
- Satellite (natural)
- Saturn
- Wind
- Mercury (planet)
- Asteroid
- Meteor shower
- Jupiter
- Telescope
- White dwarf
- List of planets
- List of comets
- Asteroid belt
- Phases of the Moon

**4.3.4 Summary**

The exploring system combines both the results from previous experiments and the fitted model. The model generated 50 topics which are used in the first layer as index...
The system can be used as a managing system for libraries or publishers. It also provides the document searching via topics, especially for students and researchers when no certain results are required but browsing related contents.

Due to the limitation of time and my personal skills, the system is a prototype with a small sample size. However, the results are convincing and usable to the goal of applying LDA topic model and displaying the usage and advantages of the model.
Chapter 5

Conclusions and Future Work

5.1 Conclusions

Organizing and understanding millions of documents, papers and other text information becomes an arduous task for researchers and publishers. As machine learning techniques have developed quickly and are widely used, we did a text mining research based on LDA topic model and proposed a documents exploring system using this method. LDA topic model is a probabilistic model using Latent Dirichlet allocation, which processes text documents and assigns data into topics. Compared to other document processing methods, the difference is that processing millions of documents to topics are easier to access, categorize and compare. Also most importantly, topics are much easier to understand.

In the thesis, we briefly talked about the background knowledge of LDA topic model and its working principles. Then we explained in depth how to apply LDA topic model to text corpus by doing experiments on Simple Wikipedia documents. The experiments include all necessary steps of data retrieving, pre-processing, fitting the model and evaluations. The result of the experiments shows LDA topic model working effectively on document clustering and finding similar documents. Based on LDA topic model, we proposed a document exploring system which allows users to organize and explore the documents by topics where related documents are easier to find and access.
5.2 Limitations

Though we used considerable effort and made detailed designs, the thesis still has a few limitations:

- With the limited computing power and little budget, the number of experiment document samples could not reach over 2000. Even with 2000 documents, the pre-processing and training time still exceeds 36 hours, which is a time and effort consuming project.

- Due to the limitation of and the early development of topic model, there is no perfect evaluation methods so far. Even Dr. Blei, who proposed LDA topic model, pointed out that “the future direction of topic model should be a method for evaluation” [6].

- The images and links in the documents are ignored in the pre-processing step. If combined with other methods to process all of the information, the results could be better.

- The documents exploring system is coded under a static topic model, which means it cannot be fed in with new documents.

- The LDA topic model is only working in English, and possibly in French or some other languages whose basic unit is a word. However for Chinese or Japanese, whose basic unit is a character, we are not able to apply topic models.

5.3 Future Work

Based on those limitations and my understandings of the research, a few thoughts could merit the future research and experiments:

- From the perspective of this research, upgrading the computing power and enlarging the sample database on Wikipedia or even a collection of scientific papers could be a very good future project. Wikipedia or papers are an actual
knowledge base. They are well worth mining and the results could be the key elements to build an super intelligent robot. This is one of the reasons why we chose Simple Wikipedia documents to be the sample in this research.

- Applying data mining techniques to social network is a hot new direction. At the beginning of the research, we also trained the LDA topic model for Twitter data. We published a paper on the proceeding of the Sixth International Conference on Computer Science, Engineering and Information Technology (CCSEIT 2016) [37]. But considering the length of the thesis and the limitation of the time, we decided to exclude this part in the thesis. Social network data mining is not only important for computer science, but also for social science and business. It is a great future project or research for the topic.

- In terms of topic models, a well designed and widely approved evaluation method is one of the most significant directions of the research.

- Some data mining scientists have already made progress on topic model for image processing. If further development is achieved, the topic model could be a complete package of document processing.
Bibliography


