

STATUS OF THESIS

Title of thesis Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA)

I WAN MOHAMMAD AFLAH BIN MOHAMMAD ZUBIR

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

Confidential

Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

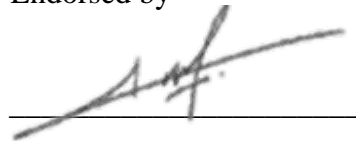


Signature of Author

Permanent address: MR2-13-06,
Sri Acappella Serviced Apartment, 1
Jalan Lompat Tinggi 13/33, Seksyen
13, Shah Alam, Selangor

Date: 03/08/2022

Endorsed by



Signature of Supervisor

Name of Supervisor
Associate Professor Dr. Izzatdin
Abdul Aziz

Date: 03/08/2022

UNIVERSITI TEKNOLOGI PETRONAS

ADAPTIVE SELECTION OF INFERENCE METHOD FOR LATENT DIRICHLET
ALLOCATION (ASIM-LDA)

By

WAN MOHAMMAD AFLAH BIN MOHAMMD ZUBIR

The undersigned certify that they have read and recommend to the Postgraduate Studies Programme for acceptance of this thesis for the fulfilment of the requirements for the degree stated.

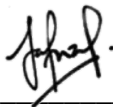
Signature:



Main Supervisor:

Associate Professor Dr. Izzatdin Abdul Aziz

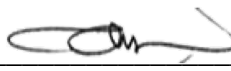
Signature:



Co-Supervisor:

Ts Dr Jafreezal Jaafar
Asse. Prof. Dean
Faculty of Science and Information Technology
Universiti Teknologi PETRONAS
Associate Professor Dr. Jafreezal Jaafar

Signature:



Ts. Dr. ALIZA SARLAN
Chair
Computer and Information Sciences
Universiti Teknologi PETRONAS

Head of Department:

Dr. Aliza Sarlan

Date:

12 August 2022

ADAPTIVE SELECTION OF INFERENCE METHOD FOR LATENT DIRICHLET
ALLOCATION (ASIM-LDA)

by

WAN MOHAMMAD AFLAH BIN MOHAMMAD ZUBIR

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR

PERAK

AUGUST 2022

DECLARATION OF THESIS

Title of thesis

Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA)

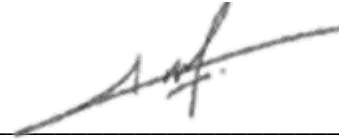
I WAN MOHAMMAD AFLAH BIN MOHAMMAD ZUBIR

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by



Signature of Author



Signature of Supervisor

Permanent address: MR2-13-06,
Sri Acappella Serviced Apartment,
1 Jalan Lompat Tinggi 13/33,
Seksyen 13, Shah Alam, Selangor

Name of Supervisor
Associate Professor Dr.
Izzatdin Abdul Aziz

Date : 03/08/2022

Date : 03/08/2022

CONFIRMATION BY PANEL OF EXAMINERS

I certify that a panel of examiners have met on 17 May 2022 to conduct the final examination of Wan Mohammad Aflah Bin Mohammad Zubir (G03592) on his thesis entitled "Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA)". The panel of examiners recommends that the student be awarded the MSc in Information Technology. The members of the panel of examiners were as follows:

Dr. Dhanapal Durai Dominic
Associate Professor
Computer and Information Science
Universiti Teknologi PETRONAS
(Chairman)

Ts. Dr. Mohd Hilmi B Hasan
Senior Lecturer
Computer and Information Science
Universiti Teknologi PETRONAS
(Internal Examiner)

Ts. Dr. Rosilah Hassan
Associate Professor
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
(External Examiner)

Assoc Prof Dr Balbir Singh Mahinder Singh
Dean
Centre for Graduates Studies
Universiti Teknologi PETRONAS
Date:

DEDICATION

I dedicate this work to

My inspiring parents:

My beloved father, Mohammad Zubir and My beloved mother, Sharifah Azhariah

My loving wife:

Farah Hanissa

My sister and brother:

Dr Wan Syahirah and Wan Muhammad Afif

And my relentless supervisors:

AP Dr Izzatdin Abdul Aziz and AP Dr Jafreezal Jaafar

I could not have completed this journey without their continuous support and prayers.

ACKNOWLEDGEMENTS

I am grateful to Allah, the most gracious, for giving me the strength and the ability to successfully complete this work.

First, I would like to express my deepest gratitude to my supervisor, AP Dr Izzatdin Abdul Aziz, who has given me the best guidance a supervisor could have ever given me. He has supported me throughout my journey with his patience and tenacity for me to complete successfully. I could not have done it without his support. I would like to also thank my co-supervisor AP Dr Jafreezal Jaafar for his knowledge and support in my journey to complete this research.

Next, I wish to thank Universiti Teknologi PETRONAS, specifically Computer and Information Sciences Department and the Centre for Graduate Studies for all the necessary support for this research. I am also indebted to the members of the Centre for Research in Data Science (CERDAS). Thank you for the technical guidance and your friendship.

Lastly, I would like to thank my wife and my family, the most important people in my life. Their endless support, patience and love motivated me to complete this study. It would not have been possible to complete my research if I did not have their support.

ABSTRACT

A lot of textual data is generated daily due to the advent of technologies, such as social media networks. Characteristics of textual data introduces challenges in analysing the data such as selecting a suitable text representation method for varying complexity of dataset. Topic Modelling is a research area that focuses in addressing this issue, by establishing the assumption that textual data are clustered in topics, rather than simply independent words. Latent Dirichlet Allocation (LDA) is one of the method in Topic Modelling. LDA is a generative probabilistic method, which allows LDA to adapt to new and unseen data, without having to retrain the model on the entire dataset. To extract topic representations, LDA uses approximate inference algorithms, such as Variational Bayesian Inference (VB) and Gibbs Sampling (GS). These two inference algorithms are selected due to their high performance in extracting quality topic distributions. Each of the inference algorithm adapts differently to different complexity dataset. The inference algorithms also have hyperparameters which need to be tuned to increase fitness to a dataset. To address these two challenges, Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA) is proposed. The objective of this research is to overcome the challenge of adapting to varying complexity of dataset by introducing two stages of refinement, namely hyperparameter optimization of individual inference algorithm and establish selection filter of best inference algorithm based on topic coherence score. The proposed algorithm is tested on three textual datasets. It is then evaluated based on performance of classification task. In this evaluation, three parameters of classification performance which are, accuracy, precision, and recall, are analysed. Based on the results, it shows that the proposed method accomplishes improvement of average of 9% in comparison to other topic modelling algorithms, despite being tested on different levels of dataset complexity. ASIM-LDA adapts through the two stages of refinement, which effectively selects the best approximate inference algorithms and best set of hyperparameters for given dataset.

ABSTRAK

Banyak data teks dihasilkan setiap hari dengan kehadiran teknologi, seperti media sosial. Ciri-ciri data telah menghasilkan banyak halangan dalam menganalisa data, seperti memastikan data mudah difahami oleh komputer. Selain itu, terdapat cabaran untuk ekstrak maksud konteks dari data. Topic Modelling ialah bidang penyelidikan yang bertumpu untuk menangani masalah ini, dengan membuat anggapan bahawa data dikelompokkan dalam topik, bukan sekadar perkataan perseorangan. Salah satu algoritma Topic Modelling ialah Latent Dirichlet Allocation (LDA). Ciri LDA yang merupakan algoritma kebarangkalian, membolehkan LDA menyesuaikan diri dengan data baru tanpa perlu melatih model di seluruh data. Untuk mengekstrak topik, LDA menggunakan algoritma inferensi anggaran seperti Variational Bayesian Inference (VB) dan Gibbs Sampling (GS). Kedua-dua algoritma inferensi ini dipilih disebabkan prestasi yang tinggi dalam mengekstrak taburan topik yang berkualiti. Setiap algoritma inferensi mempunyai prestasi yang berbeza mengikut kerumitan data yang berbeza. Algoritma inferensi juga mempunyai hyperparameter yang perlu diubah suai untuk meningkatkan prestasi. Untuk mengatasi cabaran ini, kajian ini mencadangkan Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA). Objektif penyelidikan ini adalah untuk mengatasi cabaran untuk menyesuaikan LDA untuk kerumitan data yang berbeza-beza dengan memperkenalkan dua tahap penyempurnaan, iaitu pengoptimuman hyperparameter algoritma dan menetapkan pemilihan algoritma inferensi terbaik berdasarkan Topic Coherence Score. Algoritma yang dicadangkan diuji pada tiga set data. Ia kemudian dinilai berdasarkan prestasi tugas klasifikasi. Dalam penilaian ini, tiga ukuran prestasi klasifikasi, accuracy, precision, and recall, dianalisa. Hasil kajian telah menunjukkan bahawa kaedah yang dicadangkan mencapai prestasi yang lebih tinggi dalam purata 9% dibandingkan dengan algoritma pemodelan topik lain, walaupun diuji pada tahap kerumitan kumpulan data yang berlainan. ASIM-LDA mampu meningkatkan kemampuan menyesuaikan LDA terhadap kerumitan kumpulan data yang berbeza-beza.

In compliance with the terms of the Copyright Act 1987 and the IP Policy of the university, the copyright of this thesis has been reassigned by the author to the legal entity of the university,

Institute of Technology PETRONAS Sdn Bhd.

Due acknowledgement shall always be made of the use of any material contained in, or derived from, this thesis.

© Wan Mohammad Aflah bin Mohammad Zubir, 2022

Institute of Technology PETRONAS Sdn Bhd

All rights reserved.

TABLE OF CONTENT

ABSTRACT.....	VII
ABSTRAK.....	VIII
TABLE OF CONTENT.....	X
LIST OF FIGURES	XIII
LIST OF TABLES.....	XIV
LIST OF ABBREVIATIONS.....	XV
CHAPTER 1 INTRODUCTION	1
1.1 Research Background.....	1
1.2 Research Motivation.....	8
1.3 Problem Statement.....	9
1.4 Research Questions.....	10
1.5 Research Objectives.....	10
1.6 Scope of the Research.....	11
1.7 Significance of Study.....	12
1.8 Thesis Organization	14
CHAPTER 2 LITERATURE REVIEW	15
2.1 Introduction.....	15
2.2 Topic Modelling Methods	15
2.3 Latent Dirichlet Allocation	19
2.4 Existing Works in Latent Dirichlet Allocation.....	21
2.5 Inference in Latent Dirichlet Allocation.....	23
2.6 Approximate Inference Algorithms for LDA	24
2.6.1 Variational Bayesian Inference	27
2.6.2 Gibbs Sampling	29
2.6.3 Hyperparameters in LDA	32
2.7 Hyperparameter Tuning Approaches.....	33
2.8 Objective Function for Latent Dirichlet Allocation.....	36
2.9 Summary.....	38
CHAPTER 3 RESEARCH METHODOLOGY	39
3.1 Introduction.....	39

3.2 Research Framework	39
3.3 Overview of Adaptive Selection of Inference Method for Latent Dirichlet Allocation	42
3.4 Design of ASIM-LDA	44
3.4.1 First Stage: Hyperparameter Optimization of Individual Inference Algorithms	48
3.4.2 Second Stage: Selection Filter for Best Instance of Inference Algorithms	55
3.4.3 Data Pre-Processing	57
3.4.4 Stop Words Removal	57
3.4.5 Stemming of the Words.....	58
3.5 Implementation of ASIM-LDA	59
3.5.1 Dataset	60
3.5.2 Data Pre-processing.....	64
3.5.3 ASIM-LDA	66
3.6 Significance of ASIM-LDA	72
3.7 Summary.....	72
CHAPTER 4 RESULTS AND DISCUSSION.....	74
4.1 Introduction.....	74
4.2 Experiment Set-Up	75
4.2.1 Evaluation Method – Performance in Classification Task.....	76
4.2.2 Scope of Experimentation	79
4.2.3 Platform.....	79
4.3 Results on Accuracy, Precision and Recall on Classification Task.....	80
4.3.1 Accuracy Analysis of ASIM-LDA on Classification Task.....	80
4.3.2 Precision Analysis of ASIM-LDA on Classification Task	82
4.3.3 Recall Analysis of ASIM-LDA on Classification Task	84
4.4 Summary.....	86
CHAPTER 5 CONCLUSION AND RECOMMENDATION	88
5.1 Research Summary	88
5.2 Research Contribution	89
5.3 Research Limitation.....	92

5.4 Future Directions 92

LIST OF FIGURES

Figure 1.1: Unstructured-Structured Data Continuum, highlighted Text Data [8].....	1
Figure 1.2: How LDA views Data[31]	4
Figure 1.3: Generative Process of LDA.....	5
Figure 2.1: Graphical Model of LDA[76].....	19
Figure 2.2: Probabilistic graphical model of variational inference in LDA [85].....	27
Figure 3.1: Research Framework for the Design and Development of Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA).....	40
Figure 3.2: Overview of the Adaptive Selection of Inference Method (ASIM) for LDA	43
Figure 3.3: Adaptive Selective of Inference Method (ASIM)	45
Figure 3.4: First Stage: Hyperparameter Optimization of Inference Algorithms	49
Figure 3.5: Cross Validation	53
Figure 3.6: Example of K-Fold Cross Validations Folds	54
Figure 3.7: Second Stage: Selection Filter.....	56
Figure 3.8: Data Pre-Processing	57
Figure 3.9: Document Term Matrix.....	59
Figure 3.10: Screenshot of Original Data and Pre-processed Twitter Airline Sentiment Data (Medium Complexity).....	61
Figure 3.11: Example of Raw Textual Dataset.....	63
Figure 3.12: Example of Stop Words List in NLTK	64
Figure 3.13: Removed Stop Words.....	65
Figure 3.14: Stemmed Words	65
Figure 3.15: Lemmatized words	65
Figure 3.16: Bigram words	66
Figure 4.1: Experimentation Set-Up.....	75
Figure 4.2: A Confusion Matrix.....	77
Figure 4.3: Accuracy Analysis for Low, Medium and High Complexity Dataset	81
Figure 4.4: Precision Analysis for Low, Medium and High Complexity Dataset.....	83
Figure 4.5: Recall Analysis for Low, Medium and High Complexity Dataset.....	85

LIST OF TABLES

Table 1.1: Research Scope Mapping	11
Table 1.2: Thesis Organization	14
Table 2.1: Comparison between four Topic Modeling algorithms.....	16
Table 2.2: Existing Works in Latent Dirichlet Allocation.....	22
Table 2.3: Comparison between Deterministic Approximate Inference and Stochastic Approximate Inference	25
Table 2.4 LDA Hyperparameters.....	32
Table 2.5: Approaches for Hyperparameter Tuning	34
Table 2.6: Objective Functions for Optimizing Latent Dirichlet Allocation.....	37
Table 3.1: Probability Distribution of Hyperparameter	50
Table 3.2: Grid Search Hyperparameters Generation Setting	51
Table 3.3: Notations for Objective Function	52
Table 3.4: Example of Stemmed Word (Argue).....	58
Table 3.5: Overview of the Datasets[98]	60
Table 3.6: Labels for Each of the Datasets	62
Table 3.7: Textual Dataset Characteristics	63
Table 3.8: Tokenized Data.....	64
Table 3.9: Random Initialization of Hyperparameters (Random Search)	67
Table 3.10: Initialization of Individual Inference Algorithms.....	67
Table 3.11: Output of a Single Training Process.....	68
Table 3.12: Final List of Topic Coherence Score (Random Search).....	68
Table 3.13: Initialization of Hyperparameters (Grid Search)	69
Table 3.14: Final List of Topic Coherence Score (Grid Search).....	70
Table 3.15: Input for Second Stage: Selection Filter.....	71
Table 4.1: Example of Confusion Matrix Classes with Results	77
Table 4.2: Computer Specifications.....	79
Table 5.1: Research Objective Mapping to Contribution	91

LIST OF ABBREVIATIONS

LDA	Latent Dirichlet Allocation
ASIM	Adaptive Selection of Inference Method
VB	Variational Bayesian
GS	Gibbs Sampling
LDA-GS	Latent Dirichlet Allocation with Gibbs Sampling
LDA-VB	Latent Dirichlet Allocation with Variational Bayesian Inference
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
HDP	Hierarchical Dirichlet Process
TF-IDF	Term Frequency and Inverse Document Frequency
pLSA	Probabilistic Latent Semantic Analysis
csv	Comma Separated Value
NLP	Natural Language Processing

CHAPTER 1

INTRODUCTION

1.1 Research Background

There has been a surge of data growth all over the world. This is partly due to the advent of new technologies such as social media platforms [1-4], the Internet of Things (IoT) [5-7] and general relevancy of internet usage [8], among others. Majority of these data points are from unstructured data. This unstructured data differs from structured data as the data is not inherently bound or confined to a set of predefined structure. Within the unstructured data realm, the processing of textual data has been touted to have a lot of potential to be unlocked. Textual data is defined as a collection of data which normally consists of natural language such as English and easily comprehend by a human being. Few examples of textual dataset are e-mails, documents, and reports. Due to the characteristics of the data, textual data is a challenging dataset to be analysed, understood, and leveraged on. Figure 1.1 depicts the unstructured-structured data continuum, depicting text data as one of the challenging highly unstructured data in comparison to other types of data.

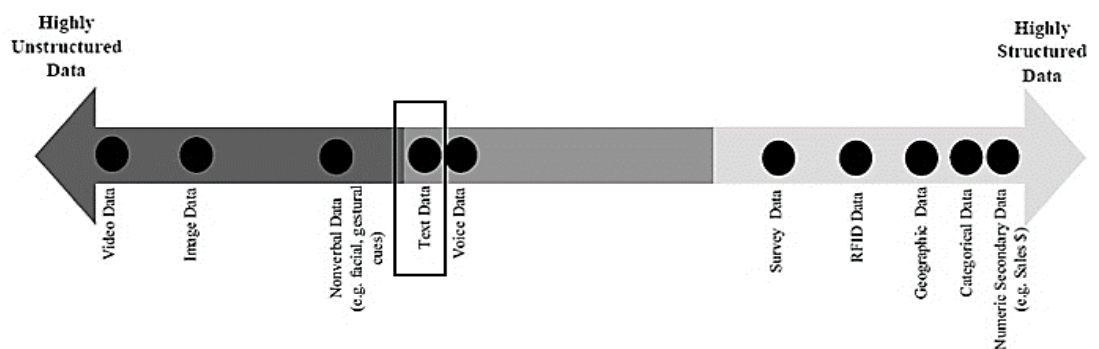


Figure 1.1: Unstructured-Structured Data Continuum, highlighted Text Data [8]

With realisation that there is a lot of potential and knowledge that can be extracted from the textual data, a lot of research is being conducted in this specific area. This research, Natural Language Processing (NLP), is applied in various fields, including the financial services [9], software engineering and development [10-12], news [13, 14], and medical sciences [15-17]. NLP research focuses on this problem, by leveraging computer science, statistics, and machine learning, to extract meaning from human language dataset. NLP aims to obtain the understanding of the textual dataset and use the knowledge extracted to perform either descriptive, predictive, or prescriptive analytics. One of the prominent areas where NLP is used as a predictive analytics enabler, is in financial markets. In this specific area, sentiment analysis, which is part of NLP, is actively used as one of the predictors for market trends [18]. Sentiment analysis has been proven to be a powerful indicator of how the market views a particular company or product, either positively or negatively. To enable this capability, various text representation algorithms are used. Text representation algorithms, or vectorizers, provide a representation of the textual dataset in a manner that a computer would understand. Text representation algorithms mainly use probabilities to represent the dataset into numerical format. One of the most popular techniques is Term Frequency (TF) [19]. TF represents the textual dataset in the form of counts, of how many occurrences of the words in the dataset. However, this is inadequate to represent the true meaning of the textual dataset. This is because of the inconsistent nature of data in real-world situations. Most of the time the real context or semantic meaning of the textual dataset are beyond than number representation of the words in the data [20]. This challenge is further amplified by the complexity of the textual dataset. Dataset complexity is measured by the number of unique words in a textual dataset. Having many unique words in the document can increase the perplexity, a measure of how surprise a model seeing a before unseen data [21]. This problem is very evident in social media, where a lot of new words and usage of slang which will result to higher complexity of data [22].

Topic modelling is one of the family of algorithms used in representing the textual dataset. Topic modelling is a statistical model which assume that textual datasets are a part of a bigger pool of textual datasets. For example, a document is not considered independent of other document, it is assumed that there is a higher meta-class or

category that this document falls into. This assumption enables the algorithm to establish relationship beyond word-level, by extracting topic-level information. Topic-level relationship is useful in understanding useful patterns in a document or even understanding the interaction of words in different kind of document. For example, the word 'pool' might have a different meaning if it comes from an Information Technology (IT) related article in comparison of the exact same word in a sports magazine. Topic modelling has also established itself as a powerful technique to perform topic discovery or semantic mining from unordered or unlabelled textual dataset. This is because topic modelling is an unsupervised machine learning technique, which does not require labelled dataset for it to learn from. Due to this nature, it has seen a popular usage in understanding customer reviews [23-26] and linguistic sciences [27].

Latent Semantic Analysis (LSA) was an early breakthrough in topic modelling research area. The aim of LSA is to solve the main challenge of extraction of meaning from textual dataset [28, 29]. By establishing the distinction of lexical level and semantic level of the textual dataset, LSA can develop a degree of meaningful feature extraction. The main idea of LSA is to map documents to a vector space of reduced dimensionality, the latent semantic space. LSA consists of two main processes, which are the development of Document-Term Matrix (DTM) and Singular Value Decomposition (SVD). DTM is a matrix that contains the frequency of words or terms that occur in a dataset. After that, SVD is applied which identify and form semantic generalisations from the textual dataset. Based on the SVD output, the dimensionality of the semantic space is lower than the number of unique words in the textual dataset. This is one of the good features of LSA, as the size of the output is smaller, projecting relationship of words co-occurring in other documents in the textual dataset. Documents that share no words would have no projection on each other in a space where each word was allocated its own dimension, but in the semantic space created through LSA, they do, which means that they can be compared with.

However, LSA have some challenges such as its inability to adapt to new and unseen data, without having to perform retraining on the entire new dataset. This is because of the deterministic nature of LSA, which develop boundary of different classes

or clusters in the dataset, solely based on the trained dataset. Based on the explicit nature of the boundaries set, LSA might not be able to adapt to words that were not in the original training data. Another issue with LSA is its usage of SVD. SVD assumes that the data is normally distributed. In real-world situation, LSA might not be a suitable choice due to the presence of non-normally distributed data.

Based on the gaps found in LSA, a growing amount of research focuses on a generative approach. Generative approach allows the relaxation of the clustering process. The important feature of the generative approach is the assumption that the dataset being trained is part of a bigger dataset. To adhere to this assumption, generative model includes the distribution of the dataset. Generative topic modelling algorithms used is the Latent Dirichlet Allocation. Latent Dirichlet Allocation (LDA) is a generative probabilistic topic modelling algorithm [30]. To cluster the data, LDA observes data in textual forms as mixtures of words. These mixtures of words belong to specific topic or clusters. Figure 1.2 depicts how LDA views text data.

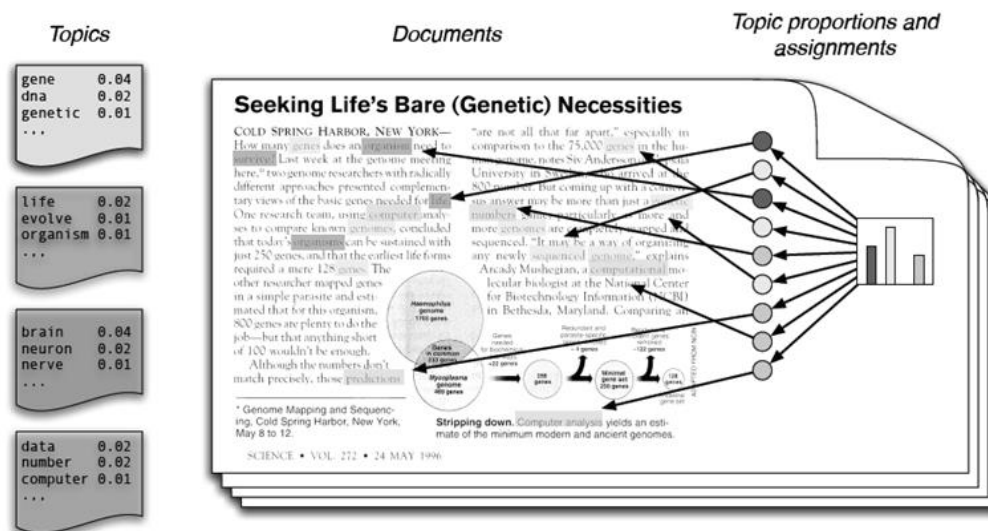


Figure 1.2: How LDA views Data[31]

The general intuition of how LDA clusters the data can be understood based on Figure 1.2. Each of the word in *Seeking Life's Bare (Genetic) Necessities* is assigned into topics. For the sake of simplicity, the figure highlighted several keywords and its corresponding topic assignment. Different topics are denoted by different shades of

circles on the right-side of the figure. Two main outputs of LDA can be seen via this example. Referring to Figure 1.2, LDA observes that there are multiple topics in a single document. The topic proportions, denoted by the histogram on the right-side of the figure, varies according to the document analysed. This enables the identification of ambiguous words. Ambiguous words are defined as words that can be interpreted differently in different scenarios or situation. For example, the word *well* in normal conversation would mean healthy but in the context of oil and gas it is a chamber (hole) in the Earth. The existence of multiple topics solves this issue by assigning weightage to words that belongs to different topics. Multiple topics assignment also allows LDA to adapt to new and unseen data, whereby different documents would have different topic probabilities and assignments.

As LDA is a generative probabilistic algorithm, it assumes that words are generated from random distribution of topics. This provides the basis on understanding the generative processes on how the data is being generated. To understand the inner workings of LDA, we need to understand several processes and assumptions made by the algorithm. Figure 1.3 depicts the generative processes that is assume by LDA.

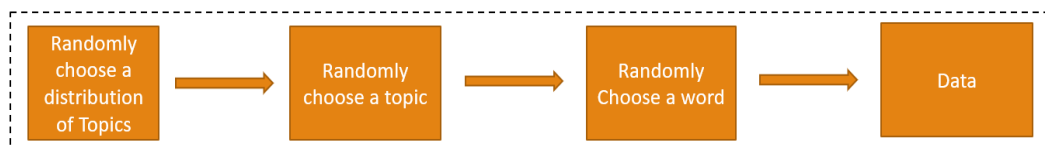


Figure 1.3: Generative Process of LDA

LDA assumes that a distribution of topics is randomly chosen. From this list of distribution of topics, a topic is randomly chosen. Based on the selected topic, a word is randomly chosen. This process repeats until there all words belong to the selected topic are exhausted. After this iterative process is complete, it is repeated for different topic. At the end, the data generated will be comprising of multiple topics. In real-life, the topics are not explicitly labelled in each of the documents. These topics are latent which is defined as hidden and unobservable [32]. In this situation, LDA is more useful on extracting the topic proportion and assignment for each word. To extract these latent structures of the data, LDA performs inference or parameter estimation. Through inference process, multiple latent knowledge on the data can be extracted. These latent knowledges (which include topic proportion and assignments) are inferred by a joint

distribution of all the unobserved variables and the observed words. The process is called as Bayesian inference. The goal of Bayesian inference is to understand or predict the posterior distribution based on observable evidence. In this context, the posterior distribution is the latent structure of the data, which are the topic proportion and topic assignment for each of the words in the document.

Inference process in LDA estimates the posterior distribution through Maximum a Posteriori (MAP) assignment. MAP performs inference or parameter estimation through maximizing the posterior distribution of the parameters that ensures the best fit to the data. MAP extends the typical parameter estimation algorithm, Maximum Likelihood (ML) by including prior information into consideration in estimating best posterior distribution. In LDA, the posterior distribution that needs to be inferred are the topic proportions and topic assignments. However, in LDA, it is not possible to infer the parameters analytically. This is because of the intractability of the marginal likelihood term. The marginal likelihood term is used as the normalizing constant, which is impossible to compute as it needs to consider all possible and probable value of the topic proportions and topic assignments. This leads to the inability to perform exact inference in retrieving the latent structure of the data. To solve this problem, approximate inference algorithms are applied in the inference process. Two widely used approximate inference algorithms are Gibbs Sampling (GS) and Variational Bayesian (VB) inference [33]. Both approaches are applied differently. GS focuses on approximating the posterior distribution via sampling method. VB approximates the posterior distribution via converting the inference problem into an optimization problem [34].

VB emerged as one of the algorithms for inferring the posterior distribution. As it approximates the posterior distribution using simpler and more tractable distribution, VB is usually used in analysing large corpora or text documents [35, 36]. Implementations of VB shown consistently outperforms other types of inference algorithms in terms of performance [37]. One of the reasons why VB can approximate the posterior distribution faster is that VB selectively approximate the posterior distribution. By selectively approximate the posterior distribution, VB only consider a family of tractable distribution in inferring the posterior distribution. One of the

examples of the tractable family of distribution is mean-field distribution. Through this process, VB simplifies the posterior distribution estimation by considering only tractable family of distribution. This removes the complex approximation needed for the posterior distribution. However, this creates another challenge. VB only consider the selected family of distribution [38]. This means that the approximation of posterior distribution does not consider other types of distributions [39]. By not considering other family of distributions, it can reduce the accuracy of the approximated posterior distribution. However, if the selected family of distribution conforms to the target distribution, the resulting inference could be of high accuracy[35].

In contrary, GS focuses on approximating the posterior distribution via sampling method. This method provides a more exact result as it provides an exact approximate of the target distribution[40]. However, to achieve the almost exact approximate of the target distribution, it requires higher computational usage in comparison to VB. Generally, the usage of GS is more viable in handling small dataset. This trade-off between speed and accuracy between using VB and GS creates a situation whereby a person who wants to use LDA to model their data needs to know the parameters involved in their computation.

Another factor that can influence the performance of inference algorithm is the selection of hyperparameters. To perform approximate inference on the data, prior information must be initialised on each of the inference algorithm. The prior information or concentration hyperparameters can affect the fitness of inference algorithm to a dataset. Most of the research focuses on setting a fixed concentration hyperparameters which might not be the best selection for the given dataset. Different complexity of textual dataset has seen to have different characteristics which requires different set of prior information. To address the different complexity of textual dataset, both the selection of inference algorithm and the process of optimizing the hyperparameters are important.

1.2 Research Motivation

Textual dataset complexity is measured by the count of unique words residing in a particular dataset. For example, textual dataset coming from social media is a high complexity dataset as there are a lot of unique words being generated daily. The complexity of textual dataset presents challenges in analysis and extracting knowledge from it, as mentioned in Section 1.1. One of the biggest challenges in analysing high complexity data is in building the suitable text representation. Textual dataset with high complexity can lead to text representation model to be perplexed when tested with new and unseen data. Perplexity is a measure of how well the text representation model learn on the textual dataset, tends to increase with higher complexity of textual dataset. This is because it is generally impossible for text representation model to learn all the unique words based on just training data as there might be new and unseen words in the testing dataset. Besides perplexity, another challenge of textual dataset is that most of textual dataset are unlabelled. This requires an unsupervised approach in the extraction of text representation. Based on Section 1.1, it has been recognized that Topic Modelling algorithm are being researched to address this issue. Topic modelling, being an unsupervised algorithm, can address non-labelled data. The design of LDA depends heavily on the inference algorithms. Inference algorithm such as VB and GS are used as they generally provide robust posterior. However, it is noted that different inference algorithm adapts differently to different complexity of textual dataset. There is also the room for improvements to optimize the inference algorithm selected in terms of tuning the hyperparameters. The hyperparameters selected, both the concentration parameters and the number of topics will have an effect to the fitness of the inference algorithm to data. Thus, the motivation of this research arises from identifying the best inference algorithm for varying complexity of textual dataset, by considering both GS and VB as well as optimizing its hyperparameters.

There is on-going research on the inference methods for LDA. This research mainly focuses on improving the efficiency of the inference algorithm. Research on improvements on GS are done through block updating of GS [41], perform parallel inference [42], perform batch Gibbs sampler, incremental Gibbs sampler. Research on improvements on VB are done through collapsed VB, incremental learning on VB, Map

Reduce LDA (Mr. LDA). As these research focuses more on improving the efficiency, there is a gap in improving the fitness of inference algorithm to the dataset. There is also some research gap in focusing on the utilising both VB and GS.

1.3 Problem Statement

Textual data has become more advent. As these types of data are constantly being generated daily, especially due to rise social media platforms, it creates a challenge in the pursuit of analysing and extracting knowledge from it. A few of the challenges in processing textual data is that most of the data generated are not labelled, and the difficulty in effectively extracting the context of the textual data. Traditional text representation methods which would not be able to extract the contextual meaning of textual dataset [19, 20]. Topic modelling emerges as one of the methods that can handle non-labelled data as it is an unsupervised algorithm [43]. Topic modelling able to perform extraction of latent structure of the data, which can extract the context of the textual data.

LDA is one of the algorithms under the Topic Modelling family of algorithms [33]. LDA, being a generative and probabilistic based model, is highly adaptive to new set of data. This removes the need for LDA to retrain on the entire dataset again if it encounters new additional data. Given the pace of the data that are being generated, this is considered as a strong feature of LDA, where less time is needed to train. The effectiveness of LDA depends on the suitable inference algorithm used. VB and GS are distinguished as the inference algorithm that are both efficient and robust in inferring the posterior distribution. VB has been an efficient algorithm, managed to perform the computation of posterior distribution quicker than GS. However, an unoptimized VB performs less effectively compared to GS, to a given textual dataset. This challenge is aggravated by the varying complexity of textual dataset. The complexity of textual dataset measured based on number of unique words in the dataset presents a challenge in developing suitable text representation method [44, 45]. For varying complexity of textual datasets, selection of prior hyperparameters is a factor in the performance of the inference method. As mentioned previously in this section, the performance of an

unoptimized inference algorithm might be more inferior in comparison to an optimized one. The performance of the inference algorithm in this research is defined by the topic coherence score.

1.4 Research Questions

From the gaps found in the existing implementations of the Latent Dirichlet Allocation's inference algorithm and problems defined in Section 1.1 and 1.2, the following research questions are identified and need to be addressed:

- i. How to improve the inference method to adapt to different complexity of dataset when hyperparameters are optimized based on topic coherence score?
- ii. How to formulate the Adaptive Selection of Inference Method for Latent Dirichlet Allocation to adapt to varying complexity of textual data environment?
- iii. What is the performance for the proposed design compared to the existing approach when modelling different complexity of textual dataset?

1.5 Research Objectives

The aim of this research is to design a high-performance inference method for LDA that utilizes both hyperparameter-optimized inference algorithms. The aim of this research is achieved by realising each of the research objectives mentioned below:

- i. To propose an adaptive selection of inference method for Latent Dirichlet Allocation (ASIM-LDA) based on maximizing topic coherence score with given textual dataset
- ii. To improve Latent Dirichlet Allocation's inference method by combining random search and grid search in hyperparameter optimization to obtain the highest topic coherence score in textual data environment.

- iii. To assess the performance parameters of ASIM-LDA, measured through accuracy, precision and recall analysis, through experimentation with different complexity of textual dataset.

1.6 Scope of the Research

In this section, the scope of this research is explained briefly. Table 1.1 exhibits the research scope mapping for this study.

Table 1.1: Research Scope Mapping

Scope	Description
Textual Dataset	Textual dataset is defined as dataset that consists of natural human language such as English or Malay language. Textual datasets are predominant in areas such as reports, social media posts and news. Characteristics of the textual dataset which can be highly complex due to number of unique words residing in the dataset is the focus of the study in this research.
Latent Dirichlet Allocation	Latent Dirichlet Allocation (LDA) is the topic modeling algorithm used in this research. As the topic modeling algorithm, LDA is used for text representation technique prior to further processing. LDA provides the capability to establish relationship between words in the dataset, enabling the extraction of contextual meaning of the words. This is done through the assumption of the words belonging to a certain number of topics. LDA relies on approximate inference algorithms to extract these structures of the data, which is inherently hidden.

<p>Approximate Inference Methods in Latent Dirichlet Allocation</p>	<p>In LDA, two approaches of approximate inference algorithms are mainly used, namely deterministic approach and sampling approach. In this research, both techniques are studied, where one inference algorithm from each of the approaches are selected. VB is selected from the deterministic approach as it provides high efficiency in processing large amount of data. GS is chosen from the sampling approach as it guarantees the posterior distribution approximated to be asymptotically exact. Both inference algorithms are studied and the hyperparameters used in the inference algorithms are optimized. Hyperparameters optimization is done to maximize a defined an objective function.</p>
<p>Topic Coherence Score</p>	<p>Topic coherence score is selected as the objective function for the hyperparameter optimization of the inference algorithms. Topic coherence measure the quality of the topics generated by analysing the degree of semantic similarity between the top words in a particular topic. This has been noted as a good intrinsic measure of the fitness of inference algorithm to a particular dataset.</p>
<p>Hyperparameter Tuning</p>	<p>In this research, the improvements proposed is based on the hyperparameter tuning of LDA. The tuning approaches used are Grid Search and Random Search. These approaches are discussed in detail in Chapter 2.</p>

1.7 Significance of Study

In processing textual data, it is generally infeasible for us to have a supervised algorithm due to lack of labelled data. Another challenge of analysing textual data is the need to have a robust text representation algorithm, that not only focuses on individual words as independent token, but also able to extract thematic structure of the

textual dataset. This also true for unsupervised techniques as it can work with non-labelled dataset. LDA has since gain popularity usage in understanding of the dataset, without requiring any label in the data. Another advantage of LDA is that its ability to extract thematic structure of the textual dataset, which is otherwise hidden. This has become is strongest feature, as this allows the notion of hidden topics which can act as soft clusters of data in the textual dataset.

One of the possible industries that would gain benefit from a high-performance clustering algorithm is the oil and gas sector. Studies have shown that data owned by oil and gas companies are largely unstructured[46]. Unstructured data is defined as data that is not managed by a structured database system or tagged by specific keywords[47]. Textual data and images are a few examples of unstructured data [48]. According to [49] , more than 80% of the data held by oil and gas companies are unstructured. Another characteristics of the oil and gas data is that it is not stored in a publicly accessed database [49]. This is to safeguard expensive datasets. The data such as geographic surveys cost millions to produce which make it one of the most valuable asset an oil and gas company owns[49].

These two characteristics of oil and gas data create a problem in searching accurately through the data [50]. The author in [50] stated that this problem consumes time as it takes a long duration to locate for the exact information. Based on a survey conducted by [46], oil and gas engineers often spend about 50% to 80 % of their time gathering and making sense of information at hand instead of utilizing their time to make decisions. This provides an excellent opportunity to utilize LDA in clustering the data for effective information retrieval. However, there have not been significant studies on clustering algorithm in oil and gas domain. Most of the implementations focused on biomedical [51], marketing [52], and social media [3]. Implementations in these fields have seen success in both retrieving correct information and reducing the time taken to make sense of the data. Thus, this research on developing efficient and effective inference algorithm will be able to provide significant impact on the domain on information retrieval for oil and gas industry.

1.8 Thesis Organization

In this section, the high-level thesis organization is presented. Table 1.2 exhibits the description of each chapter in this thesis.

Table 1.2: Thesis Organization

Chapter	Description
Chapter 1	This chapter discusses the introduction of the thesis. Introduction to the research work on LDA, and approximate inference algorithms used is done.
Chapter 2	This chapter describes the existing works on approximate algorithms used in inferring posterior distribution of LDA. Both VB and GS approaches are discussed.
Chapter 3	This chapter presents the construction of adaptive selection of inference algorithm for Latent Dirichlet Allocation (ASIM-LDA). The algorithm is explained in both algorithm and its mathematical foundation.
Chapter 4	This chapter discusses the experiments done to evaluate the proposed algorithm based on different complexity of datasets.
Chapter 5	This chapter presents the research findings and possible future work on improving inference algorithm.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the literature study conducted is discussed in more clarity. The literature review covers the review of existing implementations and analysed the research gap identified. Section 2.2 exhibits the comparison between Topic Modelling methods. Section 2.3 discusses the Latent Dirichlet Allocation in depth. Section 2.4 elaborates on the existing works of Latent Dirichlet Allocation. Section 2.5 explains the inference process in Latent Dirichlet Allocation. Section 2.6 discusses the approximate inference algorithms used in Latent Dirichlet Allocation. In this section, the strength, and the weaknesses of each of the inference algorithms are explained. The gaps found from this study becomes the basis of development of ASIM-LDA. To develop the fundamentals of ASIM-LDA, hyperparameter tuning approaches and selection of objective function are further studied. Section 2.7 explores the different approaches of hyperparameter tuning. Section 2.8 discusses the selection of objective function for Latent Dirichlet Allocation, which compares between perplexity score and topic coherence score. Section 2.8 summarizes the chapter.

2.2 Topic Modelling Methods

Topic modeling is used to uncover hidden insights and underlying concepts of textual datasets and word documents [53]. It is developed to overcome the weakness of traditional text representation algorithms or text vectorizers which relies heavily on word frequency extraction. The main weakness with the traditional approach is its inability to extract contextual meaning of textual dataset. Due to the lack of consideration of hidden thematic context of the textual dataset, it does not provide

capability to retrieve information based on the underlying conceptual topic or meaning of the document. This is important to be addressed as documents are normally written with meaning and intentions [31]. Topic modeling is one of the approaches developed to consider the underlying semantic structure of the data in the retrieval process. Throughout the years, multiple topic modeling algorithms have been developed [33, 54, 55]. Table 2.1 contains the comparison between four Topic Modelling methods.

Table 2.1: Comparison between four Topic Modeling methods

Methods	Strengths	Weaknesses
TF-IDF	TF-IDF is simple and does not require a lot of computational usage. Due to its simplicity, it provides an easy to understand metric to describe the query done[56, 57].	It is not able to create relationship with synonym words. This reduces its ability to extract full meaning of the textual dataset[58, 59].
Latent Semantic Analysis (LSA)	LSA establishes semantic similarity measures between words. Through formalizing the semantic similarity measures, it is able to detect better synonym words [32]. LSA reduces the dimensionality of the textual data, as it incorporates Singular Value Decomposition.	LSA is not able to handle polysemy terms. Polysemy term is a word that has multiple meanings[60]. It is not able to differentiate polysemy terms because each term in the dataset is represented as equal single point in the concept space. This reduces the ability to distinguish polysemy terms .
Probabilistic Latent Semantic Analysis (pLSA)	As an extension of LSA, pLSA is better able to handle polysemy. It is because pLSA	pLSA tends to overfit to the training data[61]. This is because the number of

	distributes the assignment of probabilities of a term corresponding to different meanings of a term[32]	parameters in the model grows linearly with the size of the data. pLSA is a deterministic model, not a generative model. Due to this nature, it is not able to assign probability to unseen data without having to retrain on the entire dataset.
Latent Dirichlet Allocation (LDA)	LDA is a generative method by establishing the generative processes of how documents are created. The generative process assumes that the seen dataset is a subset of a bigger unseen dataset. This LDA to generate topic assignments for new unseen documents[53].	As it is a generative model, LDA is highly dependent on the inference algorithm that is selected.

Based on Table 2.1, a literature analysis has been made between four different types of topic modeling methods. TF-IDF or Term Frequency-Inverse Document Frequency is one of the simplest topics modeling algorithm. It assigns weightage to each term in the document by analysing the occurrence in a document and comparing it to their respective occurrence in the whole document set or corpus. TF-IDF is relatively easy to implement[56]. However, it does not establish any semantic relation between the words in the document. This reduces its ability to extract the full meaning of the textual dataset as words might have contextual meaning across different documents or inter-documents. Creating relationship with synonym words are also one of the challenges when implementing TF-IDF. It views the words as independently inside the document which restrict its capability to detect synonym words.

LSA improves by establishing the concept of semantic relationship. The semantic relationship is defined by the concept of larger text segment, where LSA introduces the notion of latent semantic structure of textual dataset. To extract the latent structure, LSA uses a two-step approach. The first step, LSA converts the textual dataset into a matrix representation. Subsequently, LSA applies Singular Value Decomposition (SVD) to the matrix generated[62]. By employing SVD in the process, LSA has been able to cluster together synonym words in the document. It also achieves dimensionality reduction which reduces the complexity of computation of the data. Polysemy is an issue that LSA does not able to address as it viewed the individual words as independent points in the context space.

pLSA is a probabilistic extension of LSA. Through establishing basis of probability in the process, it is now able to distribute probability assignment to each term occurring in the documents. Based on the probability assignment, pLSA adopted the soft-clustering approach, which provides the capability to address the polysemy issue. As it is not a generative model, it is not able to assign probability or topic assignment to unseen data without having to retrain the entire dataset. The number of parameters in the algorithm also grows linearly with the size of data. This could cause computational issue with large sets of data.

LDA, having a well-defined generative feature, solves two main problems of pLSA. LDA can cope with unseen data, by generating or assigning topics to new data without having to retrain the entire dataset. By relying on latent variables in assigning topics to the documents, the number of parameters does not increase accordingly to the size of the data. However, through the introduction of latent variables into the process, LDA is highly reliant on the inference algorithm selected. Through the comparison between the topic modeling algorithms, LDA has been selected for this study. It is mainly because it addresses all the issues that have been found in the previous algorithms for topic modeling. It is both computationally efficient and able to handle new data effectively. Section 2.3 discusses more details on how LDA works.

2.3 Latent Dirichlet Allocation

LDA is a generative probabilistic method used to cluster collections of unstructured data [30, 33]. LDA is one of a variant of topic model, which cluster data based on themes [63]. The method selected can classify into their respective meaningful themes. It is generated based on an idea that documents are consisting of mixtures of latent topics and the topics are characterized by the distribution of words. Latent is defined as something which presence but not visible. LDA implements the bag-of-words model which ignores the order of the words [64]. Bag-of-words model is a model that tokenize every word regardless of the grammar and their word order. However, LDA will view the words independently and the occurrence of each word is utilized in training a classifier[65]. Figure 2.1 illustrates the graphical model of how LDA works.

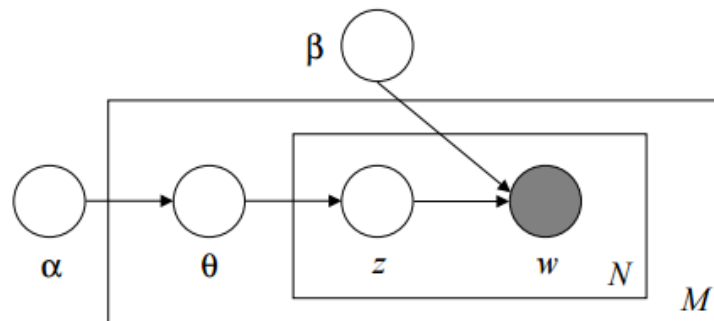


Figure 2.1: Graphical Model of LDA[31]

In Figure 2.1, M denotes the documents, N represents the set of words in the document, w represents the denoted words in the document, z represents the topic for the denoted word, θ represents the topic distribution for the document, α denotes the parameters set for the topic distribution per-document and β denotes the parameters set for the topic distribution per-topic word[66]. Based on Figure 2.1, it also depicts LDA as a three-level model. The three-level model can be further breakdown into following components:

- i. For corpus-level, α and β are the respective parameters. These parameters are conditioned and sampled once per the entire collection of documents.
- ii. For document level, the parameter is θ_d , a draw from multinomial clustering variable. It is sampled once for each of the document.
- iii. For word-level, z_{dn} and w_{dn} are the parameters. These parameters are sampled once for each of the words inside the entire collection of documents.

For each document w , LDA assumes the following generative processes:

- i. For each D documents:
 - a. Draw $\theta \sim \text{Dirichlet}(\alpha)$
 - b. For each N words w_n :
 - i. Draw a topic $Z_n \sim \text{Multinomial}(\theta)$
 - ii. Draw a word w_n from $p(w_n|Z_n, \beta)$

Based on the generative processes exhibited, there are two assumptions made on how the model view the data. The assumptions are:

- i. The dimensionality k of the Dirichlet distribution is assumed to be known.
- ii. Word probabilities are parameterized by β is assumed to be fixed.

As LDA is a generative probabilistic model, it is often expressed in the form of joint distribution as:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^N p(z_n|\theta)p(w_n|z_n, \beta) \quad (2.1)$$

To obtain the marginal distribution of a document, two processes need to be done which are, first integrating over the topic distribution,

Process 1

$$p(z, w|\alpha, \beta) = \int p(\theta|\alpha) \prod_{n=1}^N p(z_n|\theta)p(w_n|z_n, \beta) d\theta \quad (2.2)$$

Then, through summation over z , the marginal distribution of a document:

Process 2

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \quad (2.3)$$

To find the probability of the collection of documents, or corpus, we take the product of marginal probabilities of each of documents:

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d \quad (2.4)$$

As the aim of the algorithm is to extract the hidden structure or 'topic', the parameters that of concern are θ and z . Both parameters are latent, which they are not observed from the data. Thus, the joint distribution of $p(\theta, z|w, \alpha, \beta)$ is:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \quad (2.5)$$

By solving the joint distribution of $p(\theta, z|w, \alpha, \beta)$, the posterior distribution of data can be extracted. This posterior distribution is the representation of the structure of the data which are not observed directly. The ability of LDA to extract latent structure of any type of data makes it one of the most used data clustering method. In the next subsection, existing works in LDA are summarized and explained.

2.4 Existing Works in Latent Dirichlet Allocation

There are studies conducted on utilizing Latent Dirichlet Allocation in different areas of research. These researches are studied, and the summarization of the analysis are discussed in Table 2.2.

Table 2.2: Existing Works in Latent Dirichlet Allocation

Author(s)	Areas of Research	Findings	Future Work
Guo, Barnes, Jia[67]	User Reviews	The inference part of the LDA is computationally intensive.	To utilize a more robust inference engine for computing topic assignments.
Tirunillai and Tellis [52]	Online marketing reviews	LDA can effectively analyse data at a highly granular level. To achieve high-quality inferring of topics, it is very computationally intensive	Adopting or improving inference to achieve quality results.
Moro et.al [68]	Business Intelligence in Banking	LDA's inference is computationally intensive and is considered as NP-hard.	Adopting effective inference for clustering data.
Dyer, Lang and Lawrence [69]	Economics	Even though LDA can process large scale data, it requires efficient inference to be able to produce high quality topics	Employ robust inference
Eddy et.al. [43, 70]	Feature location	Time required by LDA to complete its task significantly depends on the performance of its inference.	Identifying the best inference settings or technique

Table 2.2 depicts the analysis of the existing research on using Latent Dirichlet Allocation as means to cluster data in various domains. The findings from all the research consistently mentioned the importance of having a robust inference in LDA. This is because inference is a major part of the LDA algorithm. In the next subsection, the inference part of the LDA is explained.

2.5 Inference in Latent Dirichlet Allocation

The key problem of LDA is discovering or obtaining the posterior distribution. The process of obtaining the posterior distribution is defined as inference. Through inference, we reverse the generative process by generating the posterior distribution through sampling of every latent variable needed conditioned on the observed data. In LDA, it means solving the joint distribution of $p(\theta, z|w, \alpha, \beta)$.

As there is possibly a huge possible value of latent variables α and β , the posterior distribution makes it impossible to exactly infer the latent structure [31, 33]. In addition, the coupling between θ and β creates difficulty in computing the logarithm of joint distribution of $p(\theta, z|w, \alpha, \beta)$, [71]. The normalization factor, $p(w|\alpha, \beta)$ also cannot be computed directly[40]. Thus, approximate inference algorithms are used. The process used in approximating the posterior distribution is Bayesian inference.

Through Bayesian inference, we are able to infer posterior probability of a random variable given some observable observations[72]. This process focuses on making conclusions about a parameter θ or unobserved variable. It is done through probability statements which are made conditional on observed values y [73]. The statements are normally written as $p(\theta|y)$ or $p(\bar{y}|y)$.

To make probability statements, $p(\theta|y)$ or $p(\bar{y}|y)$, a joint probability distribution must be made. The joint probability distribution is the product of two densities: the prior distribution $p(\theta)$ and $p(y|\theta)$. The product is written as:

$$p(\theta, y) = p(\theta)p(y|\theta) \tag{2.6}$$

The posterior distribution or density is obtained through conditioning the joint probability distribution with the observed data y [74]. This conditioning is possible through Bayes rule. The posterior distribution or density obtained is

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)} \quad (2.7)$$

Where $p(y)$ is the sum over all possible values of θ . The posterior distribution can be re-written as:

$$p(\theta|y) \propto p(\theta)p(y|\theta) \quad (2.8)$$

After establishing how approximate inference going to infer the posterior distribution, Section 2.6 discusses the comparison of approximate inference algorithms utilized by Latent Dirichlet Allocation.

2.6 Approximate Inference Algorithms for LDA

Based on the explanations with regards to LDA inference process, there are latent parameters that needs to be taken into consideration. This makes it infeasible to evaluate the posterior distribution or compute exact inference for the given distribution. It is due to the high dimensionality of the latent space taken into consideration. As LDA deals with discrete variables, the marginalization involves summing over all possible configurations of the latent variables. As there is a possibility of exponentially many latent states, this will certainly cause exact calculation to be expensive. To address this challenge, LDA resorts to approximation inference algorithms.

There are several approximation inference algorithms designed and implemented in approximating the posterior distribution for complex problems. In this subsection, several inference algorithms used in Latent Dirichlet Allocation are discussed. Inference algorithms used to approximate the posterior distribution for LDA can be

divided into two broad categories. The categories are the deterministic approximations method and stochastic sampling method. Table 2.3 shows the comparison between deterministic approximate inference algorithms and stochastic approximate inference algorithms.

Table 2.3: Comparison between Deterministic Approximate Inference and Stochastic Approximate Inference

Type of Approximate Inference Algorithms	Strengths	Weaknesses
Deterministic Approximate Inference	Deterministic approximate inference algorithms tend to be more efficient when compared to stochastic approximate inference algorithms [75].	It tends to be stuck in local optima [34]. As deterministic approximate inference neglects the conditional dependencies, it often yields under-estimated posterior variance [36].
Stochastic Approximate Inference	Given unlimited computational resources, stochastic approximate inference can generate exact results [76].	Stochastic approximate inference algorithms are computationally demanding [77]. It is often limited to small-scale problems. It is difficult to determine whether the samples generated independent from the required distribution [37].

Deterministic approximate inference algorithms are family of approximate inference algorithm that are mainly used to compute posterior distribution through providing the analytical approximation of the said posterior distribution. This is done through converting the problem of computing the posterior distribution into an optimization problem. Deterministic approximate inference algorithm uses a much simpler family of distribution as an approximation of the true posterior distribution. As it is an optimization problem, the loss function defined for this technique is the Kullback-Leibler Divergence (KL Divergence). The KL Divergence is a measure of how different one probability distribution to another probability distribution is. By combining these two concepts, the aim of deterministic approximate inference algorithm is to compute the best posterior distribution, by using a simpler family of distribution that minimizes the KL divergence. The main technique used in deterministic approximate inference algorithm is the variational Bayesian inference. This technique is discussed in more detail in Section 2.5.1.

Stochastic approximate inference algorithms approximate the posterior distribution by sampling from the desired posterior distribution. One of the mostly used technique in stochastic approximation, Markov Chain Monte Carlo (MCMC) sampling provides a process of systematic random sampling from high dimensional probability distributions. In general, MCMC provides approximation to the posterior distribution by drawing random and independent samples from the posterior distribution. Based on these samples, the expectation can be approximated by the finite sum of the random samples. By directly sampling from the posterior distribution, MCMC provides guarantee that it is asymptotically exact. The main technique used in the stochastic approximate inference algorithm is Gibbs Sampling. Gibbs Sampling is discussed in Section 2.5.2.

In this research, both stochastic and deterministic approximate inference algorithms are used. This is since both techniques are complementary to each other. By leveraging both techniques, the adopted method can reap the benefits of the strength from both techniques. This also minimizes the drawback effect as both techniques are used.

2.6.1 Variational Bayesian Inference

This inference algorithm is used in the original implementation of LDA. By using variational inference, tractable distribution is obtained through consideration of family of lower bounds. The family of lower bounds consists of variational parameters. This relaxed the distribution which allows adjustable lower bound on the likelihood of the exact posterior distribution. By approximating using variational inference, lower bound of the log likelihood is determined and iteratively tightened the bound [78]. To implement this, modification is done to the original probabilistic graphical model of LDA [53]. The modification removes the edges that caused the coupling between θ and β which are θ , z and w . Thus, the modified probabilistic graphical model is as depicted in Figure 2.2.

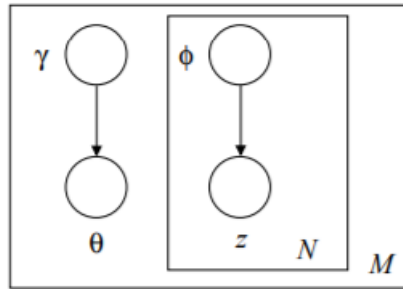


Figure 2.2: Probabilistic graphical model of variational inference in LDA [75]

Based on Figure 2.2, variational parameters are introduced γ and ϕ . Thus, the new joint distribution for each document is:

$$q(\theta, z|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^N q(z_n|\phi_n) \quad (2.9)$$

In this new joint distribution or posterior distribution, γ is the Dirichlet parameter and ϕ is the multinomial parameter. Both parameters are free variational parameters. The inference problem now evolved into optimization problem in obtaining optimal value of both γ and ϕ . As stated earlier, variational inference algorithm aims to find the tightest lower bound of the distribution. The problem of finding the optimal lower bound is:

The optimization problem is to minimize the Kullback-Leibler (KL) divergence between the exact posterior distribution and the modified variational distribution.

KL divergence is a measure of the divergence between a revised prior distribution and posterior distribution . To achieve minimal value of KL divergence, iterative fixed-point method is utilized [33, 79]. The update equations obtained in this method are:

$$(\gamma^*, \phi^*) = \arg \min_{(\gamma, \phi)} D(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta)) \quad (2.10)$$

The update equation represents the Dirichlet update conditioned to the observed words and based on the variational distribution, $E[z_n|\phi_n]$. Thus, the aim of iteratively updating γ is to find the optimal multinomial distribution for the observed words.

$$\phi_{ni} \propto \beta_{i w_n} \exp\{E_q[\log(\theta_i)|\gamma]\} \quad (2.11)$$

$$\gamma_i = \alpha_i + \sum_{n=1}^N \phi_{ni} \quad (2.12)$$

The expectation in the update for ϕ_{ni} is computed as:

$$E_q[\log(\theta_i)|\gamma] = \Psi_{(\gamma_i)} - \Psi\left(\sum_{j=1}^k \gamma_j\right) \quad (2.13)$$

Ψ is the first derivative of $\log \Gamma$ function. This is calculated using the Taylor approximation:

$$\begin{aligned}
\Psi(x) \approx & \left(\left(\left(0.00416 \frac{1}{(x+6)^2} - 0.003968 \right) \frac{1}{(x+6)^2} \right. \right. \\
& \left. \left. + 0.0083 \right) \frac{1}{(x+6)^2} - 0.083 \right) \setminus \frac{1}{(x+6)^2} + \log(x) \\
& - \frac{1}{2x} - \sum_{i=1}^6 \frac{1}{x-i}
\end{aligned} \tag{2.14}$$

After both ϕ and γ are obtained, values of the hyperparameters α and β can be estimated. This is done through finding the optimal lower bound of the log likelihood of:

$$l(\alpha, \beta) = \sum_{d=1}^M \log p(w_d | \alpha, \beta) \tag{2.15}$$

Both steps, obtaining variational parameters and hyperparameters, are essential in variational inference. These steps are better known as Expectation- Maximization (EM) algorithm. The classification of the steps are as follows:

Expectation Step: Finding optimal value for the variational parameters. This enables the expectation of log likelihood of the corpus.

Maximization Step: Finding the tightest lower bound of the log likelihood of $l(\alpha, \beta)$. This enables the maximization likelihood estimates.

2.6.2 Gibbs Sampling

This inference algorithm belongs to the family of Markov chain Monte Carlo (MCMC) . MCMC is a family of approximate inference algorithm that samples from a distribution in a Markov chain [76]. The Markov chain advantage is that it can sample from complex distributions, in this case the desired posterior distribution of LDA .By constructing a Markov chain for all possible distribution, the computation of the posterior distribution is possible through [80]:

$$E[f(s)]P \approx \frac{1}{N} \sum_{i=1}^N f(s^{(i)}) \quad (2.16)$$

Where:

P is the desired posterior distribution.

$f(s)$ is the desired expectation.

$f(s^{(i)})$ is i sample from the desired posterior distribution.

In the context of LDA, Gibbs sampling is used by stimulating high-dimensional distribution through the sampling of low-dimensional parameters. The parameters are conditioned on each other's value. The sampling is done iteratively until the sampled values closer to the target posterior distribution [42]. To achieve optimal posterior distribution (latent structure of document), Gibbs sampling sequentially sampled all parameters based on observed distribution conditioned to other states of distribution. To perform sampling, a full conditional distribution is needed to be define. This distribution is obtained through modification of the joint distribution of LDA.

$$P(z_i = j | z_{-i}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(.)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,j}^{(d_i)} + T\alpha} \quad (2.17)$$

Where:

$n_{-i,j}^{(.)}$ is the parameter ignoring the current $z_i - i$

T is the topic, where Tn is the total number of topics in the data

W is the word, where Wn is the total number of words in the data

$(.)$ are all other observed parameters.

Based on the full conditional distribution (16), the first ratio is the probability of w_i condition to topic j . The second ratio is the probability of j in document d_i . Thus, the conditional distribution assigned a word to a topic based on both likelihood of that word for a topic and the likelihood of the topic to the document. Based on both ratios, the approximate topic proportion for topic j is obtained by divide them by the sum of all topics T [42].

After the value of z is obtained for every word, the value of φ , word-topic distribution and θ , topic-document distribution can be estimated. This is done through:

$$\phi_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + W\beta} \quad (2.18)$$

$$\theta_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_j^{(d)} + T\alpha} \quad (2.19)$$

The results from both equations provide the distribution of unseen w_i of j and the distribution of unseen d_i of j . The entire Gibbs sampling approach to approximate posterior distribution can be summarized into the following algorithm. First, the algorithm initializes random topic z and assigns word token to it. For each sampling of the current word token, n is reduced by one to exclude the current iteration. Next, new topic is sampled from the distribution computed. This process is iterated for all N words in the document. In early iterations of the Gibbs sampling, it was found that they are poor estimates of the posterior distribution. Thus, burn-in period is introduced which removes the earlier iterations and only saves the iteration after the period. This provides a more accurate approximation to the desired posterior distribution. In the next subsection, the hyperparameters used by both the inference algorithms are explained in more detail.

2.6.3 Hyperparameters in LDA

Hyperparameters can be defined as parameters that are used to better fit the machine learning model to different dataset. The selection of appropriate hyperparameters is important as it can affect the performance of the model, given different datasets. In LDA, there are multiple hyperparameters that are important in the improving the fitness of LDA. Table 2.4 below exhibits the list of hyperparameters in LDA.

Table 2.4 LDA Hyperparameters

Hyperparameters	Description	Valid Values
Alpha	A prior estimate on the topic probability	Positive Float
Beta	A prior estimate on the word probability	Positive Float
Number of Topics	The number of topics for LDA to extract from the data	Positive Integer

Table 2.4 lists out three LDA hyperparameters that affects the fitness of the LDA model to a particular dataset. The first hyperparameter, alpha, is a prior estimate on the topic probability assigned by LDA. This affects the document-topic density of the final probability distribution. A high alpha value will result in more topics being considered in a document. Vice versa, a low alpha value will signify a lower number of topics in a document. The second hyperparameter, beta, is a prior estimate on the word probability. This affects the topic-word density of the final probability distribution. A high beta value will result in more words being considered as a topic, while a low beta value will result in topics with less words. The last hyperparameter, number of topics, is the number of topics to be extracted from the dataset.

These three hyperparameters values highly affects the resulting quality of topics extracted by LDA. Different values of hyperparameters are best for different sets of textual datasets. To optimize these values for a given dataset, hyperparameter tuning methods are employed. The hyperparameter tuning approaches are further discussed in Section 2.7.

2.7 Hyperparameter Tuning Approaches

Both Gibbs Sampling and Variational Bayesian Inference use hyperparameters to better fit to different sets of textual datasets[31, 33]. In general, hyperparameters are used to configure various aspects of the learning algorithm. The values can have strong effects on the resulting model and its performance. It is important to assign the best value for hyperparameters for a given algorithm or model. Hyperparameter search or optimization can be generally defined mathematically as:

$$h^* = \arg \min_h F(X; M, L) \quad (2.20)$$

Where:

h^* is the best hyperparameters value.

$\arg \min_h$ is the argument that minimizes.

$F(X; M, L)$ is the objective function or score, given set of data X and model M .

Based on the above equation, the objective function F takes a tuple of hyperparameters h and returns the associated loss L . The data X are given and the model M and loss function L are chosen. To search for the best hyperparameter value, researchers employ multiple different techniques or methods. A few popular methods used for hyperparameter optimization are rule-of-thumb, grid search and random search. Table 2.3 discusses the main strengths and weaknesses between the three main methods used for hyperparameter optimization.

Table 2.5: Approaches for Hyperparameter Tuning

Approach	Strengths	Weaknesses
Rules-of-thumb	<p>The set of hyperparameters are predefined based on successful trials of experiments.</p> <p>Lower computational requirement.</p> <p>Rules-of-thumb hyperparameter tuning is simple to implement.</p>	<p>The set of hyperparameters recommended might not be the best if tested on different dataset.</p> <p>The reproducibility of the experiments results conducted based on the recommended hyperparameters are often difficult to achieve.</p>
Grid Search	<p>Grid search is simple to implement.</p> <p>Grid search is parallelizable.</p> <p>Grid search is reliable in low dimensional data.</p>	<p>The search of hyperparameters is limited to the search space initialized.</p> <p>The time taken to find the best hyperparameters are highly dependent on the size of the search space.</p>
Random Search	<p>Random search is more efficient in high-dimensional space.</p> <p>Random search is better at achieving generalization in different dataset in comparison to grid search and rules-of-thumb method.</p>	<p>Random search might miss out on incremental improvement as it takes samples from the search space, instead of testing every combination.</p>

Based on Table 2.5, there are three main types of hyperparameters tuning, rules-of-thumb, grid search and random search. Rules-of-thumb is the simplest technique in hyperparameter tuning. Through this approach, the sets of hyperparameters used are based on existing research or experiments conducted. As it does not require any search space or experimentations to get the best hyperparameters, it does not use a lot of computational resources. However, the result achieved from the selected hyperparameters might not be the same as what has been achieved by other researchers. It is because there are different factors that can affect the results, such as the domain and complexity of the datasets. Based on this reason, rules-of-thumb approach might not be the best approach for hyperparameter tuning.

Grid search is a hyperparameter tuning technique which uses and defines a search space as a grid of hyperparameter values. Based on the defined search space, grid search will evaluate all the positions in the grid. This allows an exhaustive evaluation of all possible combination of hyperparameter values defined in the search space. As each combination of hyperparameter are independent from one to another, grid search can be parallelizable. The main weakness of grid search is that the time taken to find the best hyperparameter will be highly dependent on the size of the search space. Due to this factor, grid search is normally used in low dimensionality search space. Another drawback of grid search is that the set of hyperparameters taken into consideration in evaluation is limited to the ones defined in the search space. To cater to this shortcoming, random search is proposed.

Random search is a method that uses a bounded domain of hyperparameters instead of grid of hyperparameter value as its search space. Based on the bounded domain of hyperparameters, random search will randomly sample different set of hyperparameter values. This randomly sampled hyperparameter values are evaluated and the loss function of each testing are collected. This presents the major strength of random search method, which is that it is more efficient in high dimensionality space of hyperparameters. Based on this feature as well, random search is better at achieving generalization on different dataset. This is important as different datasets might require different sets of hyperparameters, which might not be within confined of grid search's

search space. However, random search might miss out on incremental improvements due to the random sampling of hyperparameters. Random search does not test all possible combinations of hyperparameters.

Based on findings from three different techniques of hyperparameter tuning, the two most promising techniques are grid search and random search. Grid search allows an exhaustive search of all possible combination of hyperparameters, in comparison to random search which only evaluate random samples of hyperparameter. On the other hand, random search is highly efficient in high dimensionality space in comparison to grid search which is only efficient in low dimensional search space. In this research, both techniques are incorporated into ASIM-LDA. This is to utilize both techniques strengths. Random search is utilized as an exploratory search and discovery of possible hyperparameters bound space. Once this is completed, a search space is defined based on the results obtained from the random evaluations. Based on this search space, grid search is employed to exhaustively evaluate all possible combinations of hyperparameters. To find the best hyperparameter for ASIM-LDA, an objective function must be defined. This is further discussed in Section 2.7.

2.8 Objective Function for Latent Dirichlet Allocation

Intrinsic evaluation of topic models ideally needs a manually annotated corpus with the topics; however, such annotations are very expensive to produce, and the gold standard topics reflect the subjectivity in the annotators' topic comprehension. Objective function for LDA has been developed to quantify the quality of topic models by measuring the coherence of words in each topic. There are multiple objective functions used in the intrinsic evaluation of how well the model fit to a particular dataset. These objective functions are described in Table 2.6.

Table 2.6: Objective Functions for Optimizing Latent Dirichlet Allocation

Objective Function	Strength	Weakness
Perplexity	It is a quantitative measure to compare different performance of models.	It does not correlate with the extrinsic evaluation score.
Topic Coherence	It is highly correlated to extrinsic evaluation scores, such as accuracy and precision.	The time taken to compute topic coherence score is more in comparison to perplexity score.

Perplexity is the result of one of the earliest works on intrinsically evaluating learned topics of LDA. To evaluate based on perplexity score, a model is learned on a collection of training documents, then the log probability of the unseen test documents is computed using that learned model. Usually perplexity is reported, which is the inverse of the geometric mean per-word likelihood. Perplexity is useful for model selection and adjusting parameters (number of topics K) and is the standard way of demonstrating the advantage of one model over another. However, perplexity score is not highly correlated with extrinsic evaluation score. Extrinsic evaluation score is score that is measured through human-in-the-loop. Based on this approach, human is used to interpret the results obtained from the learned topics. Due to this problem, topic coherence score is developed.

Topic coherence score is a measure of how coherent the words are residing in a particular learned topic. Through this evaluation, a more correlated score to an extrinsic evaluation can be obtained. As the computation of the topic coherence score requires it to take into consideration all the words residing in a topic, it generally takes a lot more time to compute in comparison to perplexity score.

2.9 Summary

Based on the literature review conducted, the main research gap is on selecting the right inference algorithm for LDA in any given textual dataset. It is noted that variational inference tends to be faster in most cases [37]. This is an important feature if user's intent to analyse large and complex data [63]. Gibbs sampling is slower as it approximates the posterior distribution via the application of stochastic transition operator for large number of times [81]. Even though its slower, Gibbs Sampling has the advantage of being asymptotically exact. This gives a closer and more accurate approximation of the posterior distribution in comparison to variational inference. Through these findings, there is a gap found in finding the right type of inference algorithms to be utilized with different complexity of textual data. This creates a need to utilise the strength of the approximate inference algorithm in inferring the robust posterior distribution. As each inference algorithm has their own hyperparameters to be selected, hyperparameters tuning approaches are explored. Based on literature study conducted, both grid search and random search are selected. Grid search is selected due to its ability to exhaustively search all possible combination of hyperparameters in a pre-defined search space. On the other hand, random search is more superior in high dimensional search space. This is mainly due to the nature of the search space of random search, which only specify the upper and lower bound of the hyperparameter values, instead of their exact values. These findings serve as the foundation for the development of the methodology. This methodology is developed to address the research problems that have been designed earlier. This also includes the proof of concept that provided a direction on the development of the final model. Next, the research timeline is discussed in terms of the phases involved during the entire project. Lastly, the entire pipeline of data processing utilizing the methodology is explained in detail.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the methodology developed for this research is discussed in detail. The methodology developed is based on the findings discussed in Chapter 2. Section 3.2 provides an overview of the research framework. This includes the activities conducted in this research. Section 3.2 outlines the main methodology developed in this research, Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA). In this section, the concepts of the methodology are explained. ASIM is developed with the basis of which it can intelligently select the appropriate inference algorithm for different set of data. This discussion covers the flowchart and the pseudocodes of the methodology. Section 3.3 introduces the overview of ASIM-LDA in general. Section 3.4 deliberates the design and implementation of ASIM for LDA. In this section, each component of ASIM-LDA, from data pre-processing until the final output of ASIM, are discussed. Section 3.5 exhibits the implementation of ASIM-LDA. Section 3.6 discusses the significant of the study and development of ASIM for LDA. Lastly, Section 3.7 summarizes the chapter.

3.2 Research Framework

This section details out the research framework for the design and development of Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA). The fundamental basis of ASIM-LDA is the development of two stages of optimization which are the hyperparameter optimization of individual inference algorithms and the selection filter between two optimized inference algorithms. This enables the selection of the best inference algorithm. Research framework is as depicted in Figure 3.1.

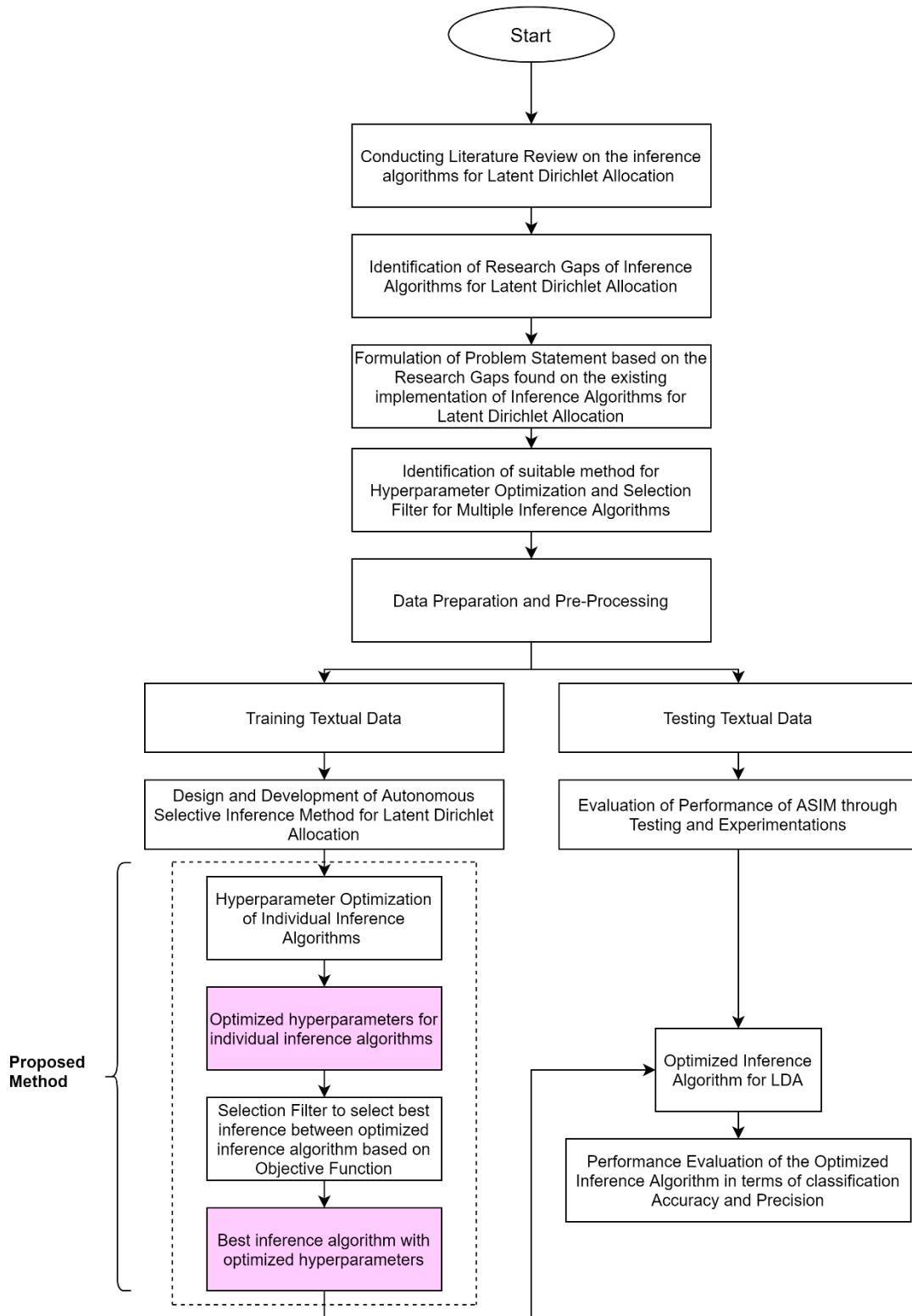


Figure 3.1: Research Framework for the Design and Development of Adaptive Selection of Inference Method for Latent Dirichlet Allocation (ASIM-LDA)

Based on Figure 3.1, there are five main stages in the Research Framework developed for ASIM. The first stage consists of the literature review on the inference algorithms used by LDA. This literature review is to study on the different inference algorithms, mainly the strengths and weaknesses of each of the inference algorithms used. In this literature study, two main inference algorithms are studied, namely Gibbs Sampling and Variational Bayesian Inference.

Based on the literature review conducted, research gaps have been identified in the existing implementations of the inference algorithms for LDA. There is a gap in identifying the best inference algorithm for the given input data. There are two main areas that the research aims to study, in solving the research gaps. First, in optimizing the hyperparameters of the individual inference algorithm. Hyperparameters are the parameters used by the inference algorithm to adjust their fit given the input data. Second, in selection of inference algorithm once each of the inference algorithms are optimized based on a defined objective function. Subsequently, the problem statement is derived based on the research gaps identified. Research questions and research objectives are developed to address the problems identified. Identification of suitable hyperparameter optimizing technique and selection filter is done based on further literature studies.

Subsequently, data preparation and pre-processing are conducted. In this stage, appropriate data is selected for the experiments. For this research, textual data is selected as it is the most suitable form of data for processing using LDA. Next, the selected textual data is pre-processed to ensure that it is suitable for processing by the developed method. The pre-processed data is split into two types, training data, and testing data.

After data pre-processing is completed, the next phase is the design and development of the proposed methodology, ASIM-LDA. ASIM-LDA consists of two main stages; the hyperparameter optimization of individual inference algorithm using combination of random search and grid search, and the selection filter to select the best inference algorithm based on the defined objective function, topic coherence score. This enables the selection of the best inference algorithm given any types of input textual data.

Lastly, the performance evaluation of the proposed methodology is performed. Performance evaluation is conducted through series of experiments with different types of textual data. Performance of the selected inference algorithm is defined as the best fit to the given input textual data based on the objective function defined.

3.3 Overview of Adaptive Selection of Inference Method for Latent Dirichlet

Allocation

Method proposed is the Adaptive Selection of Inference Method for LDA (ASIM-LDA). ASIM-LDA is an algorithm that adaptively select suitable inference algorithm for the respective textual datasets. This algorithm analyses the text data and select the appropriate inference algorithm to be used in inferring the topics from the data, whether it is Gibbs Sampling or Variational Bayesian Inference. Full flow of the proposed method is as depicted in Figure 3.2.

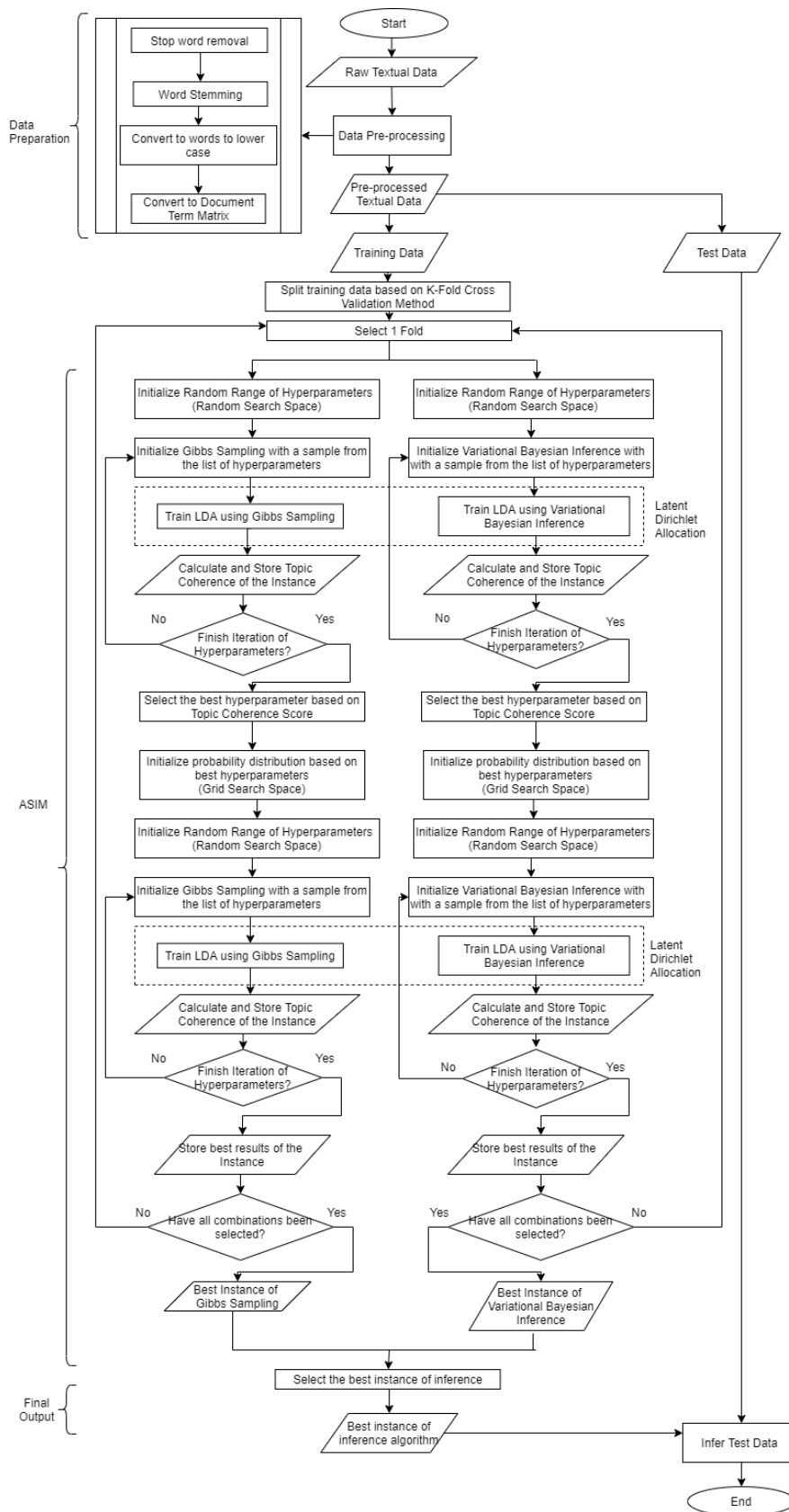


Figure 3.2: Overview of the Adaptive Selection of Inference Method (ASIM) for LDA

Based on Figure 3.2, ASIM-LDA consists of two main stages, which are data preparation and the ASIM-LDA stage. ASIM-LDA stage consists of two sections, which are the hyperparameter optimization of individual inference algorithm through combination of random search and grid search, and the selection filter between inference algorithms. Based on the results from the selection filter, the final output from the process is obtained. The main contribution of this research is the ASIM-LDA stage, which incorporates two layers of hyperparameter optimization for individual inference algorithms and the selection filter between inference algorithms. This stage allows the consideration different possible combinations of hyperparameters of each individual inference algorithm and selects the best inference algorithm based on these optimized hyperparameters. This combination of hyperparameter optimization techniques provides the flexibility of not having to initially define an exhaustive list of hyperparameters. The second module in ASIM-LDA is to select among the best instance of individual inference algorithm based on the objective function defined.

The proposed methodology, ASIM-LDA uses two inference algorithms which are Gibbs Sampling and Variational Bayesian Inference. Both inference algorithms are used as each of the inference algorithm is suitable for different types of datasets. The inner workings of ASIM-LDA are explained in more detail in Section 3.4. The input for ASIM-LDA is pre-processed textual data. Data pre-processing is explained in Section 3.4.3, after explanation of the main method. Output of ASIM-LDA is the best instance of inference algorithm which has two main characteristics, the best inference algorithm, and the best set of hyperparameters. This enables a more accurate representation of textual data using LDA.

3.4 Design of ASIM-LDA

ASIM-LDA consists of two stages of refinement. The two stages of refinement are the hyperparameter optimization of individual inference algorithms and the selection filter after the hyperparameter optimization. These two stages enable a more refined selection of inference algorithm which are adaptive to the textual data trained and tested. Figure 3.3 depicts the processes involves in ASIM.

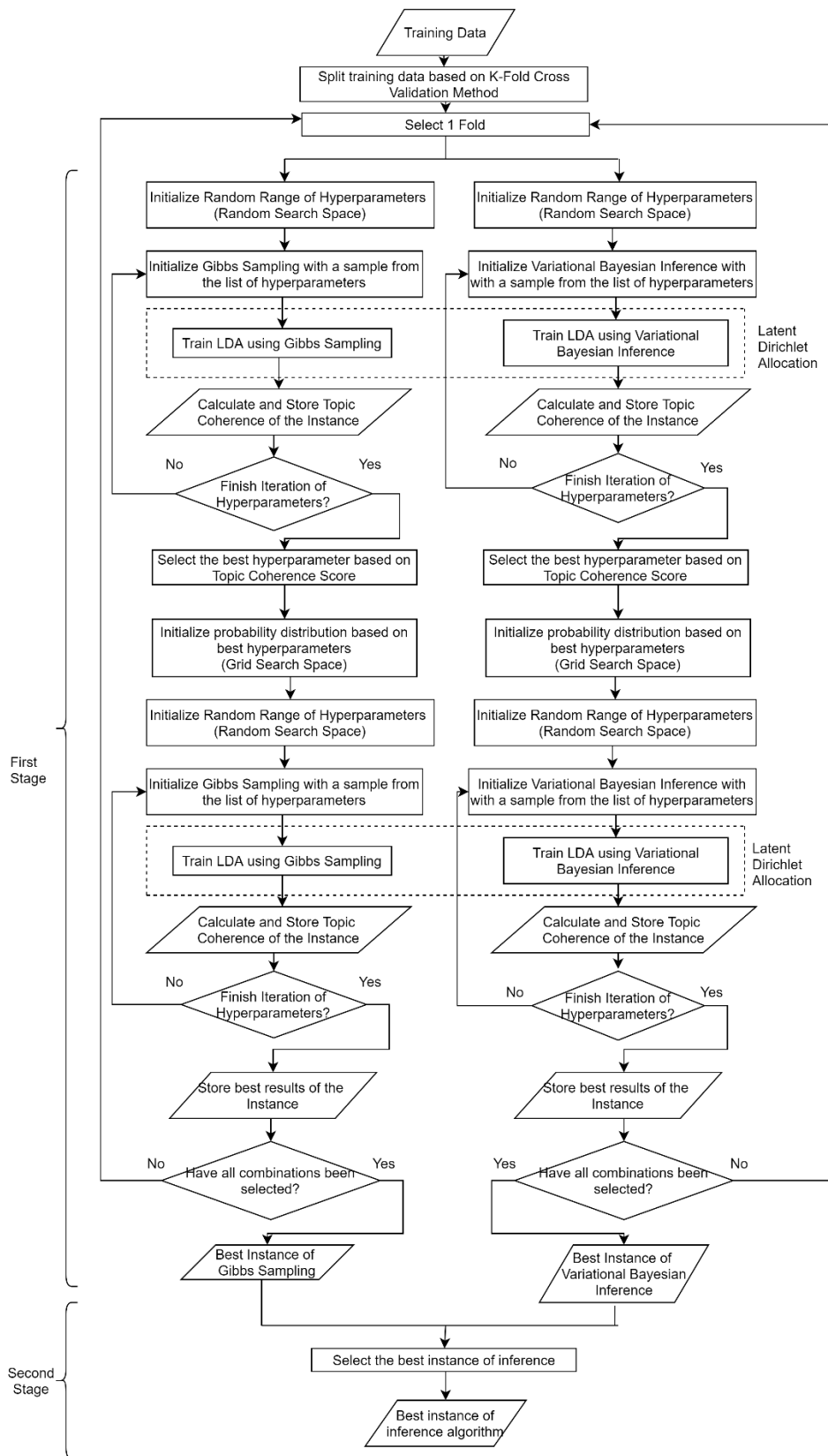


Figure 3.3: Adaptive Selective of Inference Method (ASIM)

Based on Figure 3.3, there are two main stage which are indicated by the first stage and the second stage. ASIM computes the topic coherence score for each of the individual inference algorithm according to the selected hyperparameters. Topic coherence score denotes the fitness of the inference algorithm to the input data. As different data fits differently to either Gibbs Sampling or Variational Bayesian Inference, the topic coherence scores of both the inference algorithms are considered. By incorporating the two different inference algorithms, the method can adapt to different types of data such as the complexity of the textual data and the size of the data. The best topic coherence scores for individual inference algorithms become for the second stage of ASIM. The second stage of ASIM selects the best topic coherence score between either of the inference algorithm and subsequently filter the best inference algorithm as the final model of LDA. Algorithm 3.1 explains in detail the entire process involves in the two stages of ASIM.

Input: Pre-Processed Data
Output: Topic Coherence Score

```

01 Initialization Random RangeHyperparameters = [alpha, eta, num_topic]
02 Foreach hyperparameters in RangeHyperparameters do
03     Train LDAGibbsSampling with hyperparameters
04     Compute topic_coherence for LDAGibbsSampling
05     Store topic_coherence and hyperparameters in list_GS
06     Train LDAVariationalBayesian with hyperparameters
07     Compute topic_coherence for LDAVariationalBayesian
08     Store topic_coherence and hyperparameters in list_VB
09 Foreach result in list_GS:
10     if index == 1:
11         best_res_GS = result
12         best_hyp_GS = hyperparameters
13     else:
14         if result > best_res_GS:
15             best_res_GS = topic_coherence
16             best_hyp_GS = hyperparameters
17 Foreach result in list_VB:
18     if index == 1:
19         best_res_VB = result
20         best_hyp_VB = hyperparameters
21     else:
22         if result > best_res_VB:
23             best_res_VB = result
24             best_hyp_VB = hyperparameters

```



```

25 Initialize disprob_VB based on best_hyp_VB
26 Initialize disprob_GS based on best_hyp_GS
27 Foreach hyperparameters in best_hyp_GS do
28     Train LDAGibbsSampling with hyperparameters on Pre-Processed Data
29     Compute topic_coherence for LDAGibbsSampling
30     Store topic_coherence and hyperparameters in list_GS
31 Foreach hyperparameters in best_hyp_VB do
32     Train LDAVariationalBayesian with hyperparameters
33     Compute topic_coherence for LDAVariationalBayesian
34     Store topic_coherence and hyperparameters in list_VB
35 Foreach result in list_GS:
36     if index == 1:
37         best_res_GS = result
38         best_hyp_GS = hyperparameters
39     else:
40         if result > best_res_GS:
41             best_res_GS = result
42             best_hyp_GS = hyperparameters
43 Foreach result in list_VB:
44     if index == 1:
45         best_res_VB = result
46         best_hyp_VB = hyperparameters
47     else:
48         if result > best_res_VB:
49             best_res_VB = result
50             best_hyp_VB = hyperparameters
51 if best_res_VB > best_res_GS:
52     best_res = best_res_VB
53 else if best_res_GS > best_res_VB:
54     best_res = best_res_GS

```

Algorithm 3.1: Algorithm of ASIM-LDA

Based on Algorithm 3.1, Line 01 depicts the initialization random hyperparameters used in ASIM. This initialized list is used to stores the list of sets of hyperparameters which are alpha, beta and the number of topics.

Line 02 to Line 50 shows the steps involve in the first stage of ASIM which is the hyperparameter optimization of individual inference algorithms. Within these lines, the two hyperparameter search algorithms are combined. First, the inference algorithms go through the random search phase, which uses the randomly initialized hyperparameters. Using these randomly initialized hyperparameters, both inference algorithms are trained, and the topic coherence scores are computed. Based on the best topic coherence

score from the random search, a new distribution of hyperparameters is initialized. The distribution is initialized based on the best hyperparameters selected. This enables the exploration of space of optimization for the inference algorithm. Section 3.4.1 contains the explanation of the hyperparameter optimization of individual inference algorithms.

Line 51 to Line 54 exhibits the second stage of ASIM. The second stage provides the selection of the best instance of inference algorithm based on the results from the first stage. 3.4.2 discusses the selection filter used to pick the best instance inference algorithm for LDA.

3.4.1 First Stage: Hyperparameter Optimization of Individual Inference Algorithms

First stage of ASIM is the hyperparameter optimization of individual inference algorithms. Each of the inference algorithms, either Gibbs Sampling or Variational Bayesian Inference has its own strengths and weaknesses depending on different types of datasets inputted into the algorithm. For example, Gibbs Sampling might perform better with small datasets but will be significantly slower if used in large datasets. However, there are certain cases where a well-defined and optimized hyperparameters for Variational Bayesian Inference algorithm outperforms unoptimized version of Gibbs Sampling. Thus, using ASIM, it explores each of possibility of sets of hyperparameters in each of the inference algorithm. The hyperparameter optimization of inference algorithm is as depicted in Figure 3.4.

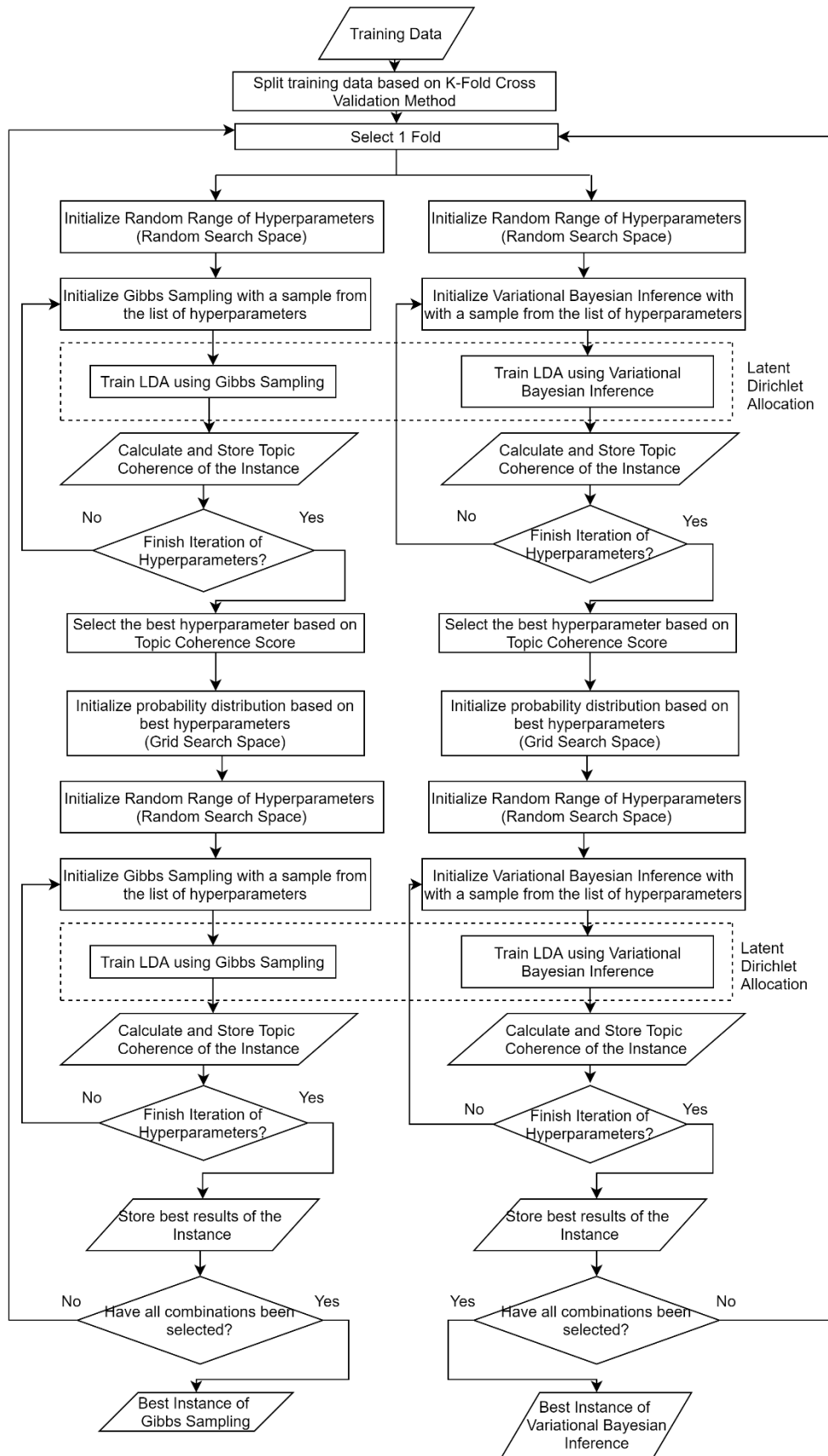


Figure 3.4: First Stage: Hyperparameter Optimization of Inference Algorithms

Figure 3.4 depicts the processes involve in the first stage of ASIM which the inner optimization of inference algorithm is. ASIM performs inner optimization of inference algorithms by iteratively train based on the input textual data using different sets of hyperparameters. These sets of hyperparameters are randomly initialized in the beginning of the first stage. Once all the initial hyperparameters are exhausted, a new batch of the hyperparameters are initialized based on the best hyperparameters selected from the first round. Section 3.3.1.1 discusses the hyperparameter initialization process.

3.4.1.1 Hyperparameter Initialization of ASIM-LDA

To perform the hyperparameter optimization, ASIM-LDA employs two steps approach in searching the best set of hyperparameters. The first step, random search, uses random initialization of hyperparameters. This step provides the ability of not having to set a fixed set of hyperparameters in the beginning. The hyperparameters are initialized based on the probability distribution listed in Table 3.1.

Table 3.1: Probability Distribution of Hyperparameter

Hyperparameter	Probability Distribution
Alpha	Log-Normal Distribution
Beta	Log-Normal Distribution
Number of Topics	Discrete Uniform Distribution

Based on Table 3.1, the hyperparameters Alpha and Beta use log-normal probability distribution. This is because hyperparameters Alpha and Beta only accepts positive float numbers. Hyperparameters for number of topics is initialized based on the discrete

uniform distribution. The utilization of discrete uniform distribution to randomly generate the hyperparameters is since number of topics only accepts integer value. Based on the values generated, ASIM-LDA randomly samples from the initialized list to train on the dataset. Once the first step is completed, the best hyperparameters are used as the basis of next generation of search, the grid search. To construct the search space for grid search, the generation of hyperparameters are configured based on Table 3.2.

Table 3.2: Grid Search Hyperparameters Generation Setting

Hyperparameter	Probability Distribution	Basis from Random Search
Alpha	Log-Normal Distribution	Best Alpha is set as the mean of the Log-Normal Distribution
Beta	Log-Normal Distribution	Best Beta is set as the mean of the Log-Normal Distribution
Number of Topics	Discrete Uniform Distribution	Best number of topics is set as the mean of the Discrete Uniform Distribution

Based on Table 3.2, the hyperparameters Alpha and Beta are re-initialized using the same probability distribution used in the first step, log-normal distribution. In this step, the best Alpha and Beta are used respectively as the mean of the generation of log-normal distribution. This provides the ability to discover new range of hyperparameters which potentially be a better fit to the dataset being trained. The same applies to number

of topics, where the best number of topics from the first step is used as the mean of the generation of discrete uniform distribution.

To determine the best set of hyperparameters, it is important to calculate the objective function. Objective function used in this hyperparameter optimization is the topic coherence. Section 3.4.1.2 discusses the objective function for ASIM-LDA.

3.4.1.2 Objective Function for ASIM-LDA

Objective function is a function that the algorithm evaluates itself against different values of hyperparameters. It gives a score of best fit of the algorithm to the given dataset. The objective function for the first stage is defined as follows:

$$(a, b, c) = \underset{(a,b,c)}{\arg \max} \text{topicCoherence}(LDA[\text{inference}(a, b, c)]) \quad (3.1)$$

Notations used in this Equation 1 are listed in Table 3.3.

Table 3.3: Notations for Objective Function

Notation	Description
<i>a</i>	alpha (hyperparameter)
<i>b</i>	beta (hyperparameter)
<i>c</i>	Number of topics (hyperparameter)
<i>topicCoherence</i>	topic coherence score computation
<i>inference</i>	selection of inference algorithm between [Gibbs Sampling, Variational Bayesian Inference]

Topic coherence score is derived to measure the fitness of the inference algorithm to different data. In this model development, topic coherence score

enables the algorithm to assess whether the best instance of the inference algorithm is selected for the given textual datasets.

3.4.1.3 K-Fold Cross Validation

To ensure that the model is not overfitted to the input data, the method used for hyperparameter optimization is the K-Fold Cross Validation technique. One example visualization of the result from K-Fold Cross Validation is as depicted in Figure 3.5.

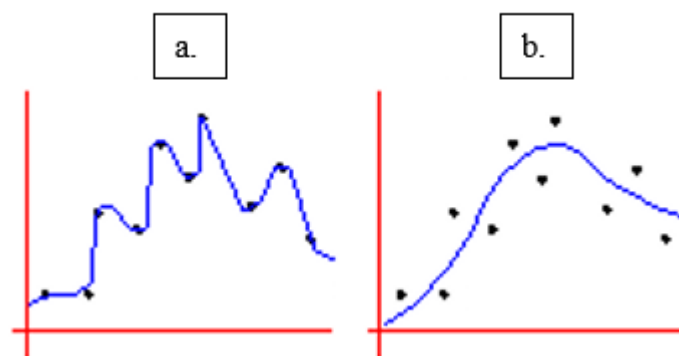


Figure 3.5: Cross Validation

Based on Figure 3.5, K-Fold Cross Validation able to reduce the overfitting of the inference algorithm to the training data. In this example, *graph a* show an overfitted model which K-Fold Cross Validation denotes it as not a good model. For *graph b*, K-Fold Cross Validation denotes it as a good model, and it does not overfit to the training data. This ensures that the inference algorithm will be able to generalize to new unseen data and not only limited to the existing training data. K-Fold Cross Validation technique provides an estimation of the ability of the inference algorithm against unseen data. This would enable a fair evaluation of the objective function as the inference algorithm would not see the whole set of data during training. Instead, the inference algorithm is trained using the divided training data and evaluated against the unseen dataset.

Figure 3.6 illustrates how the K-Fold Cross Validation splits the data into individual folds.

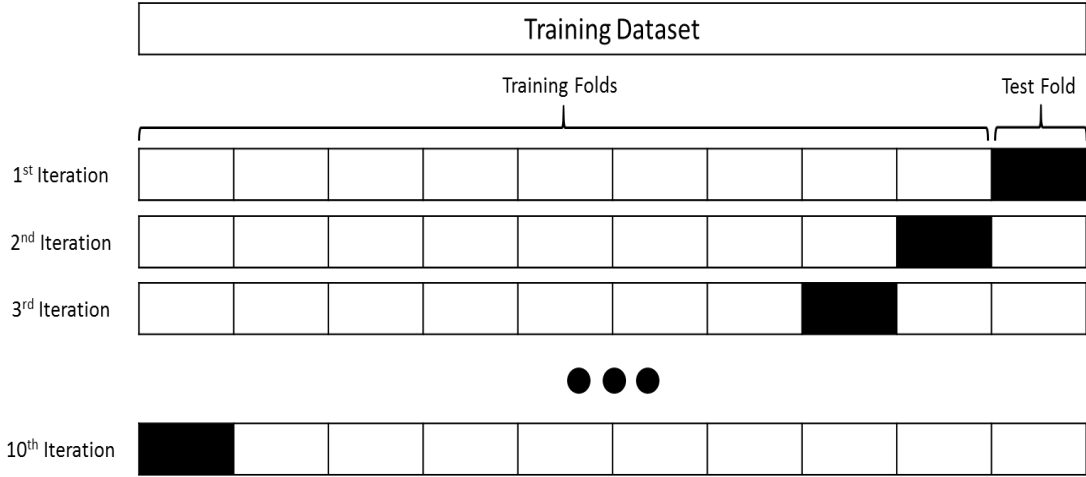


Figure 3.6: Example of K-Fold Cross Validations Folds

Based on Figure 3.6. the entire training dataset is split into 10 separate folds. For the first iteration, the algorithm only trains on the first 9 folds and held out the last fold for testing. This process repeats for the entire folds, to ensure that each of the held-out fold is tested. In this experiment, there is 10 iterations in total to complete the testing of the entire dataset. For each of the iteration, the objective function calculates the topic coherence score. After completing all iterations, the topic coherence scores are aggregated. To understand how K-Fold Cross Validation works, the algorithm for K-Fold Cross Validation technique is depicted in Algorithm 3.3.

Input: Training Dataset $D = (x_1, y_1), \dots, (x_m, y_m)$
Set of Hyperparameter Values $\Theta = \theta_1, \dots, \theta_n$
Inference Algorithm A
Integer k

Output: Best hyperparameter values $\theta^* = \operatorname{argmax}_{\theta} [\operatorname{score}(\theta)]$
Best instance of inference $h_{\theta^*} = A(D; \theta^*)$

```

01 Partition  $D$  into  $D_1, \dots, D_k$ 
02 Foreach  $\theta \in \Theta$  do
03   Foreach  $i = 1 \dots k$ 
04      $h_{i,\theta} = A(D_i; \theta)$ 
05    $\operatorname{score}(\theta) = \frac{1}{k} \sum_{i=1}^k A_{S_i}(h_{i,\theta})$ 
06 Return  $\theta^*$  and  $h_{\theta^*}$ 

```

Algorithm 3.3: Algorithm for K-Fold Cross Validation

Based on Algorithm 3.3, the training of the inference algorithm is done on multiple partitions of the training data. Line 01 denotes the partitioning of the training data into

k partitions of data. Line 02 repeats the function inside for every set of hyperparameters value in the list of hyperparameters. Line 03 repeats the function inside for each partition of the data. Line 04 calculates the objective function defined based on the results from the inference algorithm. Line 05 aggregates and find the average score of the objective function based on each partition of the data. After each value of hyperparameters are explored, K-Fold Cross Validation returns the best set of hyperparameters value and the best instance of the inference algorithm. The best set of hyperparameters is the one that returns the best topic coherence score given the data. After the best instance of individual inference algorithms is defined, ASIM-LDA proceeds to the Second Stage which is the selection filter among the results from the individual inference algorithms.

3.4.2 Second Stage: Selection Filter for Best Instance of Inference Algorithms

The second stage of refinement in ASIM deals with selecting the best inference algorithm among Gibbs Sampling and Variational Bayesian Inference. This refinement enables the selection of not only the best set of hyperparameters for individual inference algorithm but also the best inference algorithm given the best set of hyperparameters. The selection of optimized inference algorithms enables the selection of best among the best inference algorithm given the input textual data. Figure 3.7 depicts the flow of the selection filter for the individual inference algorithms.

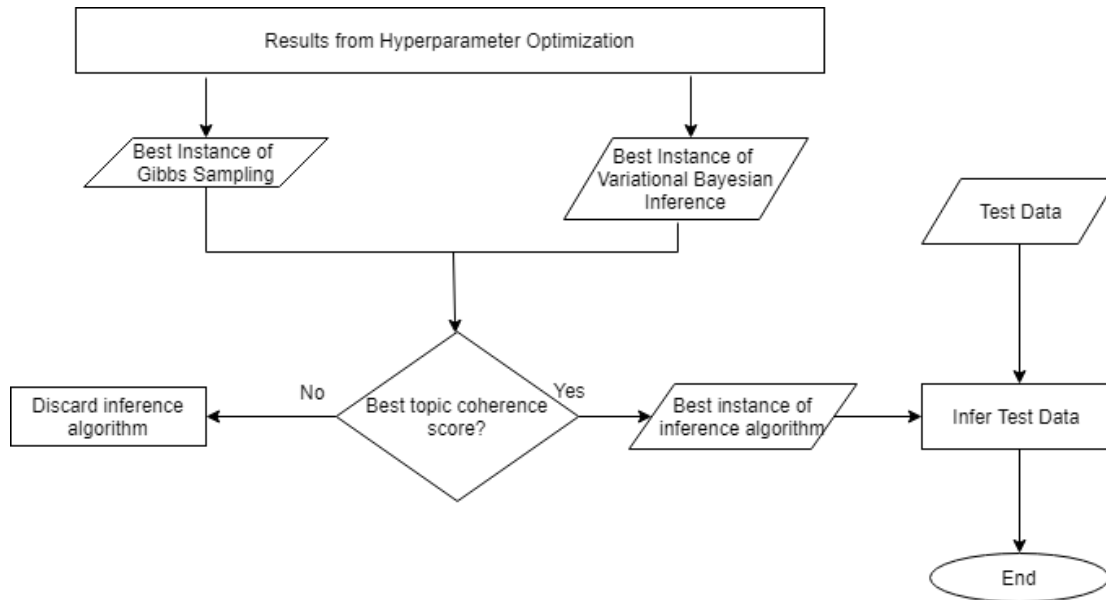


Figure 3.7: Second Stage: Selection Filter

Based on Figure 3.7, best instances for each individual inference algorithms, Gibbs Sampling and Variational Bayesian Inference are used as input. An example of the input for this selection filter is as in Table 3.15.

These optimized individual inference algorithms are compared among each other based on the earlier defined objective function results. In the filter, the inference algorithm which has better results in terms of the objective function is selected. The best result is defined as the one that achieve a higher topic coherence score among the two. Subsequently, this selected instance of inference algorithm is used to infer and model the test data. This test data went through the same pre-processing which the training data used as well. Section 3.4.3 explains on the data pre-processing involves in the textual data used.

3.4.3 Data Pre-Processing

Raw data are pre-processed prior fitting into ASIM. Pre-processing ensures noises in text data used are reduced. After the reduction of noises, the processing of the data using the main model will generally be more efficient. Flow of pre-processing steps are depicted in Figure 3.8.

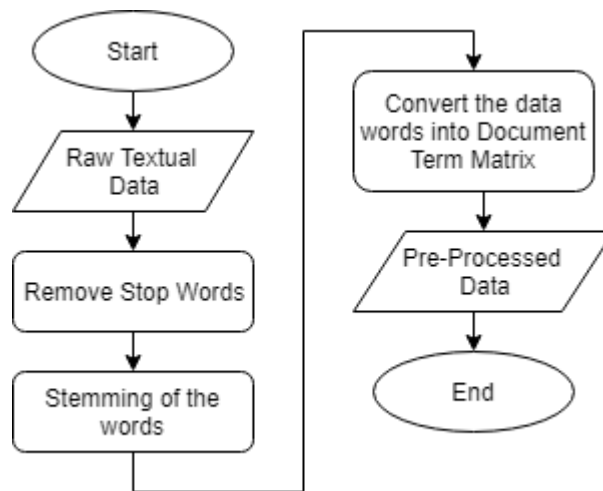


Figure 3.8: Data Pre-Processing

3.4.4 Stop Words Removal

Stop words are removed from the content. Stop words are defined as words that do not have or bring any additional meaning to the context of the content. In this scope of research, stop words used are limited to those defined in English such as function words (e.g., *the, is, a*). As stop words commonly occur in the content of the text data, it must be removed to prevent more noise entering the topic modelled later. If stop words are not ignored in the modelling process, it might lead to dummy topics being created which consists mostly of this contextless words.

As stop words are required to be defined first, an existing dictionary of English stop words is used. The *NLTK* library for stop words is used in this removal process. The process can be defined as in Algorithm 3.2.

```

01 Initialization of NLTK Stop Words Library as stop_words
02 Processed_words = []
03 For word in input_data:
04     If not word in stop_words:
05         Processed_words.append(word)
06 Return processed_words

```

Algorithm 3.2: Stop Words Removal Algorithm

Based on Algorithm 3.2, the stop words removal algorithm iteratively read and evaluate each word in the input data. A new list is instantiated to store the processed words. The algorithm populates the *processed_words* list with words that is not in the *stop_words* defined. The *processed_words* list is inputted to the next process of data pre-processing.

3.4.5 Stemming of the Words

After removal of stop words, stemming is applied to the remaining words. Stemming is the process of reducing the words to its root words or base form. An example of words being stemmed to its base form is shown in Table 3.4.

Table 3.4: Example of Stemmed Word (Argue)

Original Word	Base Form
arguing	argue

The word *argue* still denotes the same meaning. Stemming prevents the topic modelling algorithm seeing the same meaning words as a different word, which will lead to redundant words in their corresponding topics. In this research, porter stemmer is used. The usage of stemming is also coupled together with the lemmatization algorithm. This is to correct any accidental change of the stemming process done towards the words. Lemmatization provides dictionary form of the word if the stemmed

word does not match with any of the words found in the English dictionary used by the lemmatization algorithm.

Subsequently, n-gram is applied to the text data, specifically bigram. Bigram analysis combines two words that co-occur together regularly in the document into one word. This is to ensure the context of the words are intact before transformation process. There is possibility of loss of context if n-gram is not applied as the data will be transformed into bag of words.

Next, the text documents are transformed into document term matrix (DTM). DTM is a representation of the bag of words which in turn represent the data. This conversion is done to obtain the vectorized form of the data. Vectorized form of the data is the input for LDA. The output of this process is the DTM; which contains the ID of the words and their corresponding occurrences in the document and the dictionary; which contains the unique words and their corresponding ID. An example of DTM is as depicted in Figure 3.9.

	intelligent	applications	creates	business	processes
Doc 1	2	1	1	1	1
Doc 2	1	1	0	0	0
Doc 3	0	0	0	1	0

Figure 3.9: Document Term Matrix

Based on Figure 3.9, three documents (*Doc 1*, *Doc 2*, *Doc 3*) are converted into DTM format. In this example, there are five unique words occurred (*intelligent*, *applications*, *creates*, *business*, *processes*). The numbers in the matrix represents the number of corresponding word occurrences in the respective doc. For example, the word *intelligent* occurs twice in *Doc 1* but only once in *Doc 2* and did not occur in *Doc 3*. DTM is the final form of pre-processed data. This will be the input data for ASIM.

3.5 Implementation of ASIM-LDA

In this section, the implementation of ASIM-LDA is discussed in more clarity in terms of how the method works given a textual input data. Section 3.5.1 describes the

input textual data used in this research. Section 3.5.2 elaborates on how the pre-processing works and the data looks like before and after the pre-processing. Section 3.5.3 provides overview on how ASIM works given the pre-processed textual data. This subsection also presents the output of ASIM which is the selection of best inference algorithm based on the topic coherence score.

3.5.1 Dataset

In this research, three different textual datasets are used. These datasets are differentiated based on their complexity. Complexity, as defined in Section 3.4.1, is measured based the count of pre-processed words in the dataset. Pre-processed words only consist of unique tokenized words, in comparison of the raw count of words which contains noises and repetition of words. Different degree of complexity of data is used to ensure fair evaluations of the proposed method. Details on the datasets used in the experiments are summarized in Table 3.5.

Table 3.5: Overview of the Datasets[82]

Dataset Name	Dataset Complexity	Count of Words	Count of Pre-Processed Words	Percentage of Noise Removed
20 Newsgroups	High	342 521	327 632	4.4%
Twitter Airline Sentiment	Medium	173 446	166 478	4.07%
Spam Data	Low	86 335	80 583	6.6%

Based on Table 3.5, three datasets that are used are the 20 Newsgroups, Twitter Airline Sentiments and Spam Data. These three datasets are denoted as High

Complexity, Medium Complexity and Low Complexity, respectively. As all the dataset in their raw formats contain noises or anomalies, they were first pre-processed. The pre-processed steps involved are removal of the stop words, performing bigram analysis and lemmatization. Pre-processing ensures that the datasets are cleaned and only contains the truly unique tokenized words. For all the three datasets, the average percentage of noise removed is 5.02%. An example screenshot of the raw dataset in comparison with the pre-processed dataset is illustrated in Figure 3.10.

text	preprocessed
@AmericanAir thank you we got on a different f...	[thank, get, different, flight, chicago]
@AmericanAir leaving over 20 minutes Late Flig...	[leave, minute, late, flight, warning, communi...
@AmericanAir Please bring American Airlines to...	[bring, american_airline, blackberry]
@AmericanAir you have my money, you change my ...	[money, change, flight, answer, phone, suggest...
@AmericanAir we have 8 ppl so we need 2 know h...	[ppl, need, know, many, seat, flight, put, sta...

Figure 3.10: Screenshot of Original Data and Pre-processed Twitter Airline Sentiment Data (Medium Complexity)

Based on Figure 3.10, the raw dataset is grouped under the text column and the pre-processed dataset is grouped under the pre-processed column. Pre-processed dataset contains only unique tokenized words which are ready for further processing. Next, pre-processed dataset is split into train and test datasets. To split the datasets, K-Fold cross validation method is used. As mentioned in Section 3.4, K-Fold cross validation is a method to validate a model through evaluating the model's ability to predict new unseen data. For each of the different folds of the dataset, the datasets are pre-processed. K-fold cross validation is used to avoid biasness.

The datasets used in these experiments includes respective labels. The labels are important for evaluating classification performance of the optimized model. Table 3.6 depicts the label class for each of the datasets.

Table 3.6: Labels for Each of the Datasets

Dataset Name	Label Class	Labels	Number of Labels
20 Newsgroups	News Categories	'rec.autos','comp.sys.mac.hardware', 'rec.motorcycles','misc.forsale', 'comp.os.ms-windows.misc', 'alt.atheism','comp.graphics', 'rec.sport.baseball', 'rec.sport.hockey','sci.electronics', 'sci.space','talk.politics.misc', 'sci.med','talk.politics.mideast', 'soc.religion.christian','comp.windows.x', 'comp.sys.ibm.pc.hardware', 'talk.politics.guns','talk.religion.misc', 'sci.crypt'	20
Twitter Airline Sentiment	Sentiment Class	Positive, Negative, Neutral	3
Spam Data	Is Spam	Spam, Not-Spam	2

Based on Table 3.6, the 20Newsgroups dataset contains the label class of news categories. The news categories denote the types of news that were reported. Twitter airline sentiment dataset contains the airline sentiment label class. As this is a Twitter dataset, the labels consist of positive, neutral, and negative sentiment labels. Spam data has the least number of labels in its label class, which are only spam and non-spam. The label class is generated based on a spam filter. The raw datasets are stored as comma separated value (.csv) files. These csv files consist of the content of the datasets and their respective labels.

The raw data is extracted in the form of csv. This raw data consists of textual data which have their original topics tagged with their respective data. An example of the textual data extracted is as depicted in Figure 3.11.

	content	target	target_names
From: lerxst@wam.umd.edu (where's my thing)\nSubject: WHAT car is this!?\nNntp-Posting-Host: rac...		7	rec.autos
From: guykuo@carson.u.washington.edu (Guy Kuo)\nSubject: SI Clock Poll - Final Call\nSummary: Fi...		4	comp.sys.mac.hardware
From: irwin@cmptrc.lonestar.org (Irwin Arnstein)\nSubject: Re: Recommendation on Duc\nSummary: W...		8	rec.motorcycles
From: tchen@magnus.acs.ohio-state.edu (Tsung-Kun Chen)\nSubject: ** Software forsale (lots) **\n...		6	misc.forsale
From: dabl2@nlm.nih.gov (Don A.B. Lindbergh)\nSubject: Diamond SS24X, Win 3.1, Mouse cursor\nOrg...		2	comp.os.ms-windows.misc
From: a207706@moe.dseg.ti.com (Robert Loper)\nSubject: Re: SHO and SC\nNntp-Posting-Host: sun278...		7	rec.autos
From: kimman@magnus.acs.ohio-state.edu (Kim Richard Man)\nSubject: SyQuest 44M cartrifge FORSALE...		6	misc.forsale
From: kwilson@casbah.acns.nwu.edu (Kirtley Wilson)\nSubject: Mirosoft Office Package\nArticle-I...		2	comp.os.ms-windows.misc
Subject: Re: Don't more innocents die without the death penalty?\nFrom: bobbe@vice.ICO.TEK.COM (...)		0	alt.atheism
From: livesey@sointze.wpd.sgi.com (Jon Livesey)\nSubject: Re: Genocide is Caused by Atheism\nOrg...		0	alt.atheism
From: dls@aeg.dsto.gov.au (David Silver)\nSubject: Re: Fractal Generation of Clouds\nOrganizatio...		1	comp.graphics
Subject: Re: Mike Francesa's 1993 Predictions\nFrom: gajarsky@pilot.njin.net (Bob Gajarsky - Hob...		9	rec.sport.baseball
From: jet@netcom.Netcom.COM (J. Eric Townsend)\nSubject: Re: Insurance and lotsa points...\nIn-R...		8	rec.motorcycles
From: gld@cunixb.cc.columbia.edu (Gary L Dare)\nSubject: Re: ABC coverage\nNntp-Posting-Host: cu...		10	rec.sport.hockey
From: sehari@iastate.edu (Babak Sehari)\nSubject: Re: How to the disks copy protected.\nOriginat...		12	sci.electronics

Figure 3.11: Example of Raw Textual Dataset

Based on Figure 3.11, the raw textual dataset is segregated based on its original topics (*target_names*). It is important that the contents are separated from the topics tagged as the topics tagged would not undergo the data pre-processing steps. The characteristics of the textual input data can be described as in Table 3.7.

Table 3.7: Textual Dataset Characteristics

Characteristics	Description
Complexity	Complexity is described as the number of individual unique words inside the training/testing textual dataset [44, 45].
Size	Size is measure by the number of rows of training/testing textual dataset.
Topics	Topics is defined as the semantic clusters of the documents.

3.5.2 Data Pre-processing

In this stage, the extracted raw textual data is pre-processed before it is processed by ASIM. First, the textual data is split into individual tokens. This process eases further pre-processing and processing steps in this methodology. An example of tokenization is as in Table 3.8:

Table 3.8: Tokenized Data

Original Textual Data	Tokenized Data
I am driving to work	"I", "am", "driving", "to", "work"

Next process involves the removal of stop words. The list of stop words used extracted from the NLTK library in Python. Some examples of the stop words used in this process are as listed in Figure 3.12:

```
[ "i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your",
  "yours", "yourself", "yourselves", "he", "him", "his", "himself", "she", "her",
  "hers", "herself", "it", "its", "itself", "they", "them", "their", "theirs",
  "themselves", "what", "which", "who", "whom", "this", "that", "these", "those",
  "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had",
  "having", "do", "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or",
  "because", "as", "until", "while", "of", "at", "by", "for", "with", "about",
  "against", "between", "into", "through", "during", "before", "after", "above",
  "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "under",
  "again", "further", "then", "once", "here", "there", "when", "where", "why", "how",
  "all", "any", "both", "each", "few", "more", "most", "other", "some", "such", "no",
  "nor", "not", "only", "own", "same", "so", "than", "too", "very", "s", "t", "can",
  "will", "just", "don", "should", "now" ]
```

Figure 3.12: Example of Stop Words List in NLTK

Based on the stop words list from the NLTK library, each of the stop words are removed from the tokenized string. An example of the tokenized words without stop word is depicted in Figure 3.13.

Tokenized Words	Stop Words Removed Words
“I”, “am”, “driving”, “to”, “work”	“driving”, “work”

Figure 3.13: Removed Stop Words

Subsequently, the tokenized words without stop word are further pre-processed with stemming. As stemming provides the morphological variants of the base words, it will reduce the word to a root word. An example based on the tokenized without stop words is as shown in Figure 3.14.

Stop Words Removed Words	Stemmed Word
“driving”, “work”	“driv”, “work”

Figure 3.14: Stemmed Words

Based on Figure 3.14, the word *driving* is stemmed into *driv*. This also shows the limitation of stemming. It provides a crude heuristic process which removes the ends of the words processed. Sometimes, through this action, it also removes the derivational affixes. This renders the word to be wrongly spelled. To fix this, lemmatization is applied to the each of the words. An example of lemmatized words are shown in Figure 3.15.

Stop Words Removed Words	Lemmatized Word
“driv”, “work”	“drive”, “work”

Figure 3.15: Lemmatized words

Figure 3.15 shows the word *driv* after lemmatization process. As lemmatization lookup the root word in a defined vocabulary, this provides an accurate representation of root word. Next, bigram analysis is applied to lemmatized words. An example of a bigram word after analysis is depicted in Figure 3.16.

Original Words	Bigram Word
“rainy”, “day”	“rainy_day”

Figure 3.16: Bigram words

Based on Figure 3.16 , if the words *rainy* and *day* occur together regularly in the documents, both words are concatenated together to become *rainy_day*. After the bigram analysis, the pre-processing steps are complete and ready for the processing by ASIM.

3.5.3 ASIM-LDA

After pre-processing steps applied towards the raw textual dataset, the pre-processed dataset is processed by ASIM. ASIM, which consists of two stages, adapts to the dataset, and iteratively refined its processes to ensure best fit to the dataset. Section 3.4.3.1 and 3.4.3.2 explained how the processes involved in ASIM.

3.5.3.1 First Stage: Hyperparameter Optimization of Individual Inference Algorithm

In the first stage, ASIM iterates through all possible combinations of hyperparameters for each of the inference algorithms. The first part of the first stage consists of randomly initializing the range of hyperparameters, which consists of ranges values of alpha, number of topics and beta. List of ranges of hyperparameters are as listed in Table 3.9.

Table 3.9: Random Initialization of Hyperparameters (Random Search)

Hyperparameters	Range
Alpha	0.09,0.18,0.11,0.98,0.24,0.47,0.74,0.88,0.12,0.05
Beta	0.13,0.42,0.06,0.18,0.04,0.07,0.98,0.85,0.21,0.12
Number of Topics	10,20,30,40,50,60,70,80,90,100

Based on the list initialized, ASIM-LDA randomly selects combinations of hyperparameters to initialize the individual inference algorithms. An example of initialization of inference algorithm is listed in Table 3.10.

Table 3.10: Initialization of Individual Inference Algorithms

Inference Algorithm	Hyperparameters
Variational Bayesian Inference	Alpha: 0.09 Beta: 0.13 Number of Topics: 10
Gibbs Sampling	Alpha: 0.09 Beta: 0.13 Number of Topics: 10

Referring to Table 3.10, each inference algorithm is initialized with their respective hyperparameters. After initialization, each inference algorithm trains the pre-

processed textual dataset. This training process outputs a topic coherence score. An example of the output after one training process is as in Table 3.11.

Table 3.11: Output of a Single Training Process

Inference Algorithm	Hyperparameters	Topic Coherence Score
Variational Bayesian Inference	Alpha: 0.09 Beta: 0.13 Number of Topics: 10	0.33
Gibbs Sampling	Alpha: 0.09 Beta: 0.18 Number of Topics: 10	0.27

The training process iterates for each combination of hyperparameters based on the earlier list of hyperparameters initialized. After ASIM-LDA finishes training each hyperparameters, it stores the computed topic coherence scores in a list. An example of the final list of topic coherence scores is as listed in Table 3.12.

Table 3.12: Final List of Topic Coherence Score (Random Search)

Inference Algorithm	Hyperparameters	Topic Coherence Score
Variational Bayesian Inference	Alpha: 0.09 Beta: 0.13 Number of Topics: 10	0.33

	Alpha: 0.11 Beta: 0.06 Number of Topics: 30	0.58
Gibbs Sampling	Alpha: 0.09 Beta: 0.13 Number of Topics: 10	0.20
	Alpha: 0.18 Beta: 0.85 Number of Topics: 30	0.50

Based on Table 3.12, ASIM-LDA selects the best topic coherence score for the individual inference algorithm. The hyperparameters associated with the best topic coherence score are used as the basis of the grid search's search space. A sample of the generation of grid search's search space is as in Table 3.13.

Table 3.13: Initialization of Hyperparameters (Grid Search)

Hyperparameters	Range
Alpha (Variational Bayesian Inference)	0.02, 0.003, 0.10, 0.11, 0.24, 0.31, 0.39
Alpha (Gibbs Sampling)	0.07, 0.10, 0.15, 0.18, 0.21, 0.33, 0.37
Beta (Variational Bayesian Inference)	0.01, 0.005, 0.03, 0.06, 0.08, 0.10, 0.12
Beta (Gibbs Sampling)	0.23, 0.37, 0.69, 0.85, 0.98, 0.86, 0.87

Number of Topics (Variational Bayesian Inference and Gibbs Sampling)	5, 11, 23, 30, 35, 46, 59
--	---------------------------

Based on Table 3.13, ASIM-LDA iteratively selects different combinations of hyperparameters to be used in training the LDA. The training process is the same as the one conducted the in the random search. For each of the iteration of different combination, the topic coherence scores are calculated. An example of the final list of topic coherence scores is as listed in Table 3.14.

Table 3.14: Final List of Topic Coherence Score (Grid Search)

Inference Algorithm	Hyperparameters	Topic Coherence Score
Variational Bayesian Inference	Alpha: 0.02 Beta: 0.21 Number of Topics: 35	0.65
	Alpha: 0.11 Beta: 0.06 Number of Topics: 30	0.58
Gibbs Sampling	Alpha: 0.18 Beta: 0.98 Number of Topics: 59	0.51
	Alpha: 0.18 Beta: 0.85	0.50

	Number of Topics: 30	
--	----------------------	--

Based on Table 3.14, ASIM-LDA selects the best hyperparameters for individual inference algorithm based on the topic coherence score. In this example, the grid search improves the topic coherence score for both Variational Bayesian inference as well as for the Gibbs Sampling. This selection becomes the basis of the second stage of ASIM-LDA.

3.5.3.2 Second Stage: Selection Filter

The input used in the second stage of ASIM-LDA consists of the results from the first stage. An example of the input for the second stage is as depicted in Table 3.15.

Table 3.15: Input for Second Stage: Selection Filter

Inference Algorithm	Hyperparameters	Topic Coherence Score
Variational Bayesian Inference	Alpha: 0.02 Beta: 0.21 Number of Topics: 35	0.65
Gibbs Sampling	Alpha: 0.18 Beta: 0.98 Number of Topics: 59	0.51

Based on Table 3.15, there three inputs for the selection filter from the first stage. ASIM-LDA only uses one of the inputs from the table, which is the topic coherence

score. ASIM-LDA compares the topic coherence score for individual inference algorithm and pick the best score. Based on this example, ASIM-LDA selects Variational Bayesian Inference with its respective hyperparameters.

3.6 Significance of ASIM-LDA

Section 3.3 and Section 3.4 has laid out explanation on the processes involve in ASIM-LDA which is applied in LDA. ASIM-LDA which comprises of two stages of selection in determining the best inference algorithm instances is described based on the flow chart of processes involved. Before ASIM-LDA starts, raw textual data is pre-processed. Pre-processing enables accurate representation of the textual data prior to the processes in ASIM.

Pre-processed textual data is then used as the input for ASIM-LDA. In ASIM-LDA, the data is inputted in both Gibbs Sampling and Variational Bayesian Inference for training using LDA. Results from the inference algorithm are further refined to get the best possible result among the inference. To achieve best results, ASIM uses two stages of refinement which are the inner optimization in each of the algorithm and selection filter based on the results obtained from individual inference algorithm.

The significant of this method is the two stages of refinement in ASIM. As explained in Section 3.1, through these stages of refinement, ASIM enables increase of performance of LDA in terms of best possible topic model instance.

3.7 Summary

In this chapter, activities involved in this research are detailed out. Research activities done starts with literature review conducted. This initial stage is vital to gain insights and understanding on LDA and the importance of inference algorithms in the modeling. Based on the literature review, research gap was identified. This becomes the foundation for this research. Research objectives and research questions were developed to scope down this research to address the earlier identified research gaps. Subsequently, the methodology, Adaptive Selective Inference Method for LDA

(ASIM-LDA) was proposed. ASIM-LDA was designed and developed through Python programming language. Experiments and evaluation of ASIM-LDA were completed and refinements were made. Final model of ASIM-LDA was built based on these refinements and the research activities are completed after results from the experimentations were completed.

Next, the methodology developed for this research, Adaptive Selection of Inference Method for LDA (ASIM-LDA), is explained. Throughout the chapter, each of the components entailed in ASIM has been discussed. The discussion includes the design of the entire processes, from data pre-processing to results. Data Pre-processing is done through several stages such as stop words removal and stemming. These pre-processed textual data become the input for each of the inference algorithm in ASIM-LDA.

In ASIM-LDA, each of inference algorithm is optimized to the input data received. Through every optimization iteration, the best set of hyperparameters are selected based on the highest topic coherence. The best set of hyperparameters represent the best possible modeling instance for the corresponding inference algorithm. Based on the best results from individual inference algorithm, ASIM performs a selection filter which selects the “best among the best” inference algorithm instance. The final instance of inference algorithm employs the best set of hyperparameters are used.

ASIM employs two steps of refinement, inner optimization in each of the inference algorithm and the selection filter which selects “the best among the best”. The two steps of refinement enable a more informative selection of inference algorithm which leads to better performance of LDA. Evaluations on ASIM for LDA has been conducted and explained in next chapter, Chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

The objective of this chapter is to validate the performance of the proposed improved inference algorithm for LDA. This is to address the third research question posed by this research which is to assess the performance of ASIM-LDA, measured through accuracy, precision and recall analysis, when trained and tested with different complexity of textual dataset. This chapter is organized as follows. Section 4.1 provides the introduction to this chapter. Section 4.2 discusses of the experimentation set-up. In the experimentation set-up, the evaluation metrics used to evaluate the proposed method are explained. Section 4.4 explains the experimentation results of the proposed method. Section 4.5 summarizes the chapter.

4.1 Introduction

In Chapter 1, the research objectives have been explained and deliberated in detail. In this chapter, the final research objective that has been proposed is discussed in more clarity. The objective is to assess the performance of ASIM-LDA, which are measured through accuracy, precision and recall when trained and tested with different complexity of textual dataset. To perform the assessment of the performance of ASIM-LDA, an experiment set-up has been established and is explained in Section 4.2. Subsequently, the results of the performance of ASIM-LDA, which comprised of evaluation of accuracy, precision, and recall, are discussed more in Section 4.3.

4.2 Experiment Set-Up

Figure 4.1 explains the flow of the experimentation set-up for the evaluation of ASIM-LDA.

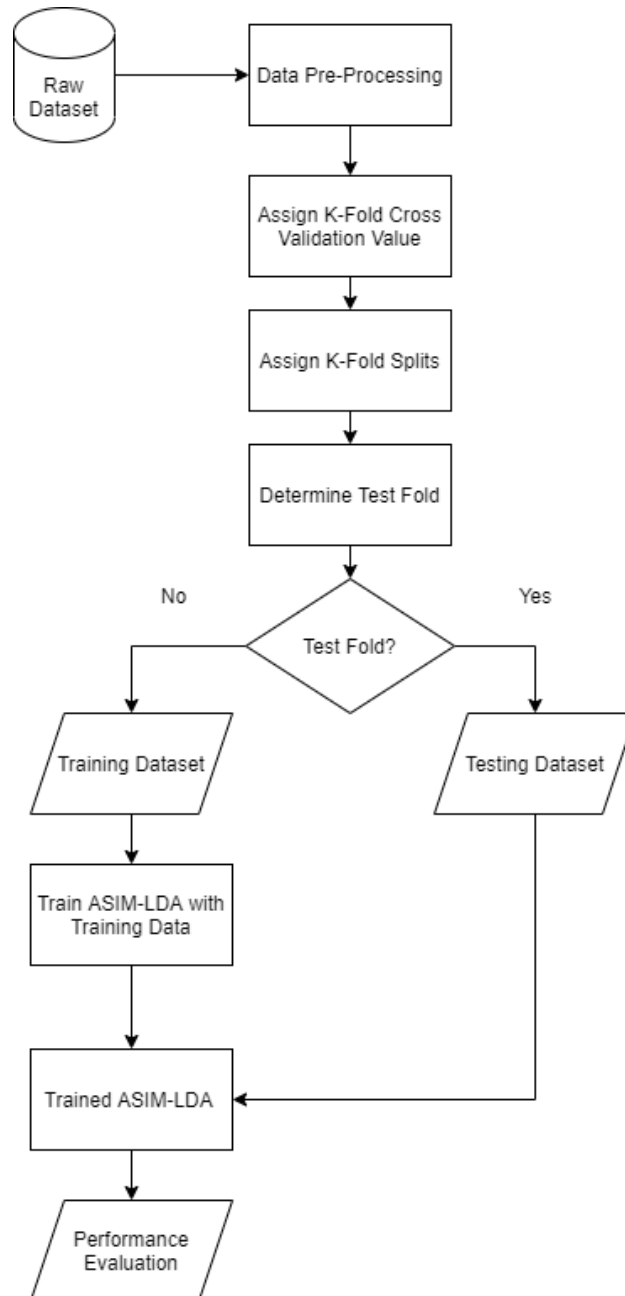


Figure 4.1: Experimentation Set-Up

Based on Figure 4.1, the raw textual dataset is pre-processed, to remove data outliers and anomalies. The pre-processed dataset contains only truly unique tokenized words. Subsequently, the data is split based on the K-Fold Cross Validation method. The value of K used in this research are 3, 5 and 10. These values are based on existing research on evaluating performance of text classification using LDA [65, 83, 84]. More than 1 value is tested to reduce the biasness of the result [85, 86]. Result will be not bias with only one set of dataset's split and ensure that the results obtained from the experiments are robust and correctly represent the real-life results. Only the dataset fold tagged as training dataset is used for training the ASIM-LDA. Once model training is completed, the trained model is used to infer the testing dataset. Based on the inference results on the testing dataset, the performance of ASIM-LDA is evaluated. To evaluate the performance of ASIM-LDA, extrinsic evaluation methods are used.

4.2.1 Evaluation Method – Performance in Classification Task

Extrinsic evaluation is performed to evaluate the performance of the model in creating features for classification task. It is coined as extrinsic as it is a proxy measurement of the performance of the model. This evaluation is done by using the features generated by ASIM-LDA to classify the textual dataset based on their respective class of labels. In this extrinsic evaluation, three different types of metrics are being tested, which are accuracy, precision, and recall.

To evaluate these three metrics, a confusion matrix is constructed for each of the classification task completed. An example of a confusion matrix is as shown in Figure 4.2 :

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Figure 4.2: A Confusion Matrix

Based on Figure 4.2, the confusion matrix consists of four different classes, namely, True Positive, False Positive, True Negative and False Negative. The true cases denote when the algorithm can classify correctly based on its respective classes. The false cases denote when the algorithm is not able to classify the classes correctly. An example of the derivation of these values from an experiment is as tabulated in Table 4.1:

Table 4.1: Example of Confusion Matrix Classes with Results

Actual Result	Predicted Result	Class
Yes	Yes	True Positive
Yes	No	False Negative
No	Yes	False Positive
No	No	True Negative

Based on Table 4.1, the classes in confusion matrix earlier are shown to be resulting from the comparison of actual result and predicted result. These classes are important in the calculation of all three of the evaluation metrics as they denote the actual condition of the model, regardless of the predicted results.

The first metric, accuracy, is defined as how well the classifier can classify correctly based on the true labels. Accuracy is calculated based on this formula:

$$Accuracy = \frac{True\ Positive + True\ Negative}{(True\ Positive + False\ Positive + True\ Negative + False\ Negative)}$$

Based on the formula, accuracy considers how many true results against the entire set of results obtained. Precision is a measure of how well the classifier can classify true positive correctly. Precision is calculated based on this formula:

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)}$$

Based on the formula, precision measures whether the positive predicted by the classifier is positive, between all actual positive results. Precision also measures the consistency the model in predicting or classification task. The third metrics being evaluated is recall which is calculated as follows:

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)}$$

Recall provides a measure of how well the model predicts relevant results compared with the rest of results being predicted by the model, whether it is true positive or false negative. To grasp the performance of the model in classification task, it is important to assess all three metrics. It is because all three metrics provide different type of measurement of performance, providing a holistic view on the performance of ASIM-LDA. In Section 4.2.2, the scope of the experimentation is discussed.

4.2.2 Scope of Experimentation

The scope of the experiments' dataset is only limited to the textual dataset. No figures, tables or images are used in the experiments. This is because only fully structured English sentences are used in the modelling. The algorithms used in the experiments are also focused only on the textual dataset as LDA was developed to solve textual datasets problem. The scope of these textual datasets does not include images or tables as it is not readily structured for application using this algorithm. In Section 4.2.3, the platform used to develop and to evaluate ASIM-LDA are explained.

4.2.3 Platform

The experiments are developed in Python programming language. Each of the experiments conducted is tested with all the datasets explained in Section 3.5.1. The computer specifications that are used in these experiments are shown in Table 4.2.

Table 4.2: Computer Specifications

Items	Specifications
Processor	Intel® Core™ i7-6600U CPU @ 2.60 GHz
RAM	8.00 GB
System Type	64-bit Operating System
Operating System	Windows 10
IDE	Microsoft Visual Studio Code
Python Version	Python 3.6.8, Anaconda Edition

4.3 Results on Accuracy, Precision and Recall on Classification Task

In this section, the results from each of the metrics specified in Section 4.2 are explained. As extrinsic performance evaluation is a proxy measurement of the model performance, a classification algorithm is used for the classification tasks. The algorithm used for all the datasets is fixed, which is Support Vector Classifier.

In this experiment, features created based on ASIM-LDA are compared with features created by other topic modelling methods, LDA with Gibbs Sampling (LDA-GS), LDA with Variational Bayesian Inference (LDA-VB), Latent Semantic Analysis (LSA), and Hierarchical Dirichlet Process (HDP). Each of the evaluation metrics are tested with the three different datasets that have been mentioned in Section 3.5.1.

4.3.1 Accuracy Analysis of ASIM-LDA on Classification Task

The accuracy score for each of the datasets are recorded based on the actual tagging from the extracted dataset, compared with the tagging predicted by the classification algorithm. For each of the method of feature engineering, varying sizes of training datasets are used. The low, medium, and high complexity dataset result for accuracy analysis is as depicted in Figure 4.3.

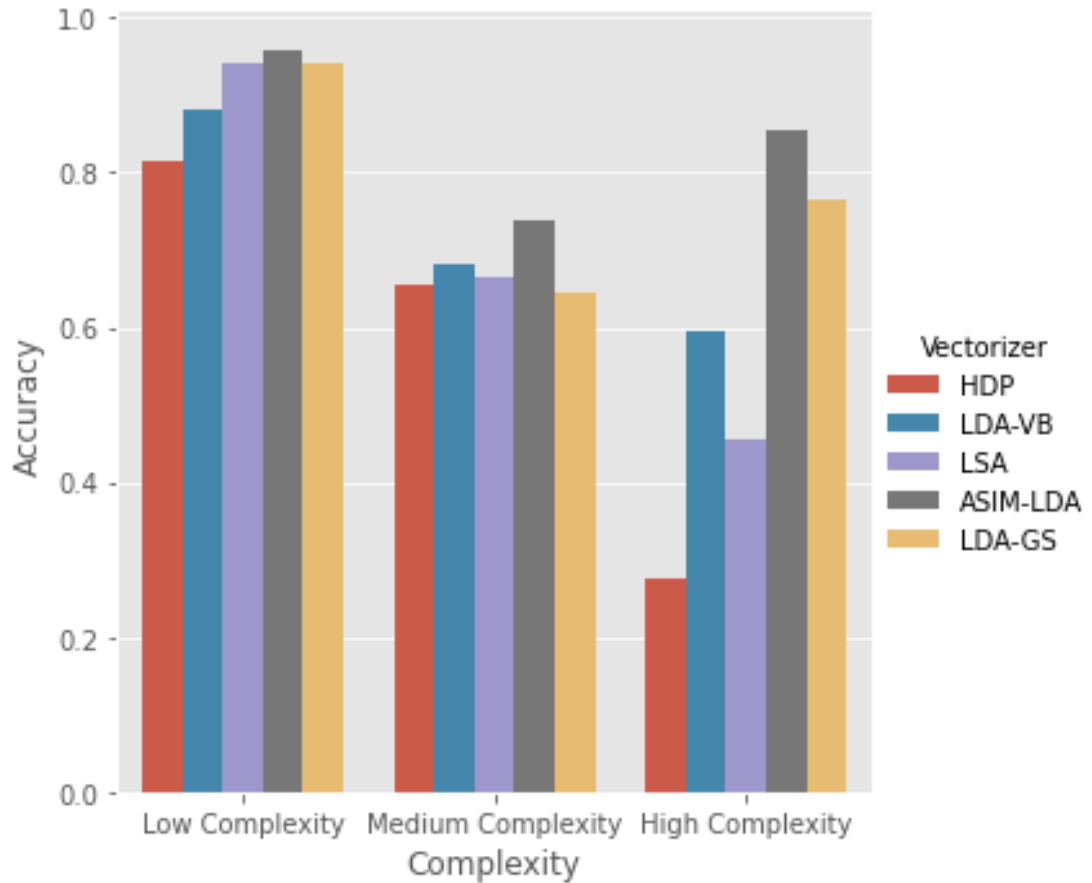


Figure 4.3: Accuracy Analysis for Low, Medium and High Complexity Dataset

For the accuracy analysis for the low complexity dataset, ASIM-LDA shows the highest accuracy score of 0.97. This shows a 3.613% improvement compared to the second-best algorithm, LSA. LSA at score of 0.94 has a comparatively same performance as LDA-GS which scores at 0.94. LDA-VB scored of 0.88, which is 6.27% lower compared to LDA-GS. HDP scored the lowest which is at 0.81.

On the accuracy analysis for medium complexity dataset, ASIM-LDA obtained the highest accuracy score of 0.73. In comparison to its performance in the low complexity dataset, in medium complexity dataset, ASIM-LDA gained 8.22% performance improvement compared to the second-best algorithm, LDA-VB which scored 0.68. The second runner-up, LSA obtained accuracy score of 0.66. The other two algorithms, LDA-GS and HDP attained accuracy score of 0.65 and 0.64, respectively.

Referring to Figure 4.3., three LDA algorithms used with high complexity dataset achieved the best three accuracy score in comparison to other algorithms. ASIM-LDA achieves accuracy score of 0.89, which is the highest. This is a significant difference in terms of accuracy score with LDA-GS which obtained accuracy score of 0.76. LDA-VB scored 0.65, while LSA attained accuracy score of 0.43. Consistently with the low complexity and medium complexity dataset, HDP achieved the lowest accuracy score of 0.29.

Based on the accuracy analyses on all three complexity datasets, ASIM-LDA consistently achieved higher accuracy score to other algorithms tested. On average, ASIM-LDA achieved 9% improvement of accuracy score in comparison to the second-best algorithm. Despite varying complexity of dataset, ASIM-LDA managed to adapt to each of the dataset and retrieve the most accurate results in comparison to its peers. ASIM-LDA has also managed to take advantage of the hyperparameter optimization feature of the base inference algorithms. This is evident when tested with high complexity dataset, where both of LDA-GS and LDA-VB achieved second and third best accuracy score, while ASIM-LDA managed to improve further by optimizing the hyperparameter of both LDA-GS and LDA-VB, selecting the best as the final algorithm. Next, Section 4.3.2 discusses the precision analysis of ASIM-LDA on the same classification task.

4.3.2 Precision Analysis of ASIM-LDA on Classification Task

Precision analysis is conducted to analyse what is the performance of the classifier in retrieving the relevant positive classes, based on all predicted positive classes. This analysis is important in ensuring that the best algorithm selected retrieves the greatest number of correct positive classes, among all the positive classes it predicted. Figure 4.4 shows the precision score achieved by each of the algorithms when tested with the low, medium, and high complexity dataset.

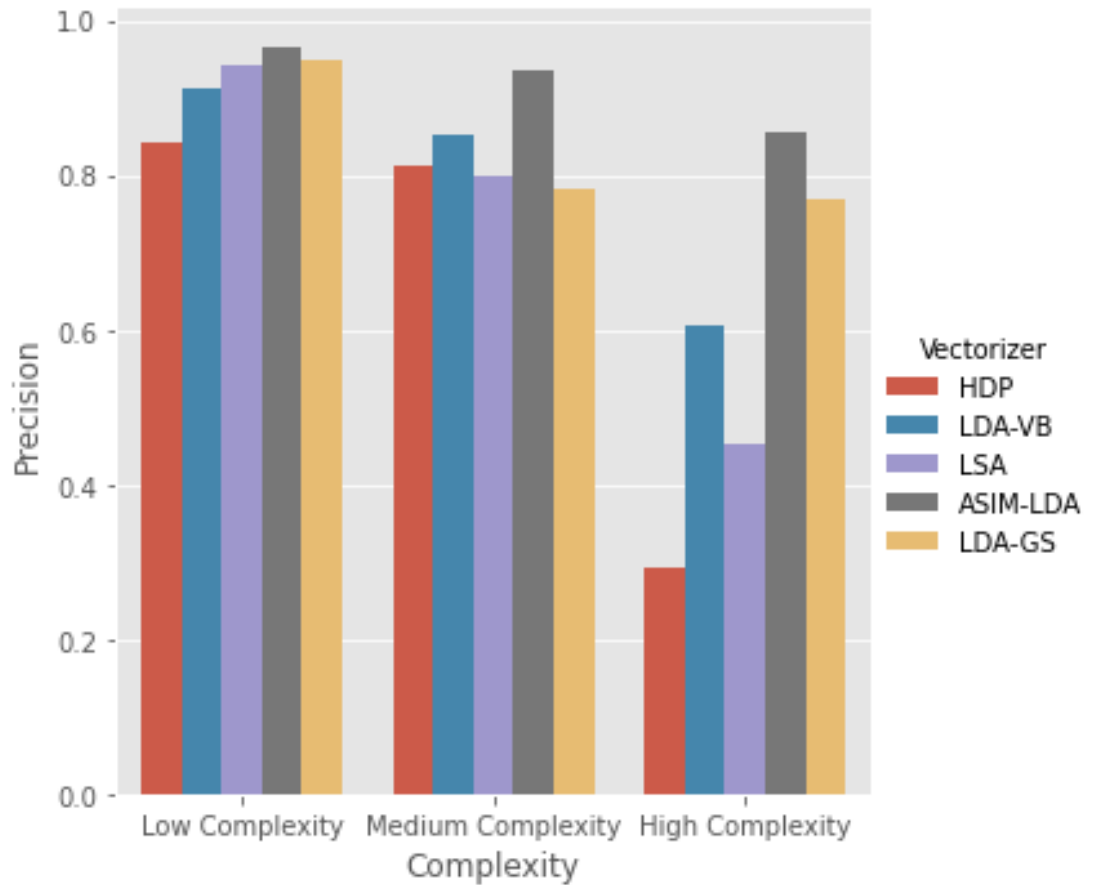


Figure 4.4: Precision Analysis for Low, Medium and High Complexity Dataset

Based on Figure 4.4, the highest precision score for low complexity dataset is achieved by ASIM-LDA, at 0.965. The second highest precision score, 0.948 is scored by LDA-GS. This records an improvement of 1.89%. LSA obtained precision score of 0.941 and LDA-VB attained precision score of 0.911. The lowest precision score for low complexity dataset is achieved by HDP, at 0.843

For precision analysis on medium complexity dataset, ASIM-LDA scored 0.934. This is an improvement of 9.62% in comparison to the second-best algorithm, LDA-VB, which scored 0.852. The third best algorithm when tested with medium complexity dataset is HDP, scored 0.812. The other two algorithms, LSA and LDA-GS, scored lesser than 0.8, at 0.797 and 0.782, respectively.

Based on Figure 4.4, ASIM-LDA maintained as the best algorithm for high complexity dataset, achieving precision score of 0.89. The second-best algorithm tested with high complexity dataset is LDA-GS, which scored 0.78. The percentage difference between the best algorithm and the second-best algorithm stands at 13.9%. LDA-VB obtained a precision score of 0.75, while LSA attained precision score of 0.66. The lowest precision score for high complexity dataset is obtained by HDP at 0.38.

All the precision analysis results show that ASIM-LDA obtained the best result. On average, ASIM-LDA achieved an improvement of 8.47% to the second-best algorithm, regardless of the dataset complexity. This shows that ASIM-LDA can capture more relevant results in comparison to the other algorithms, consistently with increasing of dataset complexity. With the extraction of relevant features, ASIM-LDA can classify more relevant positive classes, while reducing the number of false positive in the classification process. ASIM-LDA also utilized the generative nature of LDA, which can adapt to unseen data better in comparison to LSA. Section 4.3.3 discusses the recall analysis of ASIM-LDA on the classification task.

4.3.3 Recall Analysis of ASIM-LDA on Classification Task

Recall Analysis is conducted to gauge how is the performance of the methods in classifying all the positive classes in the dataset. Recall score is measured by the number of positive classes identified based on all the positive classes present in the dataset, regardless of whether it is classified or not. Figure 4.5 depicts the recall analysis result for different vectorizers on low complexity dataset.

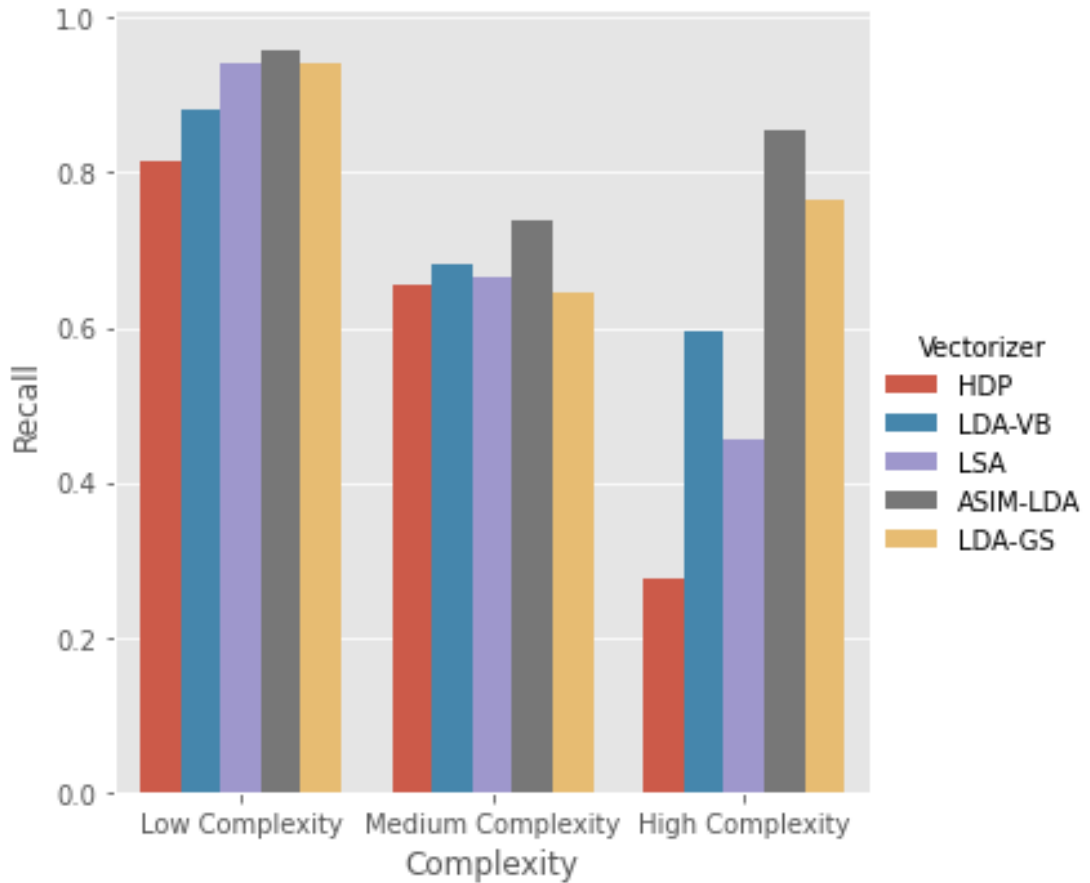


Figure 4.5: Recall Analysis for Low, Medium and High Complexity Dataset

Based on Figure 4.5, ASIM-LDA achieved the highest recall score for low complexity dataset against all the other vectorizers, by scoring 0.97. The second-best algorithm, LSA achieved recall score of 0.92, having a percentage difference of 5.1%. Both LDA algorithms, LDA-GS and LDA-VB scored 0.91 and 0.79, respectively. The worst performing algorithm in the precision analysis is HDP, which scored 0.78.

For recall analysis on the medium complexity dataset is achieved by ASIM-LDA, scoring 0.737. This shows an improvement of 8.06%, when compared to the next best algorithm, LDA-VB which achieved recall score of 0.682. The third best algorithm, LSA attained recall score of 0.664. The last two algorithms, HDP and LDA-GS both achieved scores lesser than 0.66, at 0.656 and 0.645 correspondingly.

ASIM-LDA is also the best algorithm in the recall analysis for high complexity dataset, achieving recall score of 0.889. This is a significant improvement in comparison to LDA-GS, the second-best algorithm, which achieved recall score of 0.767. Next, LDA-VB attained recall score of 0.657. The rest, LSA and HDP managed to only attain recall score of below 0.5. LSA scored 0.440 and HDP managed to score 0.290.

Based on all the recall analysis results, it is evident that ASIM-LDA is consistently the best algorithm out of all the algorithms tested. ASIM-LDA shows an average improvement percentage of 9.67% against the second-best algorithm in each of the dataset complexity. This shows that despite varying dataset complexity, ASIM-LDA has been able to predict the greatest number of relevant positive classes, in comparison to the other algorithms. The improvement is mainly possible due to adaptiveness feature of ASIM-LDA, which utilizes both Gibbs Sampling and Variational Bayesian Inference. Based on results when tested with medium complexity and high complexity dataset, it is evident that LDA-VB is more suited with the former, while LDA-GS is better with the latter. ASIM-LDA takes advantage of this feature and can perform better than each of the component inference algorithm.

Based on the analysis done, ASIM-LDA has consistently established its ability to adapt to varying dataset. This is demonstrated by the algorithm achieved the best score in accuracy analysis, precision analysis and recall analysis. The sets of experiments are done to answer the third research question investigated in this study, which is to assess the performance of ASIM-LDA, which are measured through accuracy, precision and recall when trained and tested with different complexity of textual dataset. Section 4.4 summarizes Chapter 4.

4.4 Summary

In this chapter, the validation of the performance of the proposed ASIM-LDA has been presented and discussed. The experiment set-up is discussed, which comprises of the evaluation method used, scope of experimentation and the platform used for the experiments. Evaluation method used in validating the performance of ASIM-LDA is

based on extrinsic evaluation method, through classification task. The classification task is done against different complexity of dataset to gauge the consistency of performance. Measurements used in the experiment are accuracy, precision, and recall. All three analyses are important as each of the analysis measure different aspect of the performance. The scope of experimentation is limited to the textual dataset which does not include any numerical information.

Three different datasets are used in the experiment, which are represented as low, medium, and high complexity datasets. As complexity is gauged by the number of unique words occurring in the dataset, each of the dataset has varying complexity. This is to test the algorithm ability to adapt to each of the dataset, irrespective to the complexity of the dataset. ASIM-LDA is tested against with four other algorithms, LDA-GS, LDA-VB, LSA and HDP in all the evaluation analysis. Based on the results discussed, ASIM-LDA is shown to be the best algorithm and managed to adapt to varying complexity of dataset. Next, Chapter 5 presents the conclusion and recommendation for the extension of this research.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

Section 5.1 condenses the research study conducted through discussing briefly on the outline of the research. These includes brief explanations of literature review done, model design and development, model evaluation and the analysis of the results. Section 5.2 discusses on the contribution of this research. Section 5.3 explains on the limitations of this research. Section 5.4 concludes the thesis with suggestions on future directions.

5.1 Research Summary

This section briefly describes on the research work done by explaining the stages taken to complete this research. For each of the stages, summary of the work completed, and the findings acquired are explained.

Literature Review: Domain of this research is in the textual dataset. This research focuses on the improvement of the inference algorithm utilised by LDA, enable it to adapt to varying complexity of textual dataset. Chapter 2 is organised in funnel fashion, discussing on comparison of different text representation algorithms. Based on this discussion, topic modelling is further discussed, through comparison of different topic modelling algorithm. Next, LDA is discussed in clarity. A thorough study of both the theoretical and application of LDA is done. The research gap is found whereby there is two main inference algorithms mainly utilised by LDA, which performed differently depending on the complexity of the dataset. Hyperparameters of the inference algorithm used are also discussed as it has an effect to the fitness of LDA to selected dataset. Based on this finding, this forms the ground for the formulation of research questions and objectives.

Research Methodology: Based on the finding from the literature review performed, the research methodology has been designed and developed. The research methodology consists of the action plan established to address the research questions. Chapter 3 is

organised into two main sections, discussing the overall research framework and the design of ASIM-LDA. The research framework describes the research activities conducted, which includes the key milestones of the research. Design of ASIM-LDA discusses the algorithm developed, the implementation of ASIM-LDA and the significance of ASIM-LDA.

Results and Discussion: Evaluation of the proposed method, ASIM-LDA is described in Chapter 4. The performance of ASIM-LDA is measured through accuracy analysis, precision analysis and recall analysis. In each of the analysis, ASIM-LDA is compared with other topic modelling algorithms. To gauge ASIM-LDA ability to adapt to different complexity of textual dataset, three different level of complexity is used. In concluding the chapter, the results obtained from all the analyses are summarised and discussed.

5.2 Research Contribution

ASIM-LDA improves on the adaptability aspect of LDA in varying complexity of textual dataset. As the complexity of textual dataset is measured through the number of unique words, this poses a challenge to existing implementation of LDA's inference algorithm. This is due to different type of inference algorithm works best with different complexity of textual dataset. Another gap found on the existing implementation of LDA's inference algorithm is in finding the best set of hyperparameters. These sets of hyperparameters are crucial in improving the fitness of the inference algorithm to a given textual dataset.

ASIM-LDA addresses both gaps found by introducing two stages of refinement of inference algorithm. In the first stage, ASIM-LDA performs hyperparameter optimization for both Gibbs Sampling and Variational Bayesian Inference. The hyperparameter optimization goal is to find a set of hyperparameters which enable the best fitness of model to the dataset. The fitness of model, or the objective function, employed in this optimization is finding the highest topic coherence score. Topic coherence score is selected as it has a strong correlation with the inference algorithm ability to adapt to a dataset.

The first stage of optimization technique used is a combination of random search and grid search. The random search initializes the list of the hyperparameters. ASIM-LDA uses the list of hyperparameters to initialise each of the two inference algorithms. Once the inference algorithm is initialised with the hyperparameters, it is trained and tested on a given textual dataset. This process is done iteratively for each of the inference algorithm. After ASIM-LDA exhausts the list of hyperparameters, the best set of hyperparameters are selected for each of the inference algorithm based on their topic coherence score. Once this completes, grid search process starts. Grid search initialises another list of hyperparameters, based on the best hyperparameters selected from random search. This list is generated based on gaussian distribution, localised by the selected hyperparameter. Then, ASIM-LDA iteratively train and test each of the inference algorithm until all sets of hyperparameters are used. Again, the best set of hyperparameters is selected based on the best topic coherence score achieved by the inference algorithm. This completes the first stage of refinement.

The second stage of refinement consists of selection of inference algorithm based on their respective topic coherence score. This is an important feature of ASIM-LDA as it takes into consideration both Gibbs Sampling and Variational Bayesian Inference and select the best of the two. This eliminates the guesswork of which inference works best for different dataset. Once selected, ASIM-LDA uses the best inference algorithm with the best hyperparameter.

To assess the performance of ASIM-LDA in real life application, an extrinsic evaluation is conducted. This extrinsic evaluation is done to evaluate the performance of ASIM-LDA in generating robust text representation for text classification task. The evaluation is done on varying complexity of textual dataset. Based on the result obtained, two conclusions can be made. First, ASIM-LDA evidently improved on both Gibbs Sampling and Variational Bayesian Inference, by employing the hyperparameter optimization. Second, ASIM-LDA managed to adapt to varying dataset complexity, which is enabled by the second stage of refinement, which only select the best inference algorithm based on the given dataset. The results prove the ability of ASIM-LDA to achieve superior performance despite tested in varying complexity of dataset.

Table 5.1 depicts the research contribution is mapped to the research objectives that have been stated earlier in Chapter 1.

Table 5.1: Research Objective Mapping to Contribution

Research Objective	Research Contributions
To propose an adaptive selection of inference method for Latent Dirichlet Allocation (ASIM-LDA) based on maximizing topic coherence score with given textual dataset	This research proposed the two stages of optimization, random search, and grid search to maximize the topic coherence, in varying complexity textual dataset. The proposed framework is discussed in Section 3.2.
To improve Latent Dirichlet Allocation's inference method by combining random search and grid search in hyperparameter optimization to obtain the highest topic coherence score in textual data environment.	In this research, the proposed framework which combines random search and grid search has been implemented. In this framework, both inference method, Variational Bayesian Inference and Gibbs Sampling, is considered to obtain the highest topic coherence score. The implementation of ASIM-LDA is discussed in Section 3.3 and 3.4.
To assess the performance parameters of ASIM-LDA, measured through accuracy, precision and recall analysis, through experimentation with different complexity of textual dataset.	In this research, performance analysis has been conducted on three complexity dataset, namely low, medium, and high complexity. Consistently, ASIM-LDA has obtained superior results in comparison to other implementations of topic modelling algorithms. The results are discussed in Section 4.3.

5.3 Research Limitation

Based on the discussion in Section 5.2, the research contribution has been summarised in terms of the improvement of the proposed method in adapting to varying complexity of dataset. However, there are several areas that of limitations on the proposed method. These are listed as follows:

- i. The first stage of refinement, the hyperparameter optimization does not guarantee in avoiding the local minima problem. This research focuses on the two steps of hyperparameter optimization which might reduce the probability of meeting local minima, but it does not eliminate the possibility entirely.
- ii. This research did not explore on the effect of two stages of refinement to the time and space complexity of the entire algorithm. As the research only focused on achieving superior algorithm performance, there is a trade-off between algorithm performance and algorithm efficiency.
- iii. As this scope of this research is limited to textual dataset, ASIM-LDA ability to adapt to different type of data (such as numerical data and geospatial data) has not been tested. The usage of ASIM-LDA in these domains might require adopting different approach, especially in defining the fitness of the model to the dataset.

5.4 Future Directions

Referring to Section 5.3, there are certain areas of improvement to extend on this research work. Several suggestions of the future directions are as follows:

- i. The proposed method can be extended in exploring method that can further minimize the effect of local minima. To address this problem, different types of hyperparameter optimization algorithm can be tested and employed in the first stage of refinement. These include the usage of gradient boosting or Bayesian hyperparameter optimization.

- ii. To reduce the time/space complexity of ASIM-LDA, dynamic programming can be employed in both stages of refinement. This includes memoization, where the program can store results of computationally expensive function calls. Cloud computing can also be utilised to cater to usage in big data, which can also improve the efficiency of ASIM-LDA. By utilising Spark or any of the big data framework, ASIM-LDA can be extended to work parallelly and use with big data.

- iii. ASIM-LDA can be extended to cater to different types of data by incorporating different measure of objective function that it maximizes. This is because the objective function defined in ASIM-LDA, the topic coherence score, might not be applicable to different type of dataset. A different objective function such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) can be tested with these data.

References

- [1] P. Martí, L. Serrano-Estrada, and A. Nolasco-Cirugeda, "Social media data: Challenges, opportunities and limitations in urban studies," *Computers, Environment and Urban Systems*, vol. 74, pp. 161-174, 2019.
- [2] A. Singh, N. Shukla, and N. Mishra, "Social media data analytics to improve supply chain management in food industries," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 398-415, 2018.
- [3] S. Stieglitz and L. Dang-Xuan, "Social media and political communication: a social media analytics framework," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1277-1291, 2013.
- [4] H. Tenkanen *et al.*, "Instagram, Flickr, or Twitter: Assessing the usability of social media data for visitor monitoring in protected areas," *Scientific reports*, vol. 7, no. 1, pp. 1-11, 2017.
- [5] F. De Prieëlle, M. De Reuver, and J. Rezaei, "The role of ecosystem data governance in adoption of data platforms by Internet-of-Things data providers: Case of Dutch horticulture industry," *IEEE Transactions on Engineering Management*, 2020.
- [6] T. Elsaleh, S. Enshaeifar, R. Rezvani, S. T. Acton, V. Janeiko, and M. Bermudez-Edo, "IoT-Stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services," *Sensors*, vol. 20, no. 4, p. 953, 2020.
- [7] M. S. Mahdavinejad, M. Rezvan, M. Barekatin, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161-175, 2018.
- [8] B. Balducci and D. Marinova, "Unstructured data in marketing," *Journal of the Academy of Marketing Science*, vol. 46, no. 4, pp. 557-590, 2018.
- [9] C. Crouspeyre, E. Alesi, and K. Lespinasse, "From Creditworthiness to Trustworthiness with alternative NLP/NLU approaches," in *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, 2019, pp. 96-98.
- [10] A. Fantechi, A. Ferrari, S. Gnesi, and L. Semini, "Requirement engineering of software product lines: extracting variability using NLP," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 2018: IEEE, pp. 418-423.
- [11] A. Ferrari and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering," *Automated Software Engineering*, vol. 26, no. 3, pp. 559-598, 2019.
- [12] F. Nazir, W. H. Butt, M. W. Anwar, and M. A. K. Khattak, "The applications of natural language processing (NLP) for software requirement engineering-a systematic literature review," in *International conference on information science and applications*, 2017: Springer, pp. 485-493.
- [13] Y. Li, J. Zhang, and B. Yu, "An NLP analysis of exaggerated claims in science news," in *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, 2017, pp. 106-111.
- [14] S. Yıldırım, D. Jothimani, C. Kavaklıoğlu, and A. Başar, "Classification of " Hot News" for Financial Forecast Using NLP Techniques," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018: IEEE, pp. 4719-4722.
- [15] M. Chary, S. Parikh, A. F. Manini, E. W. Boyer, and M. Radeos, "A review of natural language processing in medical education," *Western Journal of Emergency Medicine*, vol. 20, no. 1, p. 78, 2019.
- [16] X. Chen, H. Xie, F. L. Wang, Z. Liu, J. Xu, and T. Hao, "A bibliometric analysis of natural language processing in medical research," *BMC medical informatics and decision making*, vol. 18, no. 1, pp. 1-14, 2018.
- [17] L. B. Fazlic, A. Hallawa, A. Schmeink, A. Peine, L. Martin, and G. Dartmann, "A novel NLP-FUZZY system prototype for information extraction from medical guidelines," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019: IEEE, pp. 1025-1030.
- [18] S. Das, R. K. Behera, and S. K. Rath, "Real-time sentiment analysis of twitter streaming data for stock prediction," *Procedia computer science*, vol. 132, pp. 956-964, 2018.

- [19] T. Dogan and A. K. Uysal, "On term frequency factor in supervised term weighting schemes for text classification," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9545-9560, 2019.
- [20] V. H. A. Soares, R. J. Campello, S. Nourashrafeddin, E. Milios, and M. C. Naldi, "Combining semantic and term frequency similarities for text clustering," *Knowledge and Information Systems*, vol. 61, no. 3, pp. 1485-1516, 2019.
- [21] A. Goodkind and K. Bicknell, "Predictive power of word surprisal for reading times is a linear function of language model quality," in *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, 2018, pp. 10-18.
- [22] M. Del Tredici, D. Marcheggiani, S. S. i. Walde, and R. Fernández, "You shall know a user by the company it keeps: Dynamic representations for social media users in nlp," *arXiv preprint arXiv:1909.00412*, 2019.
- [23] Q. Liang and K. Wang, "Monitoring of user-generated reviews via a sequential reverse joint sentiment-topic model," *Quality and Reliability Engineering International*, vol. 35, no. 4, pp. 1180-1199, 2019.
- [24] E. O. Park, B. K. Chae, J. Kwon, and W.-H. Kim, "The effects of green restaurant attributes on customer satisfaction using the structural topic model on online customer reviews," *Sustainability*, vol. 12, no. 7, p. 2843, 2020.
- [25] V. Rakesh, W. Ding, A. Ahuja, N. Rao, Y. Sun, and C. K. Reddy, "A sparse topic model for extracting aspect-specific summaries from online reviews," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1573-1582.
- [26] S. Xiong, K. Wang, D. Ji, and B. Wang, "A short text sentiment-topic model for product reviews," *Neurocomputing*, vol. 297, pp. 94-102, 2018.
- [27] J. L. Boyd-Graber, Y. Hu, and D. Mimno, *Applications of topic models*. now Publishers Incorporated, 2017.
- [28] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology*, vol. 38, no. 1, pp. 188-230, 2004.
- [29] T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch, *Handbook of latent semantic analysis*. Psychology Press, 2013.
- [30] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993-1022, 2003.
- [31] U. Chauhan and A. Shah, "Topic modeling using latent Dirichlet allocation: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1-35, 2021.
- [32] T. Hofmann, "Probabilistic latent semantic indexing," in *ACM SIGIR Forum*, 2017, vol. 51, no. 2: ACM, pp. 211-218.
- [33] H. Jelodar *et al.*, "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15169-15211, 2019.
- [34] A. K. Dhaka, A. Catalina, M. Welandawe, M. R. Andersen, J. Huggins, and A. Vehtari, "Challenges and Opportunities in High Dimensional Variational Inference," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7787-7798, 2021.
- [35] D. Sontag and D. Roy, "Complexity of inference in latent dirichlet allocation," in *Advances in neural information processing systems*, 2011, pp. 1008-1016.
- [36] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008-2026, 2018.
- [37] T. Salimans, D. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1218-1226.
- [38] R. E. Turner and M. Sahani, "Two problems with variational expectation maximisation for time-series models," *Bayesian Time series models*, pp. 115-138, 2011.
- [39] A. C. Miller, N. Foti, and R. P. Adams, "Variational Boosting: Iteratively Refining Posterior Approximations," *arXiv preprint arXiv:1611.06585*, 2016.
- [40] W. M. Darling, "A theoretical and practical implementation tutorial on topic modeling and gibbs sampling," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 642-647.
- [41] R. Balasubramanyan and W. W. Cohen, "Block-LDA: Jointly modeling entity-annotated text and entity-entity links," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2011: SIAM, pp. 450-461.

- [42] D. Cheng and Y. Liu, "Parallel gibbs sampling for hierarchical dirichlet processes via gamma processes equivalence," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 562-571.
- [43] B. P. Eddy, N. A. Kraft, and J. Gray, "Impact of structural weighting on a latent Dirichlet allocation-based feature location technique," *Journal of Software: Evolution and Process*, vol. 30, no. 1, 2018.
- [44] J. Al Qundus, A. Paschke, S. Gupta, A. M. Alzouby, and M. Yousef, "Exploring the impact of short-text complexity and structure on its quality in social media," *Journal of Enterprise Information Management*, 2020.
- [45] M. Ereemeev and K. Vorontsov, "Lexical quantile-based text complexity measure," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 270-275.
- [46] R. B. V. Padmanabhan. "Big Data analytics in oil and gas." <http://www.bain.com/Images/BAIN BRIEF Big Data analytics in oil and gas.pdf> (accessed).
- [47] W. H. Inmon. "Untangling the Definition of Unstructured Data." <http://www.ibmdatahub.com/blog/untangling-definition-unstructured-data> (accessed 28 November, 2016).
- [48] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137-144, 2015.
- [49] S. Ali, "Key Considerations For Managing Unstructured Data," *Pipeline & Gas Journal*, vol. 240, no. 7, 2013. [Online]. Available: <https://pgjonline.com/2013/07/18/key-considerations-for-managing-unstructured-data/>.
- [50] K. Boman, "The Big Challenges of Big Data for Oil, Gas," 2013. [Online]. Available: http://www.rigzone.com/news/oil_gas/a/130590/The_Big_Challenges_of_Big_Data_for_Oil_Gas/?pgNum=0.
- [51] A. Holzinger, J. Schantl, M. Schroettner, C. Seifert, and K. Verspoor, "Biomedical text mining: state-of-the-art, open problems and future challenges," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*: Springer, 2014, pp. 271-300.
- [52] S. Tirunillai and G. J. Tellis, "Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation," *Journal of Marketing Research*, vol. 51, no. 4, pp. 463-479, 2014.
- [53] C. Geigle, "Inference Methods for Latent Dirichlet Allocation," 2016. [Online]. Available: <http://times.cs.uiuc.edu/course/598f16/notes/lda-survey.pdf>.
- [54] J. Qiang, Z. Qian, Y. Li, Y. Yuan, and X. Wu, "Short text topic modeling techniques, applications, and performance: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1427-1445, 2020.
- [55] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439-453, 2020.
- [56] M. Das, S. Kamalanathan, and P. Alphonse, "A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset," in *COLINS*, 2021, pp. 98-107.
- [57] A. Adebisi, O. M. Ogunleye, M. Adebisi, and J. Okesola, "A comparative analysis of tf-idf, lsi and lda in semantic information retrieval approach for paper-reviewer assignment," *Journal of Engineering and Applied Sciences*, vol. 14, no. 10, pp. 3378-3382, 2019.
- [58] G. A. Dalaorao, A. M. Sison, and R. P. Medina, "Integrating collocation as tf-idf enhancement to improve classification accuracy," in *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 2019: IEEE, pp. 282-285.
- [59] S. Qaiser and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25-29, 2018.
- [60] P. Ciaccia, "Latent Semantic Indexing," 2012.
- [61] P. Kulkarni, S. Joshi, and M. S. Brown, *Big data analytics*. PHI Learning Pvt. Ltd., 2016.
- [62] C. W. Schmidt, "Improving a tf-idf weighted document vector embedding," *arXiv preprint arXiv:1902.09875*, 2019.
- [63] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77-84, 2012.
- [64] L. Wu, S. C. Hoi, and N. Yu, "Semantics-preserving bag-of-words models and applications," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1908-1920, 2010.

- [65] E. Wahyudi and R. Kusumaningrum, "Aspect based sentiment analysis in E-commerce user reviews using Latent Dirichlet Allocation (LDA) and Sentiment Lexicon," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, 2019: IEEE, pp. 1-6.
- [66] A. Knispelis, "LDA Topic Models," ed, 2016.
- [67] Y. Guo, S. J. Barnes, and Q. Jia, "Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation," *Tourism Management*, vol. 59, pp. 467-483, 2017/04/01/ 2017, doi: <https://doi.org/10.1016/j.tourman.2016.09.009>.
- [68] S. Moro, P. Cortez, and P. Rita, "Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1314-1324, 2015.
- [69] T. Dyer, M. Lang, and L. Stice-Lawrence, "The evolution of 10-K textual disclosure: Evidence from Latent Dirichlet Allocation," *Journal of Accounting and Economics*, vol. 64, no. 2-3, pp. 221-245, 2017.
- [70] L. R. Biggers, C. Bocovich, R. Capshaw, B. P. Eddy, L. H. Etzkorn, and N. A. Kraft, "Configuring latent dirichlet allocation based feature location," *Empirical Software Engineering*, vol. 19, no. 3, pp. 465-500, 2014.
- [71] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *Advances in neural information processing systems*, 2002, pp. 601-608.
- [72] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131-146, 2008.
- [73] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press Boca Raton, FL, 2014.
- [74] W. M. Bolstad and J. M. Curran, *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.
- [75] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859-877, 2017.
- [76] G. L. Jones and Q. Qin, "Markov chain Monte Carlo in practice," *Annual Review of Statistics and Its Application*, vol. 9, pp. 557-578, 2022.
- [77] J. S. Speagle, "A conceptual introduction to markov chain monte carlo methods," *arXiv preprint arXiv:1909.12313*, 2019.
- [78] J. Hockenmaier, "Lecture 7: Variational Inference for LDA," UIUC, 2010.
- [79] D. Shao, C. Li, C. Huang, Y. Xiang, and Z. Yu, "A news classification applied with new text representation based on the improved LDA," *Multimedia Tools and Applications*, pp. 1-25, 2022.
- [80] I. Yildirim, "Bayesian inference: Gibbs sampling," *Technical Note, University of Rochester*, 2012.
- [81] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.
- [82] D. a. G. Dua, Casey, "UCI Machine Learning Repository," ed, 2019.
- [83] A. H. Razavi and D. Inkpen, "Text representation using multi-level latent Dirichlet allocation," in *Canadian Conference on Artificial Intelligence*, 2014: Springer, pp. 215-226.
- [84] D. Inkpen and A. H. Razavi, "Topic Classification using Latent Dirichlet Allocation at Multiple Levels," *Int. J. Comput. Linguistics Appl.*, vol. 5, no. 1, pp. 43-55, 2014.
- [85] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The 'K' in K-fold cross validation," in *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2012: i6doc. com publ, pp. 441-446.
- [86] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586-1594, 2019.

List of Publications

- W. M. A. M. Zubir, I. A. Aziz and J. Jaafar, "A survey on textual semantic classification algorithms," 2017 IEEE Conference on Big Data and Analytics (ICBDA), Kuching, 2017, pp. 1-6, doi: 10.1109/ICBDAA.2017.8284098.
- Zubir, W. M. A. M., Aziz, I. A., Jaafar, J., & Hasan, M. H. (2017, September). Inference algorithms in Latent Dirichlet Allocation for semantic classification. In Proceedings of the Computational Methods in Systems and Software (pp. 173-184). Springer, Cham.
- Zubir, W. M. A. M., Aziz, I. A., & Jaafar, J. (2018, September). Evaluation of Machine Learning Algorithms in Predicting CO_2 Internal Corrosion in Oil and Gas Pipelines. In Proceedings of the Computational Methods in Systems and Software (pp. 236-254). Springer, Cham.
- Zubir, W. M. A. M., Aziz, I. A., Haron, N. S., Jaafar, J., & Mehat, M. (2016, August). CO₂ corrosion rate determination mechanism implementing de Waard-Milliams model for oil & gas pipeline. In 2016 3rd International Conference on Computer and Information Sciences (ICCOINS) (pp. 298-303). IEEE.
- Hassanudin, S. N., Aziz, I. A., Jaafar, J., Qaiyum, S., & Zubir, W. M. A. M. (2017, November). Predictive analytic dashboard for desalter and crude distillation unit. In 2017 IEEE Conference on Big Data and Analytics (ICBDA) (pp. 55-60). IEEE.