Thesis no: MSSE-2016-32

A Quality Criteria Based Evaluation of Topic Models

Veer Reddy Sathi and Jai Simha Ramanujapura

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Veer Reddy Sathi
vesa15@student.bth.se

Jai Simha Ramanujapura
jara15@student.bth.se

University advisor:
Michael Unterkalmsteiner
Department of Software Engineering

Faculty of Computing	**Internet**	**: www.bth.se**
Blekinge Institute of Technology	**Phone**	**: +46 455 38 50 00**
SE-371 79 Karlskrona, Sweden	**Fax**	**: +46 455 38 50 57**

ABSTRACT

Context. Software testing is the process, where a particular software product, or a system is executed, in order to find out the bugs, or issues which may otherwise degrade its performance. Software testing is usually done based on pre-defined test cases. A test case can be defined as a set of terms, or conditions that are used by the software testers to determine, if a particular system that is under test operates as it is supposed to or not. However, in numerous situations, test cases can be so many that executing each and every test case is practically impossible, as there may be many constraints. This causes the testers to prioritize the functions that are to be tested. This is where the ability of topic models can be exploited. Topic models are unsupervised machine learning algorithms that can explore large corpora of data, and classify them by identifying the hidden thematic structure in those corpora. Using topic models for test case prioritization can save a lot of time and resources.

Objectives. In our study, we provide an overview of the amount of research that has been done in relation to topic models. We want to uncover various quality criteria, evaluation methods, and metrics that can be used to evaluate the topic models. Furthermore, we would also like to compare the performance of two topic models that are optimized for different quality criteria, on a particular interpretability task, and thereby determine the topic model that produces the best results for that task.

Methods. A systematic mapping study was performed to gain an overview of the previous research that has been done on the evaluation of topic models. The mapping study focused on identifying quality criteria, evaluation methods, and metrics that have been used to evaluate topic models. The results of mapping study were then used to identify the most used quality criteria. The evaluation methods related to those criteria were then used to generate two optimized topic models. An experiment was conducted, where the topics generated from those two topic models were provided to a group of 20 subjects. The task was designed, so as to evaluate the interpretability of the generated topics. The performance of the two topic models was then compared by using the Precision, Recall, and F-measure.

Results. Based on the results obtained from the mapping study, Latent Dirichlet Allocation (LDA) was found to be the most widely used topic model. Two LDA topic models were created, optimizing one for the quality criterion Generalizability (T_G), and one for Interpretability (T_I); using the Perplexity, and Point-wise Mutual Information (PMI) measures respectively. For the selected metrics, T_I showed better performance, in Precision and F-measure, than T_G. However, the performance of both T_I and T_G was comparable in case of Recall. The total run time of T_I was also found to be significantly high than T_G. The run time of T_I was 46 hours, and 35 minutes, whereas for T_G it was 3 hours, and 30 minutes.

Conclusions. Looking at the F-measure, it can be concluded that the interpretability topic model (T_I) performs better than the generalizability topic model (T_G). However, while T_I performed better in precision, recall was comparable. Furthermore, the computational cost to create T_I is significantly higher than for T_G. Hence, we conclude that, the selection of the topic model optimization should be based on the aim of the task the model is used for. If the task requires high interpretability of the model, and precision is important, such as for the prioritization of test cases based on content, then T_I would be the right choice, provided time is not a limiting factor. However, if the task aims at generating topics that provide a basic understanding of the concepts (i.e., interpretability is not a high priority), then T_G is the most suitable choice; thus making it more suitable for time critical tasks.

Keywords: Topic models, Topic interpretability, Test cases, Latent Dirichlet Allocation, Topic model optimization.

ACKNOWLEDGEMENTS

We would like to thank our supervisor Dr. Michael Unterkalmsteiner for his invaluable insights, and his support in carrying out this research; and for believing in us, and motivating us whenever required. we would also like to thank him for the quick support, and suggestions without which, we wouldn't have been able to complete the thesis.

We would also like to thank our parents for their tremendous support, in fulfilling our dreams. A special thanks to all our friends who have been with us throughout this journey. Especially Susheel Sagar, and Venkatesh Boddapati, for sharing your incredible tech expertise, and knowledge with us.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Over the past decade, the world has seen a lot of technological advancement in various fields. The knowledge and collective information of these technological advancements also keeps piling up day-by-day. Be it in the form of books, databases, research articles, scientific journals, audio and visual data, online blogs and websites, or endless lines of source code; knowledge is always expanding [1]. Looking for the desired information in such large piles of data seemingly becomes a complex, and a time consuming task. This led to a lot of research being directed towards finding an effective means to search, classify and retrieve these large chunks of data. It was found that machine learning approaches can be efficient in handling such tasks as they save human resources and time [2].

The key idea behind machine learning in general, is, to construct the appropriate models that can accomplish the desired tasks, by utilizing the apt and relevant characteristics, or resources [3]. The approaches of machine learning can be classified into supervised and unsupervised learning methods [4]. In supervised learning approaches, the labels are pre-defined [3], and the data is categorized accordingly based on its likelihood [5]. These labels are usually class names, or the function values that describe a group of things with similar attributes or properties [3]. These labels are defined by considering a training set of data and identifying the likelihood of those labels in the selected set of data [5]. Whereas, in unsupervised learning approaches neither prior information is provided [5], nor the labels are defined [3]; and the entire procedure takes place in an autonomous fashion. However, for supervised learning approaches to perform efficiently, they require the training data consisting of, labelled units [6]. Manually labelling each and every unit in the data set will require a lot of capital, human labor, and time as resources [3]. Moreover, it has also been reported that the supervised learning approaches tend to generate results of poor quality [7], [8]. The supervised techniques exhibit poor performance, when it comes to retrieving semantically meaningful information from general data [7]. In [8], Schnober et al. say that supervised techniques show feeble results even for simple tasks such as detecting sentence boundaries, lemmatization, tagging etc. This is where the unsupervised learning approaches tend to be more effective, as they can be applied over an unorganized, unstructured, or even unlabeled datasets; and thereby attracted a lot of attention in the area of text mining [8].

Probabilistic topic models are such unsupervised machine learning algorithms that were designed and developed to explore large corpora of data uncovering their hidden thematic structure. Apart from revealing the hidden thematic structure in the datasets, the topic models also allow us to know the relation between the themes, and also how these themes evolved with time [1]. The utility of topic models can also be harnessed within the field of software engineering. In order to save time and effort in software engineering related tasks, incorporating machine learning algorithms has proven to be highly beneficial [9]. Recent studies have shown the usage of topic models over different software engineering tasks like traceability link criteria, feature location, and software artifact labeling [10]. As the research advances, researchers have also put forward the idea of implementing topic models in the field of software testing, specifically in test case prioritization [11]. In the context of software testing, many activities fulfilled by humans are implicitly error prone [9]. A high number of tasks require software engineers to identify the available resources to assign to a task, and focus their effort. Although a lot of effort, and resources are being invested into software testing, the accuracy and the quality of testing is not quite splendid. There are a high number of applications of machine learning in the field of software testing. The most important ones include software defect prediction, test planning, test case management, and debugging [10].

The usage of topic modeling enables us to perform complex tasks such as classification, retrieval, and analysis on large chunks of textual data. Such complex tasks can also be performed using test cases as input data. Topic models can be applied to cluster the test cases, identify the redundant test cases etc. But most of the research done using topic models has been performed on textual data. Very few researchers tried to apply topic models for test cases [11]–[13], which leaves a research gap. Such high potential and interesting applications of topic models has motivated us to investigate this research gap. Apart from this, most of the studies done on topic models have been on, applying the topic models for text analysis, and text classification. Very few research has been done on evaluating the quality of the topic models, and the topics generated by them. This is another research gap that must be investigated.

In our research, we intend to investigate the applicability of topic models for test case prioritization, and thereby combining the two research gaps, that we mentioned in the above paragraph. We achieve this by applying topic models on test cases obtained from a selected test case repository, and evaluating the quality of the generated topics in terms of user interpretability for an interpretability task, thereby evaluating the quality of topic models. Through this, we also address the research gap of the lack of studies that used test cases as the dataset for topic models. We have chosen an interpretability task because, it is important both in case of test cases, and topic models as well. Each test case is an independent set of test conditions that lets the user verify, if a particular product or system meets the requirements. In other words, a test case describes the typical behavior of a system [14]. Hence, it is very important for these test cases to be user interpretable, so that the user can better understand the test cases and prioritize them to test the product, or system under evaluation. Interpretability is essential not only in case of test cases, but also machine learning algorithms. In [15], Lipton et al. discusses that, in order for a machine learning algorithm, or model to be trusted, it must be interpretable apart from being good.

The quality of the topic models can be evaluated in different ways. In this document we also investigate the various quality criteria, evaluation methods, and metrics that can be used to evaluate topic models. We achieved this by conducting a systematic mapping study. Based on the results of the mapping study, we have chosen two quality criteria, namely generalizability, and interpretability. The motivation for selecting these two quality criteria has been clearly described in Section5.2.9. Generalizability can be defined as the extent to which a particular topic can be generalized, in a given set of data [16]. Whereas interpretability can be defined as the extent to which a particular topic is interpretable, i.e. the terms defining the topic from a consistent and coherent meaning that can be understood by humans. Two optimized topic models have been generated for these quality criteria; one for generalizability (T_G), and the other for interpretability (T_I). We evaluated the performance of these two topic models through an experiment designed for an interpretability task. The goal of the experiment was to determine whether it matters if you use a topic model that is optimized for generalizability, or interpretability for the task of interpreting the topic model's topics. This was to test our assumption that, in order to achieve better prioritization results, a task specific optimization of topic models is more suitable (in our case, T_I for an interpretability task). More detailed description regarding our assumption has been provided in Section 4.2.2.2, along with an example.

The contents of the document are organized in the following way. Chapter 2 provides the background knowledge of the chosen research area, and works related to software engineering and topic models. Chapter 3 highlights the aims and objectives of the authors, and the research questions that the authors intend to answer. The research methods that were used in this study were described in chapter 4. The results of the systematic mapping study have been explained in chapter 5. Whereas chapter 6 provides an overview of the experimental results. Analysis, and discussion was presented in chapter 7. Chapter 8 outlines the conclusion and future work. Apart from these, Chapter 8 also discusses, what our results mean in case of test case prioritization.

2 BACKGROUND AND RELATED WORK

In this chapter, we discuss about topic models in detail. Through this chapter, the readers will have answers to questions such as what is a topic model, why is it used, what are the applications of topic models, how do they work etc. We have also explained the working of topic models in detail, with examples. The readers will also be introduced to the LDA topic model. Furthermore, we will also discuss about test case prioritization, and the previous research that has been done in relation to topic models, and test case prioritization. This chapter also provides an insight into the applications of machine learning, and topic models in the field of software engineering.

2.1 Topic Modeling

Topic modeling is an unsupervised method of machine learning, where algorithms are used to uncover the latent thematic structure in document collections. These are used for analyzing large sets of unlabeled text. Topic models help to organize, understand and summarize the vast archives [17].

Topic modeling is widely used in various number of applications. Beside, exploring scientific data, they are also used in e-discovery and social media. In natural language processing, topic models show positive results when engaged with tasks of document classification, topic tracking, event detection, word sense disambiguation, POS tagging etc., [10], [18]. Topic models have been introduced with a generative model named Latent Dirichlet Allocation (LDA) [17]. Since then, many models have been explored based on LDA.

2.1.1 How does a topic model work?

Now that we have seen what topic models are, let us see how they do what they do, in the simplest way possible. A topic model takes a set of documents as an input. Then, the user defines the desired number of topics (k) that are to be retrieved from the given set of documents. Then the topic model analyses the set of documents and finds the k-topics that best describe the documents. The below figures show in brief, how topic models work.

Let us consider a set of newspaper articles as our documents (see Figure 1).

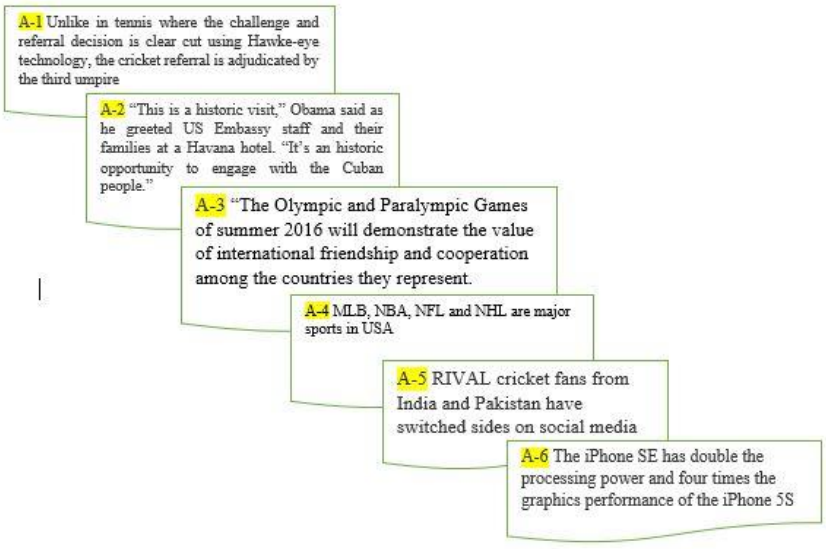

Figure 1: Sample data

If the user chooses the number of topics to be k=3, then the expected topics may be about Sports, technology and relation between countries, as shown in Figure 2. In

addition to the most frequently occurring words in each topic, we also get the proportion of each article in that topic. We can observe that the article A4, entirely deals with sports, the document-topic proportion of article A4 is very high for topic 3 (if doc-topic proportion of article 4 is 0.99 for topic 3, this means 99% of article completely falls under topic 3). Also in the case of article 3, where it explains about the Olympics and foreign relations, the doc-topic proportions of article 3 may be 0.5 for topic-3, and 0.5 for topic-2. This can be seen in the Figure 2, the article 3 falls between the topic 3 and topic 1. In the case of article 5, which describes the rivalry between cricket fans of two countries on social media. As this article is a mixture of sports, technology and foreign relations, we can observe that this article lies in the center of triangle, in Figure 2, stating that doc-topic proportions for this article may be similar for all the topics (0.4, 0.3, and 0.3).

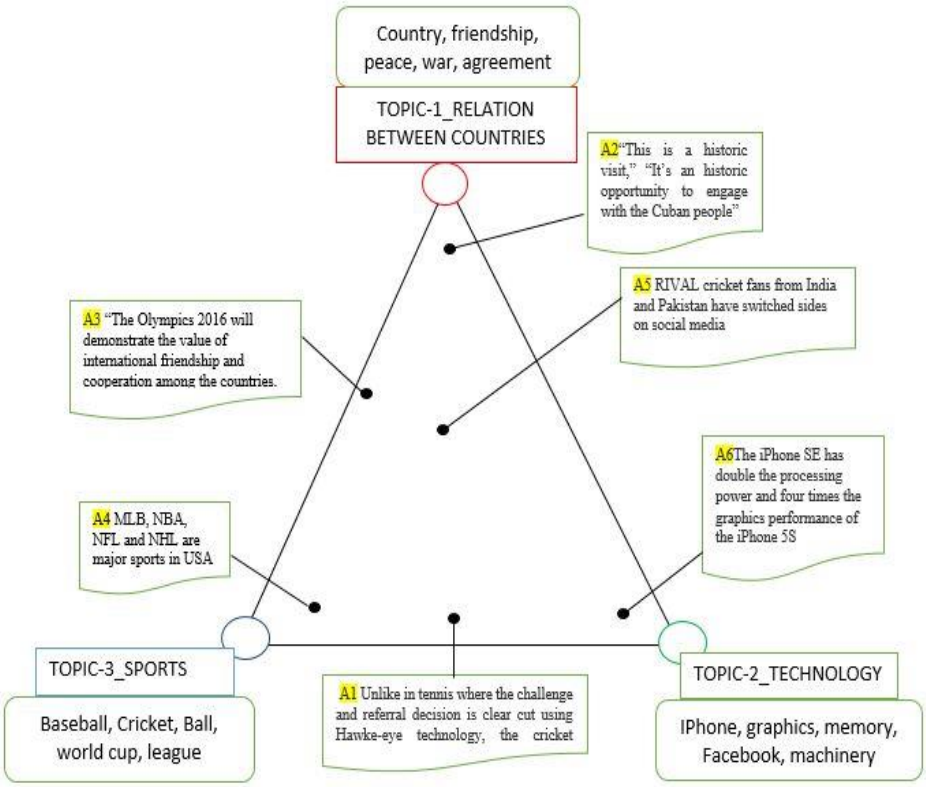

Figure 2: Topics generated by the topic model

In the above example, the topic interpretability is high i.e., when a subject takes a look at the topics, he/she can easily understand the topics and the story behind them. But, if we consider another example, say Figure 3, where the input data is the reviews from the website trip advisor. The number of topics here is same as the above example, that is 3. However, it can be noticed that the resulting topics are not that easy to interpret (Table 1).

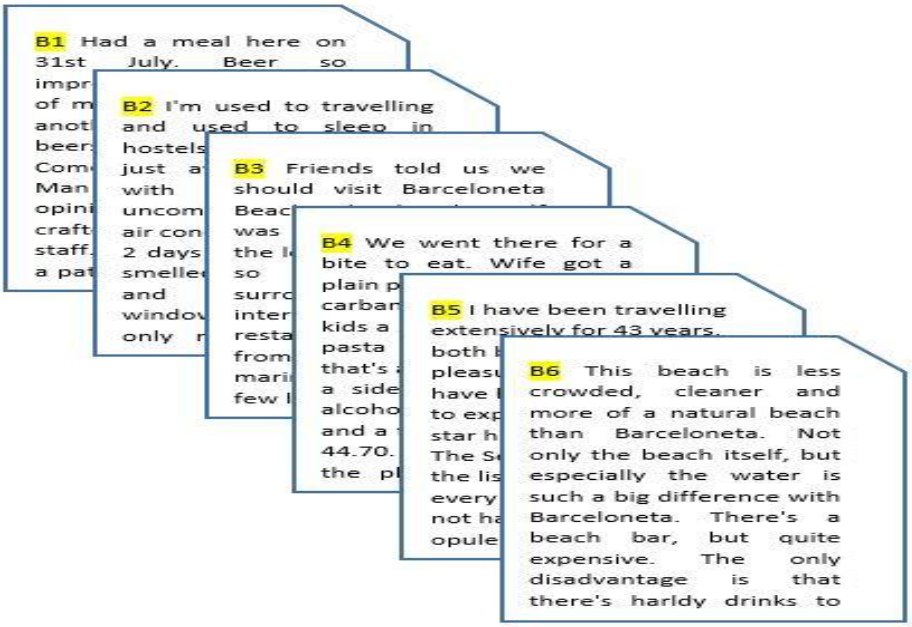

Figure 3: Another set of sample data

The topics that have been generated are:

Table 1: Topics with low interpretability

TOPIC_1	TOPIC_2	TOPIC_3
Spaghetti	Stay	Beach
Told	Make	walk
Plain	People	Beer
Happy	Cheap	Food
Staff	sincerely	barcelonata

Here, the input data is a mixture of documents containing reviews about restaurants, hotels, and beach. Looking at the words in Topic_1 one may think that the topic might be about food, seeing the word "spaghetti"; and that the topic might be about a restaurant that made people "happy". It can be seen that Topic_2 says something about "stay", but it is difficult to interpret the word "cheap", as it can either mean cheap price, or cheap quality. Topic_3 has the words "beer", "beach", "food" etc., and can be interpreted as having some beer, and food in the beach. Overall, all the three topics are hardly interpretable. This may be due to the parameters involved in the topic model. Because the topics might have been more clear, and interpretable if the number of topics (k) was more than 3. Hence, number of topics (k), also plays a crucial role, in the interpretability of a topic model.

2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a basic and most popular topic model. It is a three level hierarchical Bayesian model in which the generative model presents the creation of documents from the dataset. Each item of the set is modeled as a finite mixture over an underlying set of topic probabilities [17].

In LDA, the data is in the form of collection of documents. Each document is treated as a collection of words. In this algorithm, it is assumed that each document is represented as mixture of latent topics and each topic is represented as a mixture over words. The process observed in the LDA model is [1]:

- For every topic, sample a distribution over words from a dirichlet prior
- For every document, sample a distribution over topics from a dirichlet prior
- For every word in the document,
  - Sample a topic from the document's topic distribution
  - Sample a word from the document's word distribution

- Observe the word

The main assumption behind the LDA model is that the document collections have hidden topics in the form of multinomial distribution of words. The users are presented by its top-N highest probability words. Before looking at the graphical model, we provide some basic definitions [17].

Word: It is the smallest unit of discrete data. The indexing is given {1,..V}

Document: It is sequence of N-words denoted by W=(w1,w2,w3…)

Corpus: It is collection of M documents denoted by D={W1,W2..}

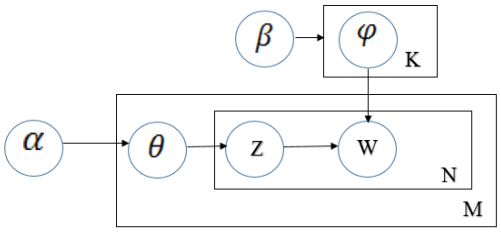

Figure 4: Graphical model for LDA using the plate [1] notation

M is number of documents to analyze
N is number of words to analyze
K is number of topics to analyze
α-is the parameter for per-document topic distribution
β-is the parameter for per-topic word distribution
φ-is word distribution for topic k
θ-is topic distribution for document i
z (i, j) is topic assignment for w(i,j)(j^{th} word in i^{th} document)

2.3 Evaluating topic models

The usage of topic modelling has been increasing significantly for analyzing large set of unstructured text collections. Many models are being evolved based on the usage and applicability. The unsupervised nature of topic model makes the evaluation tasks difficult [19]. No standard universal evaluation method is present for all the topic models. Hence, there is a need for finding the best and most suitable evaluation methods for respective models [20]. In our research, we evaluate the topic models based on the interpretability quality criteria, through an interpretability task. Furthermore, there are several methods to evaluate the quality of topic models based on different quality criteria. We have performed a systematic mapping study, and presented the various methods, that we have identified, using which one can evaluate the topic models, in Chapter 5. Based on the results, we chose a few quality criteria, and evaluation methods, and used them to optimize the selected topic model, which was clearly discussed in Chapter 6.

2.4 Machine learning in software engineering

Many areas in the software engineering have witnessed the use of machine learning algorithms. In software engineering, these algorithms have been mainly used for estimating the software size, development cost and effort involved [21]. It is also used to predicting the defects, quality, and reliability of software [9]. As software systems tend to grow into more complex and complicated, the need for automated software testing methods emerges [22]. Machine learning algorithms have been proven to be of great use in this automation process [22]. Some of the software engineering tasks where the machine learning algorithms have been implemented are:

2.4.1 Software quality prediction

Many machine learning techniques have been applied to predict the software quality. In 1998, Evett et al. presented a case study that included genetic programming for software quality modelling. Here, the model accepts the metrics of the software development as the input, and predicts the number of faults that might occur in the later stages of the development [23]. In 2000, Ganesan et al. used Case-based reasoning (CBR) for predicting the significant measures of software quality. CBR learning technique takes the data from the previous similar projects, and uses that data to address the issues in the current project [24]. Ping Guo et al. believes that software metrics help in knowing the information about the software code in different stages of development. They predicted the quality of the software using Expectation-Maximization algorithm (EM algorithm), which analyzes the software metrics, and understands the relationship between them. This helps in capturing the defect prone modules, and thereby allowing to take actions against them, at a very early stage of development [25].

2.4.2 Software size estimation

In 2000, Dolado et al. introduced a component based method, for estimating the size of the software. Here, Neural Networks (NN), and Genetic Programming (GP) were used to evaluate the working of component based method. The authors have compared NN learning to other linear regression methods, and stated that this model works very well in identifying the non-linear relationships between lines of code, and the number of elements. Here, GP was used as an automated symbolic regression for deriving the equations. The authors stated that, the derived equations are apprehensible and provided better results than the regression equations [26].

2.4.3 Software defect prediction

Fenton et al., introduced a model for software defect prediction based on Bayesian-belief Networks (BBN). A BBN is a graph structure that represents the relationships among the variables. The graph consists of nodes and edges. The nodes represent the variables, and the edges explain about their relationship with each other. In the prototype given by Fenton et al., the nodes represent the stages of software development life cycle. The variables that were considered are problem complexity, design size, defects introduced, design effort, etc. Using these edges, we can predict new defects that are likely to be introduced. For example, we can know that, any mismatch between problem complexity and design effort is likely to introduce new defects [27].

2.4.4 Software release timing

In 1999, Dohi et al., proposed a technique to estimate the optimal software release timing, which adapts the criteria of cost minimization via artificial neural networks. By using the graphical approach method provided in the article, managers can choose the optimal release time from the various estimates of fault-detection time [28], [29].

2.5 Topic modeling and Software engineering

Many researchers have started to implement Information Retrieval (IR) methods to analyze the textual information in the software artifacts [30]. There are many prominent informational retrieval techniques present. Among these, Topic modelling based on LDA is also used. Initially, LDA is only used upon natural language processing tasks of enormous textual data. In recent studies, LDA is also emerged as one of the promising IR technique that is being used on software artifacts [18].

2.5.1 Software traceability

In 2010, Hazeline et al. proposed a method for automating the traceability links, by combining traceability with topic modelling. The proposed approach automatically records the traceability links in the development life cycle, and learns a topic model over software artifacts. The learned topic model is used to categorize the artifacts

semantically, and helps to visualize the software system in a topical manner. The approach has been evaluated on different data sets, and results showed that it is very efficient [10].

2.5.2 Evolution analysis

In software development, there is a custom to store the data of the previous projects. This data may be structured or unstructured. For every new change that is made, the data becomes messy, and gradually it becomes hard to understand. The data from all the software artifacts is used to generate topics via topic modelling algorithm. In 2010 Thomas et al., proposes a method to software evolution analysis, by using topic modelling. The motivation behind this was that, by understanding how the topics evolves or changes in the software project over the time of development, manager or other stakeholder can monitor the direction of the topic [31]. They evaluated the model by performing a case study on a JHotDraw software system, and suggested that using the topic model helped the stakeholders in notifying the changes in software activities.

2.5.3 Source code labeling

Andrea et al., proposed source code labeling by using the LDA topic model. Here, authors used the code and comments to extract the latent topics. The evaluation is done by using the human subjects. They were instructed to label the ten classes of two java systems (JHotDraw and eXVantage) with a set of 10 keywords. These keywords are compared to the keywords extracted by LDA. They have computed the overlap and stated that results are found to be efficient and highly over lapping [32].

Along with the above tasks, studies have proven that topic models show effective results also when they are applied on the tasks of bug localization [33], impact analysis [34], and expert identification [35].

2.5.4 Test case prioritization

It is the most related task to our studies. In [11], Thomas et al. proposed a topic based black-box static test case prioritization using topic modelling. Here, they used LDA topic model to extract topics from each test case. By using these topics, they prioritized the test cases by calculating the dissimilarity between pairs of the test case. They evaluated the results with results of other existing prioritization technique and justified that their technique works very well to detect the unique faults in the software [11].

2.6 Research relating topic models and test case prioritization

Testing a software product without any priority in test cases has proven to be time and effort consuming task. Test cases are defined to be the list of actions which need to be executed to certify the functionality of the product or software. Ordering the crucial test cases in sequence helps testers to detect the faults in early stages and thus facilitates the testing in a much efficient way [36]. The technique of scheduling and executing the test cases in an order is called as test case prioritization. This technique saves time, and cost in testing phase by executing the most crucial and critical test cases in early stages. Test engineers may schedule test cases based on their goal of achievement in testing. The main goals include code coverage, rate of fault detection etc. To achieve this goals, a great number of prioritization techniques have been proposed based on customer requirements, chronographic history, cost effective and code coverage criteria [37].

Researchers have formulated various types of test case prioritization techniques to achieve above mentioned goals. There is some limited research also present, where test case prioritization is done by including only the linguistic data of the test case. Linguistic data in the test cases include comments, identifier names, string literals etc., Topic modelling is also included in some test case prioritization techniques [11], [12], [38]. Thomas et al. proposed a topic based black-box static test case prioritization using topic

modelling [11]. Here, they used LDA topic model to extract topics from each test case. By using these topics, they prioritized the test cases by calculating the dissimilarity between pairs of the test case. They evaluated the results with results of other existing prioritization technique and justified that their technique works very well to detect the unique faults in the software [11].

In [12], Hemmati et al. performed a comparison of three different test case prioritization strategies that focus on aspects such as code coverage, test case diversity, and risk driven techniques. Unterkalmsteiner et al. [13], proposed a new way to select the test cases by making use of probabilistic topic models, so as to help the domain experts. Industrial test cases were used as a dataset in this research. The results were validated through a case study, and the domain experts were found to have achieved better results in test case selection using this approach.

3 RESEARCH DESIGN

3.1 Aim

As discussed in Chapter 1, we intend to investigate the applicability of topic models for test case prioritization, by evaluating the topic model quality for an interpretability task. But in order to do that, first we must gain knowledge of all the methods, metrics, and criteria that can be used to evaluate the topic models. Hence the aim of the research should be in such a way that, it will let us gain an overview of the desired knowledge, and then allow us to apply the gained knowledge for our investigation. Hence, the aim of our research is as follows.

The main aim of this research is to identify the different topic model quality criteria, evaluation methods, and metrics that can be used to evaluate the quality of topic models. Based on this, we intend to choose some of these quality criteria, and evaluation methods, and optimize the chosen topic model for these criteria using those evaluation methods. We will then compare the performance of these topic models by selecting a few metrics, and conduct an experiment. Depending on the results, the topic model that provides the topics with better interpretability will be determined.

3.2 Objectives

To achieve the above mentioned aim, certain objectives have been set to help us reach the aim. The main research objectives to achieve our aim are:

- **O1:** Conduct a systematic mapping study to identify various methods that are present to evaluate the quality of topic models.
- **O2:** Choose appropriate quality criteria, evaluation methods, and metrics for the experiment.
- **O3:** Understand the selected topic model and data set that are going to be used for the experiment.
- **O4:** Determine the optimal number of topics, given a particular topic model quality evaluation method.
- **O5:** Generate the topic models as per the optimal number of topics.
- **O6:** Design and implement an experimental task using human subjects.
- **O7:** Collect, analyze and compare the results for the used topic models.
- **O8:** Determine the topic model with the best results for the designed task.

3.3 Research questions and Motivation

For achieving our objectives, the following research questions were formulated.

RQ1) What are the different evaluation methods that can be used to evaluate the quality of topic model?

RQ 1.1) What is the previous research that has been done in the context of evaluating the topic models?

RQ 1.2) What are the various types of evaluation methods that can be used to evaluate the performance of topic models?

RQ 1.3) What are the various quality criteria, and metrics that have been used in the evaluation of topic models?

Through our first research question we intend to get an overview of the various methods, and metrics that can be used for evaluating topic models. Answering this research question will also help us in finding the most suitable metric for our experiment, through which we can compare the performance of the topic models. The RQ1 has been divided further into three sub research questions RQ 1.1, RQ 1.2, RQ 1.3. These sub research questions will be used as the research questions for the systematic mapping study, which has been discussed in detail in Chapter 4.

RQ2) Among the two topic models that are optimized for different quality criteria, which topic model provides topics that have better interpretability, when used for an interpretability task?

The methods, and metrics found by answering the first research question, will be used in answering our second research question, which is to determine the topic model with the better interpretability results. Through this research question we want to analyze the effectiveness of two topic models, that have been generated for different quality criteria, when applied for the same task. The motivation behind this research question is that, with the increasing use of topic models, humans are also being involved in a way that, they are being directly exposed to the output of topic models [39]. Hence, it is vital that the topics generated by a topic model have good interpretability. Apart from this, very few, or no research has been done, using an interpretability task to evaluate the topic models for test case prioritization. So, through this research question, we want to determine the topic model which produces the topics, that are more easy to interpret.

4 RESEARCH METHODOLOGY

The research methodologies that we have chosen to find answers to our research questions are Systematic Mapping Study, and Experiment. Figure 5 illustrates how the chosen research methods contribute to answering the research questions, achieving thereby the stated objectives and overall aim of the thesis. By conducting the systematic mapping study, we intend to get an overview of the research that has been done on evaluating the quality of topic models. This includes various topic model evaluation methods, metrics etc. that can be used for evaluation. Thus, conducting a systematic mapping study would be appropriate as it helps us answer our research question (RQ1), and objectives (O1, O2). Once, the study has been conducted and we have identified the suitable quality criteria, evaluation methods and metrics, we will conduct an experiment, where we would like to optimize the topic model for the selected quality criteria, using appropriate evaluation methods; on the selected dataset. The topics from these topic models will then be given to the subjects as a part of an experimental task. The identified metrics will be calculated for both the topic models, and the results will be used to determine the topic model that performed well for the designed task. Conducting an experiment would help us answer our research question (RQ2) to determine which topic models performs well for the task at hand, thereby helping us meet our remaining research objectives (O3-O8); thus completing the aim of this research.

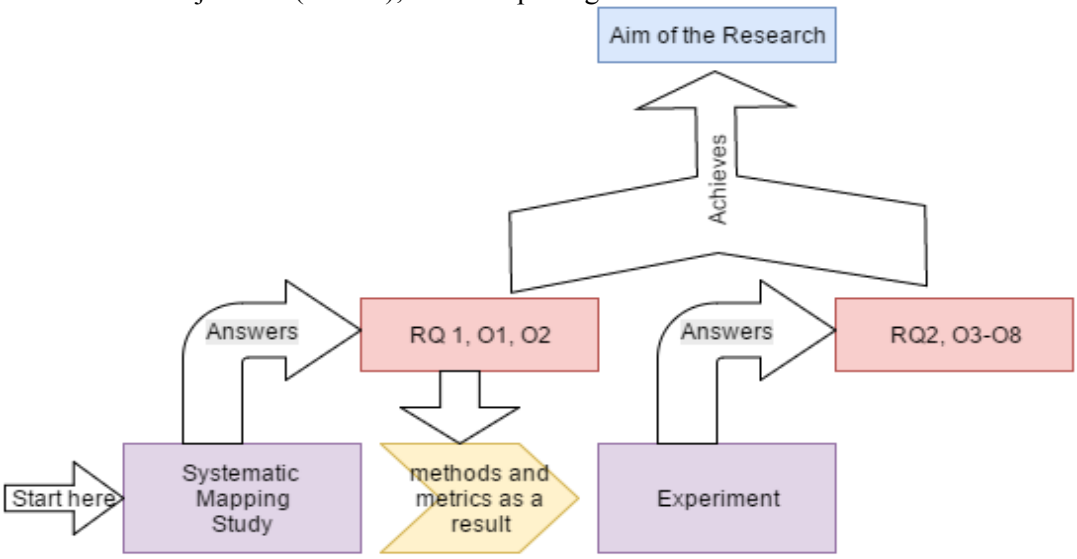

Figure 5: Research methods and what they achieve

4.1 Systematic Mapping Study

A systematic mapping study is a methodical procedure that allows the researcher to identify the nature of the research, as well as the extent to which it has been conducted [40]. A systematic mapping study is different from a systematic literature study. In a systematic literature review, the researcher identifies, evaluates, and interprets all of the existing research related to a topic, or object of interest; in order to answer the research question [41]. Where as in a systematic mapping study, the researcher tries to provide an overview of the research that has been done in a particular research area, rather than trying to find answers to a particular research question. In other words, with a systematic mapping study, the coverage of research in a particular area can be identified and described. This can be achieved by analyzing the frequency of publications done in that particular research area per year [42], various venues that involve the research in that area, or several other factors that the researcher is interested in. The results of a

systematic mapping study are summarized, and provided in visual forms that are often referred to as maps [42].

The systematic mapping study in our research involves the following steps, based on the systematic mapping process detailed by Petersen et al. [42].

Step 1: Formulation of research questions for the study
Step 2: Define the search string
Step 3: Perform the search in selected databases
Step 4: Filtration based on inclusion and exclusion criteria
Step 5: Extract the desired data
Step 6: Prepare the maps

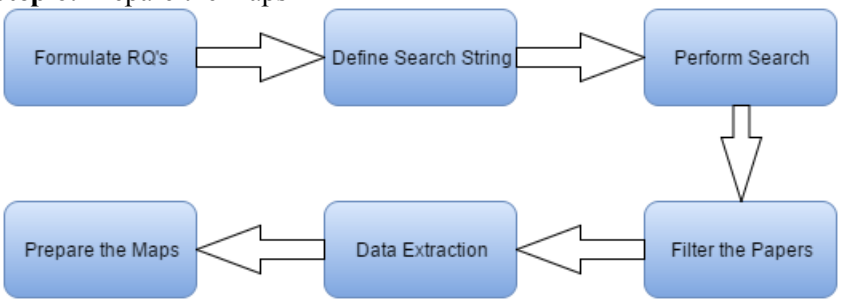

Figure 6: Systematic mapping study process

Upon successful completion of the systematic mapping study process, the outcome that we expect to achieve is an overview on the amount of research that has been done on topic model evaluation, various methods, metrics, and quality criteria that we can use in our research. The following sections explain each step in Figure 6, in a detailed way, as per our research.

4.1.1 Formulation of research questions for the study

The main objective behind the execution of this systematic mapping study is to gain an overview of the amount of research done on the evaluation of topic models, and various methods that can be used to evaluate the quality of the topic models, which in turn is our RQ1 (See Section 3.3). The sub research questions of RQ1 have been used as the review questions for the systematic mapping study:

R.1) What is the previous research that has been done in the context of evaluating the topic models?

R.2) What are the various types of evaluation methods that can be used to evaluate the performance of topic models?

R.3) What are the various quality criteria, and metrics that have been used in the evaluation of topic models?

By finding answers to the above research questions, we intend to gain an overall knowledge on the various existing topic model evaluation techniques. This will help us in choosing topic model quality criteria, and appropriate methods that can be used in our experiment, which constitutes the second part of our thesis.

4.1.2 Defining the search string

4.1.2.1 Identifying the keywords

Based on the above formulated research questions, the following keywords have been identified to define our search string.

Keywords: topic model, topic models, LDA, HLDA, approaches, techniques, methods, frameworks, evaluate, measure, analyze, assess, tools.

In order to increase the scope of the search process and to cover more number of articles, possible synonyms and alternative words have also been included in the keywords.

4.1.2.2 Selection of databases

Inspec and Scopus databases have been selected as the target databases to find the relevant articles for our study. One of the major reasons for choosing them was that, both of them have an efficient search features, that provide more control to the researcher over the search process [43]; thereby allowing the researcher to find exact desired results. Inspec is also highly recommended for looking up research articles, when compared to other databases [44]. Both of the selected databases are reference databases and include results from various scientific databases such as ACM, Elsevier, IEEE Xplore, Springer etc. Google scholar was not chosen because, the search feature of google scholar is not as efficient as the selected databases. Another drawback is that google scholar generates too many search results, and does not constrain the scope of the search results. This makes it hard for the researchers to find the relevant and useful articles. Furthermore, there are also some doubts regarding the quality of the scholarly articles in google scholar [44].

4.1.2.3 Search string formulation

The process of conducting search process among the selected databases has been divided among the authors of this thesis. This is done by assigning one particular database to each author. The details of the authors, and the databases assigned to them are provided in Table 2.

Table 2: Authors and Databases

S.No	Name of the author	Database assigned
1	Veer Reddy Sathi	Scopus
2	Jai Simha Ramanujapura	Engineering Village (Inspec)

4.1.2.4 Search string

As per the databases assigned, each of the authors were responsible for defining the search string for their database. As a result, two search strings were defined for the databases. The search strings are as shown in Table 3.

Table 3: Databases and search strings

S.No	Database	Search string
1	Scopus	((((approach OR approaches OR method OR methods OR framework OR frameworks OR tool OR tools OR technique OR techniques) WN KY) AND ((evaluation OR measure OR analyze OR evaluate OR assess) WN TI)) AND ((topic model OR topic models OR LDA OR HLDA) WN KY)) Where WN KY= In Subject/Title/Abstract
2	Inspec	(TITLE-ABS-KEY ("framework" OR "frameworks" OR "technique" OR "techniques" OR "method" OR "methods" OR "tool" OR "tools") AND TITLE-ABS-KEY ("measure" OR "analyze" OR "evaluate" OR "assess") AND TITLE-ABS-KEY ("topic model" OR "topic models"))

The procedure in which the study has been conducted at each step of the process is explained in detail, in chapter 5.

4.1.3 Filtering the papers

The initial results obtained through the search process have been filtered and refined to get the final set of articles, using the following inclusion and exclusion criteria.

Inclusion criteria:

- Articles between the timeline of 2003 and 2016
- Only research articles, journals, and conference papers
- Documents written in English language

Exclusion criteria:

- Articles that are unavailable as full text documents
- Articles that are not peer-reviewed
- Studies that focus on topic models other than LDA, and HLDA
- Studies that lack an evaluative perspective on topic models

The timeline has been chosen from 2003 because, the LDA topic model was initially introduced by Blei et al. in 2003 [17]. And so, having 2003 as the start of the timeline will provide us access to all the research that was conducted on LDA, which in turn would be very relative and helpful to our research. The articles that were not available as free pdf's in the databases were excluded as we have very limited resources to carry out the research. The articles that were not peer-reviewed have also been neglected so as to ensure the quality of the results. Also, the studies that do not have an evaluative perspective of topic models have also been rejected as they do not relate to neither our research, nor the research questions. The above mentioned criteria have been applied at several stages of the search process (see Figure 10, in Section 5.1) until the final set of papers were identified for the study.

4.1.4 Data extraction

A data extraction form has been created to extract the desired data from the selected articles. The articles obtained from the search process were split into equal proportions, and the data extraction work was equally divided between both the authors. The data extraction form has been presented in Table 4, and the review questions (Section 4.1.1) that they answer, have also been presented. The results and maps are presented in Chapter 5.

Table 4: Data Extraction Form

Data extraction form field	Relevant RQ
Year	R1
Country	R1
Venue	R1
Research areas/ Key words	R1
Topic models	R1
Evaluation methods	R2
Dataset	R1
Research method	R1
Quality criteria/ metrics used for evaluation	R3

Every article selected for the systematic mapping study will be evaluated as to extract answer for each field of the data extraction form. The questions in the data extraction form have been presented in Appendix 10.1, Table 20. The questions in the data extraction form have been formed in such a way that each question in the data extraction form is related to a research question mentioned in Section 4.1.1.

4.1.5 Why a systematic mapping study? Why not a systematic literature review?

One of the primary reasons for not choosing a systematic mapping study over a systematic literature review, as a research method is, since our objective is only to gain an overview of the various existing methods to evaluate topic models, and not to study them in depth, a systematic mapping study seemed to be more efficient. Furthermore, performing a systematic literature review would involve a great deal of effort and time [40]. As we are constrained by time to complete the thesis; planning, conducting, and reporting a systematic literature study seemed to be a practically impossible task.

4.2 Experiment

After conducting a systematic mapping study and answering our first research question, the second research question of our study is answered by conducting an experiment. Other possible alternative that we could have considered instead of an experiment is a case study. An experiment is an investigation of a hypothesis, where the independent variables are manipulated, to measure their effect on the dependent variables. An experiment allows us to determine how the variables are related, and helps us determine if there is any cause effect relation between the variables. Whereas, a case study can be defined as an empirical investigation of a phenomena within its real life scenario [45].

In our research, we intend to evaluate the performance of two different topic models for the same interpretability task. So far, the most effective way to measure the interpretability of a topic model is through human evaluation. So, we also need to involve humans for evaluating the topics that are generated by two different topic models. This provision of involving humans to evaluate the topic interpretability is absent in the case studies. So, we have excluded case study in the research. Hence, we chose a controlled experiment as our research method. An experiment in which a single independent variable is altered at a time to see the change on the dependent variables, is considered as controlled experiment. If two sample populations are exposed to two different treatments, then it is labeled as comparative experiment [46]. In our experiment, the only variable that we change to know the effect is the number of topics. The two conditions i.e. Topics from the two topic models are evaluated using the same human subjects. Hence, it is recognized as controlled experiment in our research.

Among the various criteria to evaluate topic model quality that have been found in the systematic mapping study, we chose topic model interpretability, and generalizability for our experiment. We provide a motivation for this choice in Section 5.2.9. The experimental task and process are described next.

4.2.1 Experimental task and process:

The main task for our experiment is to evaluate the interpretability and understandability of test case topics generated by two topic models, T_G and T_I. The main objective behind this task is to see, if a topic model that is optimized for generalizability yields the same performance as that of a topic model optimized for interpretability. The dataset that will be used for this purpose is, the test cases obtained from the Mozilla Moztrap test case repository. The test case data that will be trained to the topic models is, the test case name, description, instruction steps, and the expected steps. So, the resulting topics from the two topic models must provide an overview of the various concepts that are present in those test cases. All of the test cases obtained from Moztrap repository have pre-defined tags assigned to them, in the repository. These tags can be defined as the labels or identifiers, that describe the test case. For instance, a tag can provide information as to which category, or product a particular test case might belong to. These tags will be used in the later stages of the experiment (described in Section 4.2.1.2). We intend to verify and validate the interpretability of these topics by assigning them to a sample of 20 human participants, through the experiment. The experimental procedure is visually presented in Figure 7.

4.2.1.1 Phase A: Generating the topics

This is the phase, in which all the preparatory work for the experiment is done.

4.2.1.1.1 Step 1 – Optimize LDA to generate two topic models for Generalizability, and Interpretability

During this step, the LDA topic model will be optimized to generate two topic models. Among the generated topic models, one will be focusing on topic optimization based on generalizability, and the other on topic interpretability.

4.2.1.1.2 Step 2 – Train the topic models on the selected data set to generate the topics

Once the topic models are generated, they will be trained on the selected set of test cases to generate the topics, as discussed at the beginning of Section 4.2.1. This step marks the end of Phase – A of the experiment.

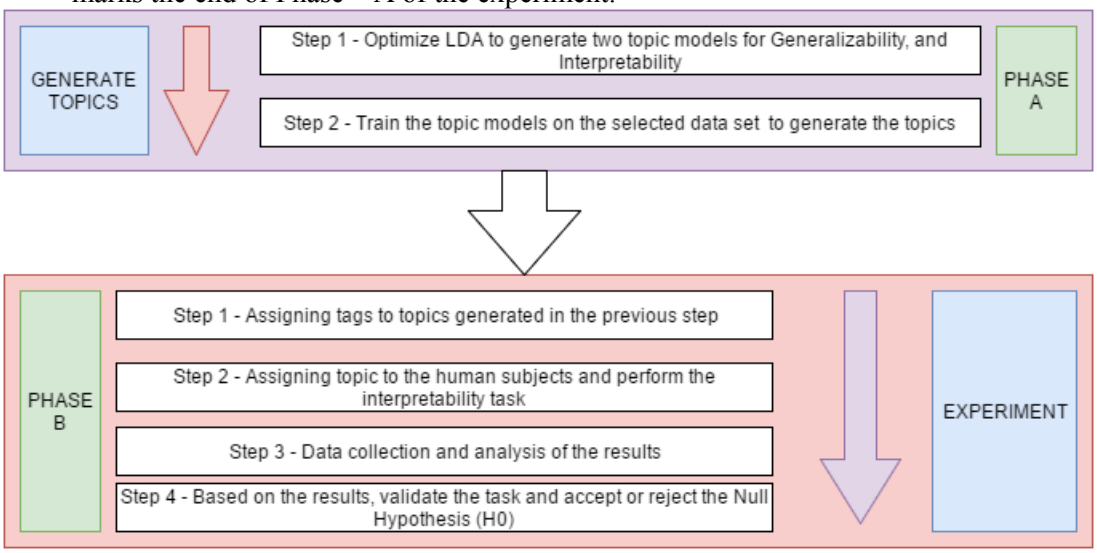

Figure 7: Experimental Procedure

4.2.1.2 Phase B: Experimental execution and tasks

This is the actual part of the experiment. The topics generated from the previous step will be assigned tags. The topic-tag information will be kept by the researchers, and only the topics will be given to the human subjects, with the tags as an alphabetically sorted list. Then the subjects assign tags to the topics, from the list of tags given to them. The results will then be evaluated to determine which topics performed better (i.e, topics based on interpretability and generalizability).

4.2.1.2.1 Step 1 – Assigning tags to topics

During this step of the experiment, the topic proportions for the topics generated in the above step will be calculated, and the researchers will assign tags to the topics. All the test cases have pre-existing tags that were available from the Moztrap test case repository (also discussed at the beginning of Section 4.2.1). However, these tags have been eliminated when the test cases were trained to the topic models, as there is a risk that the presence of tags may result in biased topics. The tags were assigned to the topics based on their proportion in a given test case. This is better understood with an example. Let us assume that, a test case 1 has the tags desktop, and camera. The test case 1 is a mixture of three topics A, B, and C in the proportions 62%, 18%, and 20%. Then, we will assign the tags desktop, and camera to Topic A; as it the topic with the highest proportion in test case 1. Tags will be assigned to all the topics that have been generated using the generalizability and interpretability models. Each topic can be a part of several other test cases. So, it is possible that a topic can have more than one tag. This can be better understood with an example (Figure 8). A brief description of test case structure, and tags has been provided in Appendix 10.6.

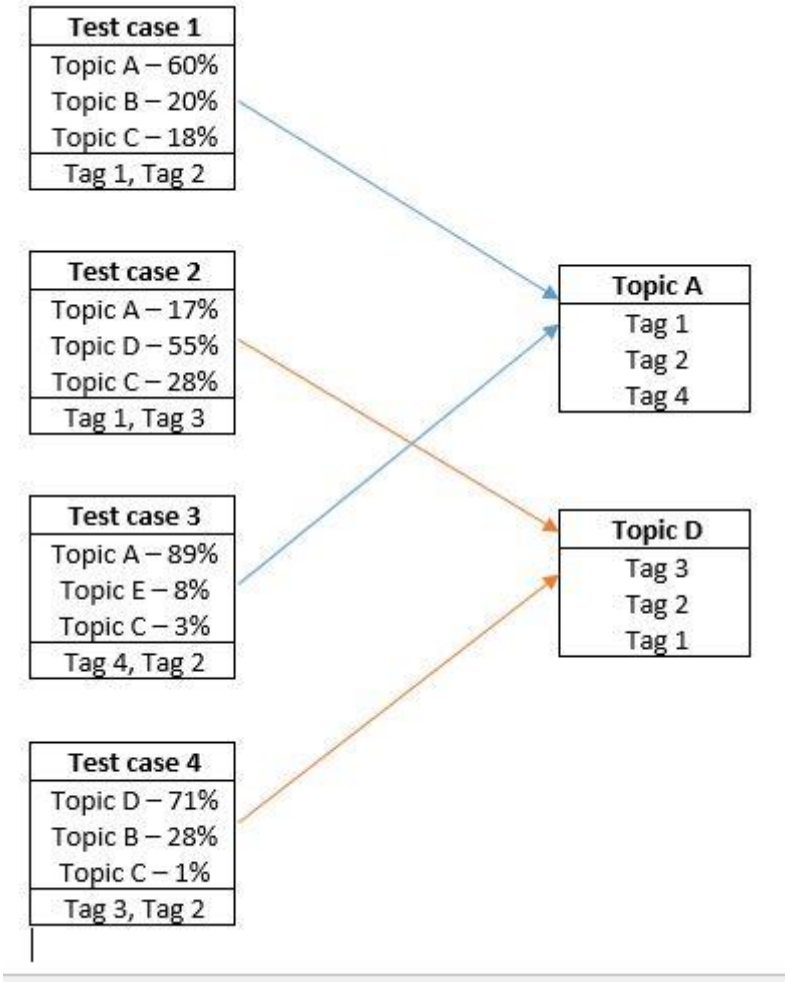

Figure 8: Assigning tags to topics

In Figure 8, we have four test cases that contain topics A, B, C, D, and E in different proportions. Topic A has major proportions in both Test case 1, and Test case 3. So it will be assigned the tags 1, 2, and 4. Similarly for Topic D, tags 1, 2, and 3 will be assigned, as it is the topic with major proportion in test cases 2, and 4. This process will be repeated for all the topics generated by both the topic models.

4.2.1.2.2 Step 2 – Assigning topics to the subjects and perform interpretability task

Once the tags have been assigned to the topics, a group of 20 human subjects will be asked to evaluate, and classify the generated topics by selecting, and assigning tags from a set of tags given to them. To each human subject, 15 topics from the generalizability model, and 15 topics from the interpretability model will be randomly assigned. Only the researchers have the information of the tags, that a particular topic is related to. Also since the dataset under investigation is test cases, not everyone can understand the topics, and assign the tags to the topics. So it is required that the subjects have some technical knowledge. Hence, Masters' students from various departments like software engineering, computer science, signal processing, and tele communication have been selected as the subjects for our experiment.

After the topics have been assigned to the human subjects, their task is to read the terms in each of the topic, and assign tags to those topics. The tags will be provided in an alphabetically sorted list. The subjects have to read the terms in a given topic, and choose a tag from the list of given tags and assign it to the topic. Also the time taken by the subjects to complete the task will also be recorded for the topics of both the models. The subjects have no information that the topics are from two different topic models. As far as the subjects are concerned, they only see a list of 30 topics for which they have to

identify the tags. Also the subjects were asked to perform the task continuously from start to the end, without any distractions or interruptions, so as to eliminate any biases. There was no time restriction given to the subjects. Every subject was allowed to perform the task for as long as he/she wanted.

4.2.1.2.3 Step 3 – Data collection and analysis of results:

The topics and the tags will be provided to the subjects as two different files. The topics will be given in a word document, and the tags will be given in an excel sheet (See Appendix 10.2). The subjects must assign the tags to the topics in the word document. Once this is done for all the topics the tags assigned by the subjects will be compared to that of the original tags of the test cases that the researchers have. Then the selected metrics (Section 4.2.2.4) will be calculated for both the topic models. The calculated results will then be used to study the performance of both the topic models for the task performed. Also the average time taken by the subjects to complete the given task will also be calculated.

Step 4 of Phase – B of the experiment, which deals with the results and null hypothesis, has been discussed in chapters 6, and 7.

4.2.2 Experiment Design

According to wohlin et al. [47], there are different types of experimental designs. The designs range from single factor-simple experiment to multiple factor-complex experiment. The researchers should choose the best design type that fits for their factors, and sample population. In our case, the design type should be suitable for an experiment with one factor and two treatments. The factor in our case is the quality criterion, and the two treatments that should be compared with each other are interpretability and generalizability.

The two design types available in this category are: completely randomized design, and paired comparison design. As our aim is to judge the topic models, by providing the topics generated by both the topic models with the same subject i.e., each subject should use the both the treatments on the same object, we need to choose the paired comparison design. The best statistical methods for this design type are T-test (paired), sign test and Wilcoxon-sign ranked test. The usage of these statistical tests are based on the normality of the data. The normality of the data can be examined by using the Shapiro-wilk test, and QQ-plot [47]. The experiment design outlines the tested hypothesis, the independent and dependent variables, and the environmental setup of the experiment. The statistical hypothesis testing methods will be discussed in Section 7.2, in Chapter 7; after presenting the results of the experiment in Chapter 6.

4.2.2.1 Null Hypothesis (H_0):

The topic model optimized for generalizability (T_G) exhibits the same performance as a topic model optimized for interpretability (T_I); when applied in a task where topic model interpretability is important.

4.2.2.2 Alternate Hypothesis (H_1):

The topic model optimized for generalizability (T_G) exhibits a worse performance, than a topic model optimized for interpretability (T_I); when applied in a task where topic model interpretability is important.

Our assumption is that, to achieve better prioritization results, it is important to optimize the topic model for the particular task at hand. We intend to test this assumption because, we haven't found any studies that have evaluated the topic models from this perspective. Also, let's say for example, if a topic model is optimized for different criteria that are irrelevant to the prioritization task that must be performed, for example a topic model was optimized to be time efficient for an interpretability task. The topic model would aim at generating the topics in as less time as possible. This may produce several faulty results, as there is a chance that some of the topics might not be identified,

and the generated topics might not be interpretable, as the focus is entirely on generating the topics by taking the least time possible. Hence, testing our assumption that topic model optimization should be done depending on the task at hand makes more sense. Hence the hypotheses have been formulated accordingly.

4.2.2.3 Independent variables:

According to Wohlin et al. [47] the variables that can be controlled and manipulated in an experiment are called independent variables. If any changes are made to the independent variables the consequences are reflected on the dependent variables. The independent variables are the selected topic model quality criterion, i.e., Generalizability, and Interpretability. Since the study focuses on evaluating the performance of the topic models that have been optimized for the quality criteria generalizability, and interpretability; the entire study revolves around those two quality criteria. If any changes, were to be made to these quality criteria, the changes will be reflected on all the selected metrics, and the results of the experiment. Hence, they were identified as the independent variables.

4.2.2.4 Dependent variables:

The variables that are affected, if any changes are made to the independent variables are called as dependent variables [47]. The dependent variables in our experiment are the metrics precision, recall, and the run time of the topic models. This is because these variables can be affected depending on the quality criteria for which the topic model has been optimized to.

Precision and Recall are two test accuracy measures, that are used in the information retrieval domain to test the accuracy of a system, when a particular user requested document is retrieved. These measures are used in the information retrieval domain to measure, how well an information retrieval system retrieves the relevant documents requested by a user. The measures are explained as follows [48]:

- **Precision:** It is defined as, the ratio of the number of relevant records retrieved, to the total number of records retrieved.
- **Recall:** Recall is defined as the ratio of number of relevant records retrieved to the total actual number of relevant records present.

Precision and recall are explained with help of confusion matrix with two classes (Figure 9). One is labeled as positive class and other as negative.

	Predicted	
	Positive	Negative
Actual True	TP	FN
Actual False	FP	TN

Figure 9: Confusion matrix depicting true and false positives

In Figure 8, True Positive (TP) is the case where a particular tag (in our case) is true, and was assigned as true. Whereas a False Positive (FP) is a case where, the tag is false, but assigned a true value. Similarly, a True Negative (TN) stands for a case where a tag is false and has been assigned a false value, and a False Negative (FN) is, when a tag is true and has been assigned a false value. Then, the precision can be calculated in terms of true positives, and false positives as follows:

$$Precision = \frac{True\ positives}{False\ positives + True\ positives}$$

The Recall can be calculated in terms of true positives, and false negatives as follows:

$$Recall = \frac{True\ positives}{Flase\ negatives + True\ positives}$$

- **F-measure:** It is the measure that is used to measure the overall quality performance of the model. F-measure is known by calculating the harmonic mean of precision and recall.

$$F\ measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The following example will help to better understand, how the precision, and recall have been calculated in our research.

- **Example:**

Tags identified by the subject: *Dialer, video, streaming*
Actual tags for the topic: *Browser, dialer, video, camera*
Here,
True positives = Number of correct tags identified by the subject = 2 (dialer, video)
False positives = Number of false tags selected by subject = 1 (streaming)
False negatives = number of correct tags that are not identified = 2 (browser, camera)

$$Precision = \frac{TP}{TP+FP} = 0.66$$

$$Recall = \frac{TP}{TP+FN} = 0.5$$

$$F.Measure = 2 * \frac{P * R}{P + R} = 0.5689$$

Our hypothesis is that the interpretability of topic models may vary depending on the quality criteria for which the topic model was optimized. Hence precision, recall, and f-measure were identified as the dependent variables of our experiment. The average time taken by the human subjects is also another dependent variable that may be affected depending on the topic model.

4.2.2.5 Environmental setup:

In this section, we present the details of the environment in which the experiment was conducted (i.e. the topic models were implemented, and the topics were generated). In order to ensure the validity of the data, it was necessary that both the topic models T_I, and T_G were implemented in the same environment. The tool Mallet has been used for topic modelling purpose, and to generate the topics. The source code for the topic model optimization is suitable for Unix like environment. But both of the researchers are windows users. Hence, Cygwin tool was used to generate a Unix like environment in the Windows operating system, so that the shell scrips, and python scripts can be executed. The topics generated after implementing the topic models have been provided to the subjects of the experiment in Microsoft Word, and Excel documents.

Table 5: Environmental Setup of Experiment

Category	Used
Operating System	Windows 8
Processor	Intel® core(TM) i5, 2.4GHz
RAM	8GB
Tools used	Cygwin, Mallet, Microsoft Word and Microsoft Excel (for assigning tags to topics, and topics to subjects)
Programs and Programming Languages	Shell scripts and Python scripts, R programming
Input data for topic modelling	Description and steps of the test cases that are taken from MozTrap testcase repository.

4.2.3 Why MozTrap?

Much research has not been done using test cases as a dataset for topic models. In their work, Unterkalmsteiner et al. [13]; and Thomas et al. [11]; case study has been chosen as the research method. The topic modelling algorithms in their research have been implemented on real data from industries. The reason behind choosing Moztrap was, the repository contains a large set of real world test cases (used for large software systems, e.g. Mozilla Firefox, Firefox OS). The Mozilla MozTrap repository has a fine-grained collection of environment specific test cases, which would be suitable for our experiment [49]. Furthermore, the test cases also contain tags associated with them [50]. These pre-assigned tags are essential in verifying and comparing the tags assigned by the subjects, in our experiment, to measure the selected metrics precision, recall, and f-measure.

4.3 Threats to validity

One of the most important part of any research, is to identify the threats to validity [47]. The threats to validity can be understood as the threats or risks, which if not planned properly, may affect the outcome, or even question the validity of the research. These threats to validity can be classified into four major types [47]. They are: Internal validity, External validity, Construct validity, and Conclusion validity. This section describes how the impact of each of these validity threats have been reduced on our research.

4.3.1 Internal validity

Internal validity is said to exist, if there is a liaison between the independent, and the dependent variables; due to some uncontrollable, or unknown factors. In other words, these are the risks that are caused by unknown factors. Selecting erroneous literature for the mapping study can be a threat. To avoid this, a concrete and thorough search process was executed. Appropriate inclusion and exclusion criteria have been used to ensure that the relevant literature was retrieved. Through half way of the search process, a cross evaluation of the articles was also performed, so as to ensure that the literature search was on the right track. However, there is a limitation that the search was only conducted in only two of the most popular databases; as performing search in every single technical database is practically not feasible.

One of the factors that may cause internal validity threat is, the way experimental subjects are treated [47]. During the experimental task, the researchers made sure that the subjects had no information that the topics presented to them were generated from two different topic models. This will ensure that the subjects will assign tags to all the topics with the same perspective. To eliminate bias in the experiment, it was ensured that all the human subjects participating in the experiment had no prior knowledge of the test cases, and the classification tags. The tasks that must be carried out during the experiment were explained to the subjects, at the beginning of the experiment. The subjects were also asked to complete the task in a single go. Having breaks in between gives the subjects time to think about the topics, and get accustomed to them. This would have resulted in some topics having better results than the others that were assigned tags in the beginning. This potential bias was eliminated.

4.3.2 External validity

External validity refers to the extent to which the results of the experiment can be generalized, and are valid out of the environmental setting in which it was conducted. External validity can be affected either by the design of the experiment, or the type of the subjects [47]. The systematic mapping study has been conducted based on a well-planned study plan. The search strings were also clearly formulated for the databases. Appropriate inclusion and exclusion criteria were applied to eliminate the irrelevant literature from the results. However, some of the articles were unavailable due to restricted access; which was a limitation for the mapping study. The results of our

mapping study can be generalized, as we have considered the literature from 2003; which is when the LDA came into existence. Hence researchers can gain an overview of the various metrics that have been used over the years to evaluate the topic model quality. Furthermore, we also provide an overview of the quality criteria context, in which the evaluation methods have been used, which is very useful for topic model optimization.

The entire design of our experiment was clearly mentioned along with the tools, and languages. The task was also clearly described so that it can be replicated in other environmental setting as well. The subjects for the experiment were also carefully chosen. As the dataset comprises of test cases of technical products (such as Firefox OS, Mozilla Firefox etc.,) not everyone can understand and assign tags to the topics. Hence the master's students from several technical disciplines such as software engineering, computer science, signal processing, tele communications were chosen as the subjects for our task.

4.3.3 Construct validity

Construct validity refers to the extent to which, the environment of the experiment affects the results. In other words, construct validity refers to the extent to which the experimental set up actually reflects the aims, and ideas of the researchers. The entire systematic mapping study was planned, and the study plan has been submitted to the research supervisor for review. The entire study plan was reviewed two times, and feedback has been provided by the supervisor suggesting changes, if any, so that the aims and intentions of the researchers were clearly reflected in the systematic mapping study plan.

This was done in case of experimental design also. All the experimental tasks and steps were designed in such a way that the objectives of the research were met. Even the files that were given to the subjects to perform the task contained the detailed instructions of the task along with an example, so that the subjects can clearly understand the task that they were supposed to perform.

4.3.4 Conclusion validity

Conclusion validity refers to how statistically significant the results are, so as to draw a valid conclusion. Wohlin et al. illustrated seven major issues in relation to conclusion validity, in [47]. These seven major issues have been discussed in relation to our thesis as follows:

4.3.4.1 Low statistical power:

The metrics that were calculated have been subjected to significance testing, so as to prove that the results are statistically significant, and are not by chance. The significance testing of the results has been described in detail, in chapter 7.

4.3.4.2 Violated assumptions of statistical tests:

In order to perform some of the significance test on the data, it is necessary that the data satisfies those assumptions. In our research we have performed three tests in total. They are the Shapiro-Wilk test, T-test, and Wilcoxon Signed rank test. Among these the Shapiro-Wilk test does not have any assumptions. T-test is a parametric test, and has assumptions about the data, which were satisfied the results obtained from the experiment. This was described in detail, in Chapter 7. Whereas, the Wilcoxon Signed Rank test is a non-parametric test and has no assumptions on the data. Thus the risk was eliminated.

4.3.4.3 Fishing and the error rate:

The research has been carried out entirely in tune with the research objectives, so as to achieve the aim of the research; thereby eliminating any potential biases, or the entire focus being directed towards a single outcome.

4.3.4.4 Reliability of measures:

It is very essential that the results are consistent for an experiment to be valid, i.e., the results must be reliable and consistent. Hence it is always advisable to use objective measures for the experiment, rather than the subjective measures, as there is a risk that subjective measures may be interpreted differently by different people. In our experiment, we have used time measures, precision, recall, and f-measure as they clearly meet the objectives of our research, and cannot be affected by personal opinions.

4.3.4.5 Reliability of treatment implementation:

While performing the experimental task it was ensured that all the subjects that participated in the experiment were treated the same, i.e., all the subjects were given the same instructions regarding the task, and the subjects were made to start the task, only when they have completely understood the instructions on what to do, and how to perform the task.

4.3.4.6 Random irrelevancies in experimental setting:

The risk of random irrelevancies has been reduced as much as possible, since the task was designed in such a way that it must be completed in a single go, without any distractions, or interruptions. However, it is practically impossible to control the random possibilities that are bound to happen. Hence, there is a little risk that certain unknown irrelevancies might have occurred during the execution of experimental task.

4.3.4.7 Random heterogeneity of subjects:

The subjects that are selected for an experiment must not be too heterogeneous as it may affect the results because of their difference; but they should not be too homogeneous also at the same time, as it may affect the external validity of the research. The subjects that were selected for the experiment were all post graduate students, but from several specializations such as computer science, software engineering, signal processing, and tele communications, so as to ensure that the sample population is general enough.

5 SYSTEMATIC MAPPING RESULTS

5.1 Search results

Once the search strings have been defined, the search was conducted in the selected databases to find the literature for the study. The entire search process is illustrated in Figure 10. The initial search results returned in a total of 2248 articles, in both of the selected databases. Then the inclusion and exclusion criteria have been applied to filter the results. The number of articles after applying inclusion and exclusion criteria were 1729 articles. The next step in the process includes filtration of the articles based on title. Both of the authors filtered the articles within their databases, based on the title. This further narrowed down the number of articles to a total of 193 articles. At this point, the authors evaluated the results of one another. A total of 35 articles were excluded during this cross evaluation. This left us with 158 articles. Once again, the authors were assigned the role of filtering articles based on the abstract. The abstracts of the articles were carefully studied and the articles that are most relevant to our research have been identified. This resulted in a total of 48 articles. Among these articles, 8 were not available as full text documents, and they were excluded. The authors performed a short iteration of forward snowballing approach and added 5 more articles for the study. In order to perform snowballing, the most related articles obtained from the database search were considered. Articles [19], [20] have been considered as the start set for snowballing. By using google scholar citation tracking, we have found a significant number of papers. By discarding the overlapping articles that we already found during the search process and by filtering them further by applying exclusion criteria, five articles have been selected. So, the final set of articles for the systematic mapping study consists of 45 articles.

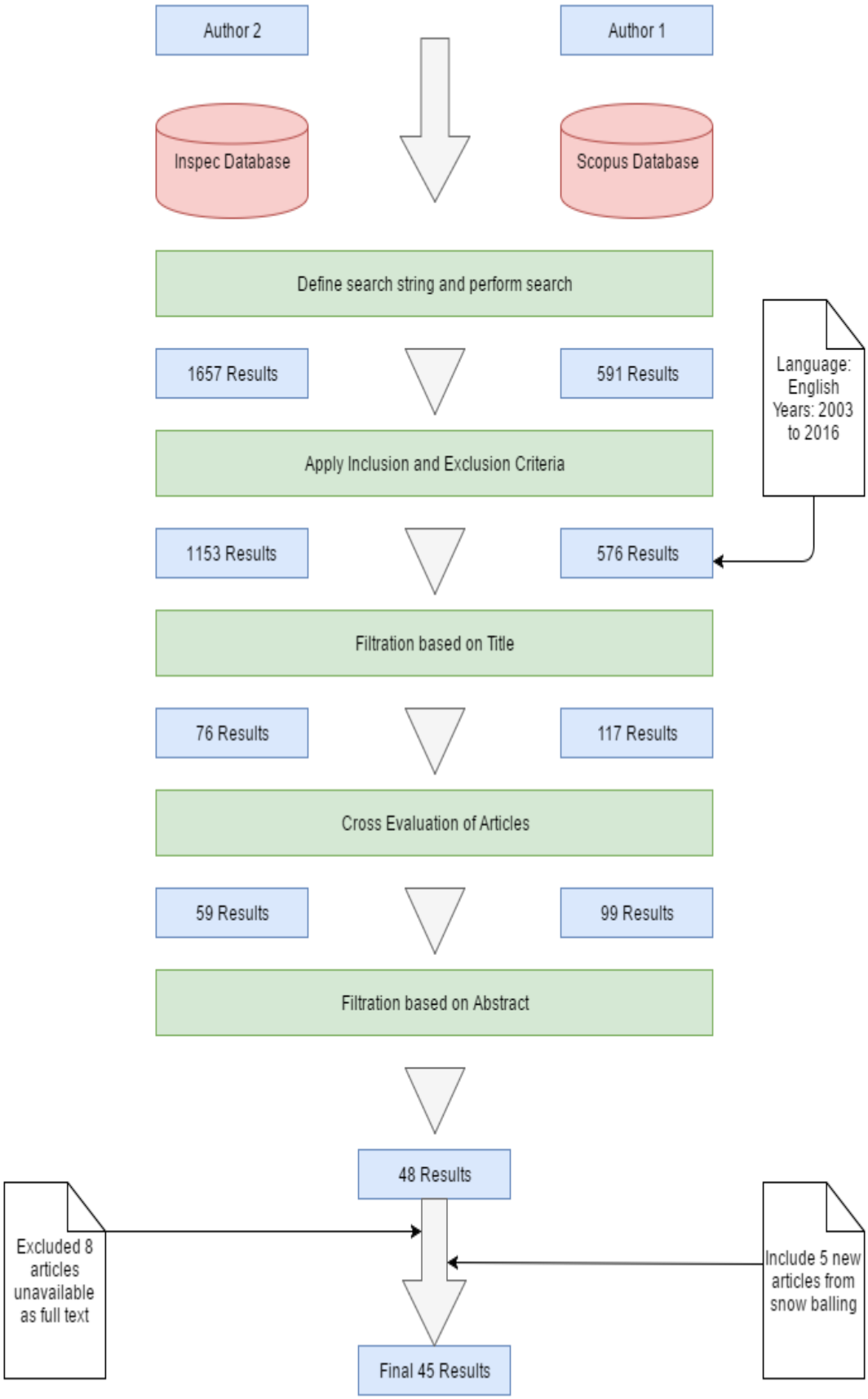

Figure 10: Search procedure for the mapping study

5.2 Mapping Study Results

The data has been extracted as per the above mentioned data extraction form (Section 4.1.4). This section represents the extracted data visually in the form of maps using graphs, tables etc., that provide an overview of the mapping study results, and the research done in the area.

5.2.1 Year and venue

Figure 11 shows the frequency of publications done over the years, starting from 2003. It also represents the venues types in which the research has been published. It is clear from the figure that, most of the research done on evaluating the topic models is published in the form of conference papers (32 conference papers). Whereas there are only 12 journal papers. It can also be observed that there is an increase in the number of publications from 2003 to 2015.

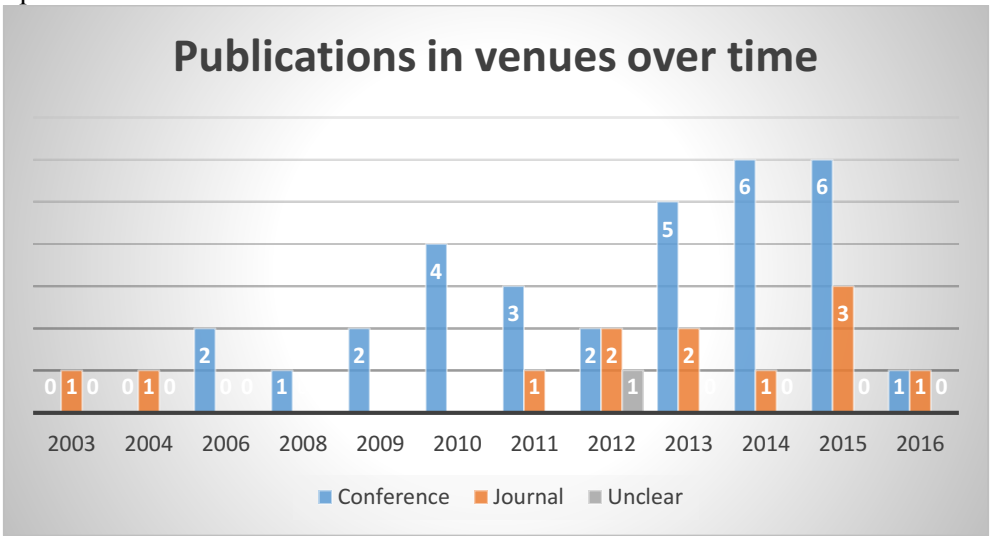

Figure 11: Publications in venues over time

5.2.2 Country of publication

This section provides an overview of the country-wise contribution in the selected set of research articles. The articles that were selected for the mapping study were the results of research conducted in 12 countries. The countries were determined based on where the research has been conducted. The contributions have been represented on a histogram (see Figure 12). A histogram uses intervals to group data based on ranges. These ranges are also referred to as intervals [51]. In our study, we have grouped the data according to three intervals ranging from 0 to 9.3, 9.3 to 18.6, and 18.6 to 27.9. The usage of these intervals can be better understood with the help of an example. Let us assume that a Country X has published 12 articles related to topic model evaluation. It will be assigned to the interval (9.3, 18.6]. having the interval boundaries between two numbers is to make sure that every value can be assigned to a particular interval [51]. It can be observed that, only one country has contributed to research with the number of articles between (18.6, 27.9]. The remaining 11 countries have contributed to the research with the number of articles ranging between [0, 9.3]. The individual contributions of the countries among the selected 45 articles are: The United States of America (USA): 22; China: 8; Australia: 3; Canada: 3; Germany: 2; The United Kingdom (UK): 2; Algeria: 1; France: 1; India: 1; Japan: 1; Romania: 1; South Korea: 1. The USA has been the major contributor in the research towards topic modelling with 22 articles, followed by china in the second place with 8 research articles. Countries such as Algeria, France, India, Japan, Romania, and South Korea have the least contribution among the selected articles, with one articles from each country.

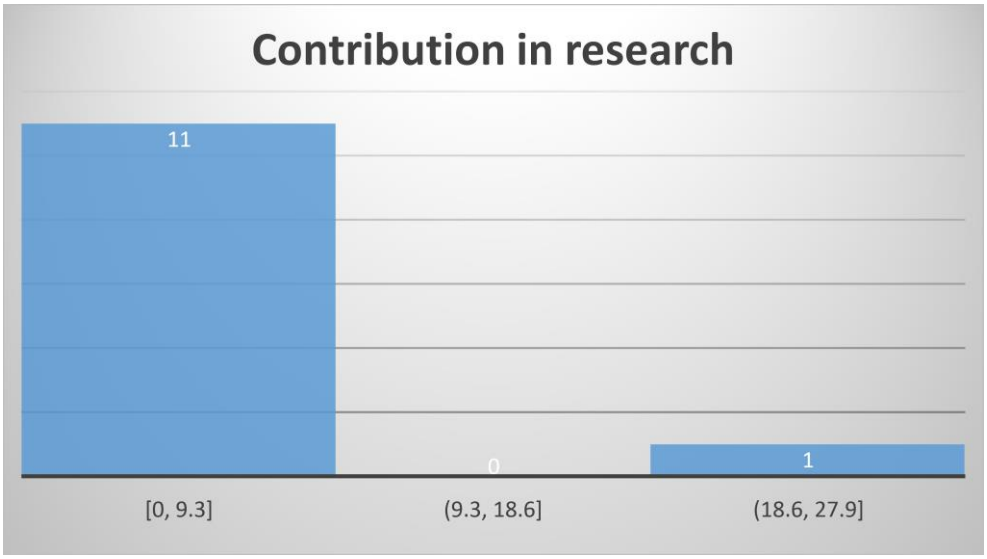

Figure 12: Country wise contributions in the research on topic models

5.2.3 Research areas/ Key words related to the research

In this section, we outline the various keywords/ research areas that were found in the selected set of articles, which are relate to our research. All the identified keywords have been assigned a frequency, which denotes the number of articles in which the exact or similar keyword has been found repeatedly. The word "Topic Modeling" was found to have the highest frequency, as it has been found in 32 different articles. Secondly, the word "Topic Model Evaluation" has been found to have the next highest frequency, as the exact or similar words have been found in 17 different articles. Other keywords such as "Information Retrieval", "Text Mining", "Clustering Techniques", "Hierarchical Topic Modeling", "Semantic Coherence", "Topic Quality", "Comparison of Topic Models", "Digital Libraries", "Text Visualization", and "Text Analysis" have also been found to have repetitive frequencies. Apart from these, several other distinctive, and non-repetitive keywords with single frequency count have also been found. All such words were represented as others, in Figure 13, which denotes the major keywords and their frequencies.

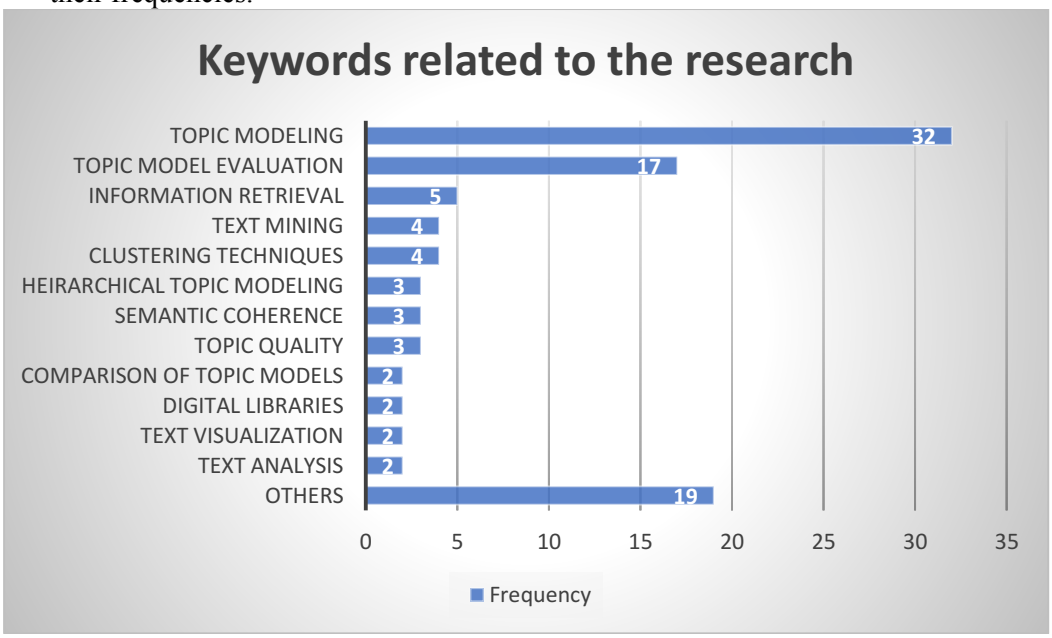

Figure 13: Majorly used keywords

5.2.4 Topic models

In most of the research done on topic models, it can be noticed that, LDA is the first choice topic model for most of the researchers. This is evident from the results of our mapping study (See Figure 14). From the figure, it can be observed that among the selected 45 articles, the researchers of 32 articles chose LDA as their topic model. With LDA being the most widely used topic model, there are several other topic models that have also been used such as Correlated Topic Model (CTM), Hierarchical Latent Dirichlet Allocation (HLDA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), variants of LDA, variants of HLDA, and some other topic models (See Appendix 10.4).

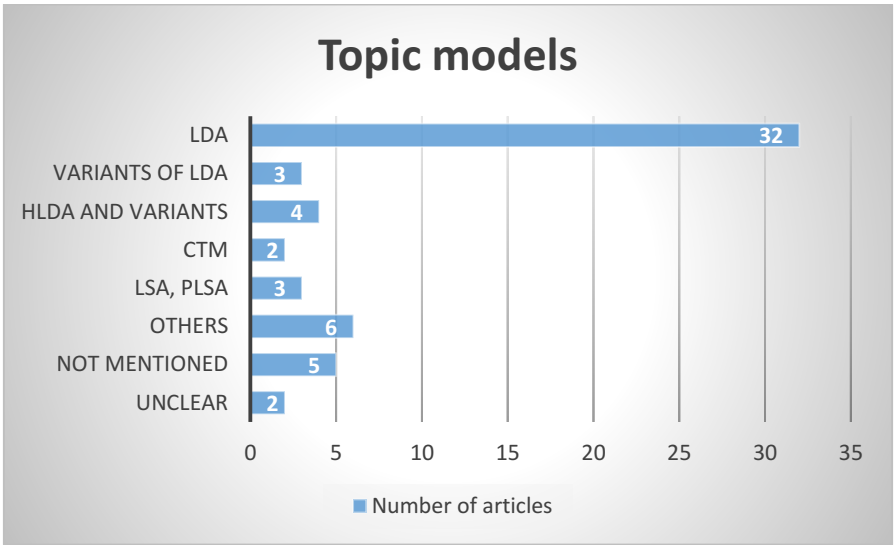

Figure 14: Topic models used in the selected articles

5.2.5 Datasets used

The datasets used in the selected set of articles have been categorized into three types namely text data, code, and music files. Since the primary objective of topic modeling revolves around text summarization, the dataset used by the researchers, in most of the experiments is text data. Authors mainly preferred handy data that is easily exportable and downloadable. Text data such as news articles, and articles from the scientific databases are most widely used. Data from Wikipedia, movie forums and Web forums are also used significantly as the dataset for the experiment. The datasets used in certain papers have not been specified clearly, but merely as text data. They were labelled as unspecified text datasets. Besides the structured text data, source code and music files have also been used in few papers. Table 6 provides a summary of the various types of data, the datasets that were used, and the articles in which they have been used.

Table 6: Datasets used in the articles

Type of data	Datasets used	Articles
TEXT DATA	News articles	[17], [20], [52]–[59]
	Wikipedia	[20], [60]–[62]
	Web forums	[63]–[67]
	Movies	[68]
	Medical reports	[69]
	Trade	[70], [71]
	Synthetic data	[19], [72]
	Scientific articles	[54], [59], [73]–[82]
	History	[70]
	Unspecified text	[6], [8], [17], [19], [83]–[87]

CODE	J-edit code	[88]
	Test case code	[11]
MUSIC FILES	Music songs	[89]

5.2.6 Research method

The most widely used research method to validate the results of the study was by conducting experiments. Among the selected 45 articles, almost 40 articles used experiments as their research method. In article [82], a survey was also conducted along with experimentation. Case study was another research method that has been used in [11], [71]. The research methods in [67], [90] have not been clearly mentioned (See Appendix 10.3).

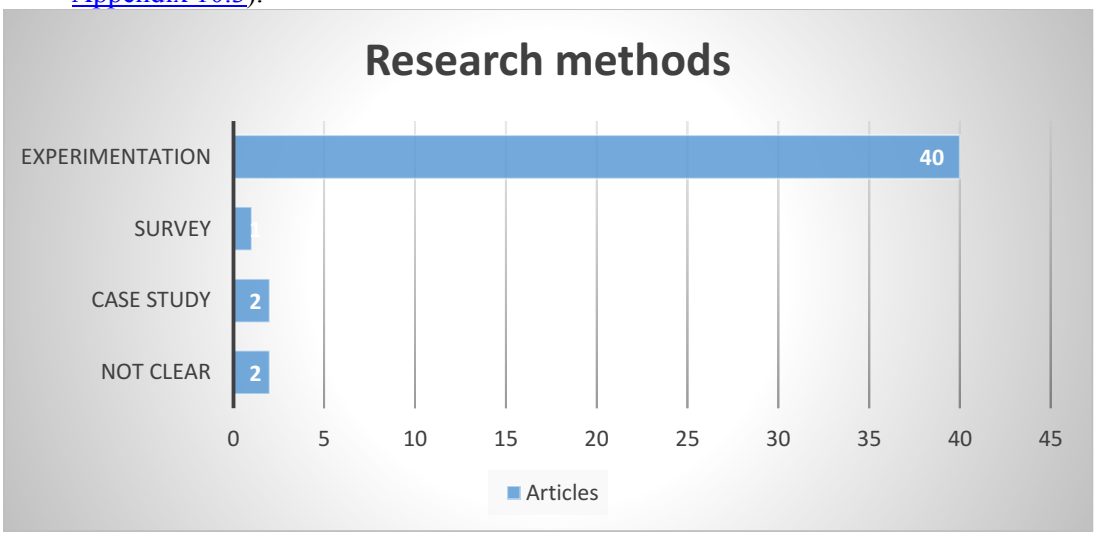

Figure 15: Research methods used in the articles

5.2.7 Quality criteria, and Evaluation methods

As the topic models are rapidly evolving, and their usage increasing day-by-day, more and more focus is being shifted towards the methods that exist to evaluate these topic models. Interpretability, generalizability and coherence are the three major criteria among several others, which are being used to evaluate the topic models. For almost every criterion, we found different evaluation methods that can be applied to the topic models. Studies show that Point-wise Mutual Information (PMI) is majorly used for evaluating the criteria such as interpretability, coherence, readability and topic consistency. The traditional human judgement is also used many papers to evaluate interpretability, coherence and understandability. Directed acyclic graph, which is an external evaluation technique, based on the graph approach was also found to be used, to evaluate coherence and reliability of the topics. Kullback Leibler divergence (KL-divergence), Annealed importance sampling are also used in few papers for evaluation.

Table 7: Quality criteria and Evaluation methods

Criteria	Method	Articles
Interpretability	PMI	[20], [53], [55], [68], [69], [81]
	Human judgement	[20], [61], [69], [70], [82], [84], [89]
	NPMI	[20], [52], [89]
	Task driven	[79]
	Annealed importance	[73], [80]
Generalizability	Clustering	[77], [86]
	Perplexity measure	[76], [85]
	Relevance computation	[70]
	Importance sampling	[19]

	Chib style estimation	[19]
	Topic weighing scheme	[6]
Coherence	PMI	[20], [53], [55], [68], [69], [81]
	NPMI	[20], [52], [89]
	Human judgement	[20], [61], [69], [70], [82], [84], [89]
	Directed acyclic graph	[52]
	Clustering	[77], [86]
	Relevance computation	[70]
	Tensor model	[60]
Clustering	Cosine distance	[56], [57], [64]
	K-center clustering	[56], [85]
Reliability	Directed acyclic graph	[52]
	NPMI	[20], [52], [89]
Relative performance of model	Annealed importance sampling	[73], [80]
	Empirical evaluation	[8], [63]
	Monte Carlo EM estimation	[54]
	KL-divergence calculation	[87]
	Predictive accuracy	[59]
	Saliency measure	[63]
	Link prediction and word prediction	[75]
Readability	PMI	[20], [53], [55], [68], [69], [81]
Topic consistency	PMI	[20], [53], [55], [68], [69], [81]
	Semantic evaluation	[66]
Understandability	Human judgement	[20], [61], [69], [70], [82], [84], [89]
Stability	KL-divergence calculation	[87]

5.2.8 Metrics

Through our study we have uncovered several metrics that have been used to evaluate the quality of the topic models. However, it is interesting that, there is a lot of diversity among the metrics used in the selected set of articles. Each paper has a different set of metrics based on its aim, and the quality criteria it chose to evaluate. Metrics such as precision, recall rate, topic coherence measures etc. have been found in more than one articles; thereby making them the frequently used metrics in our selected set of articles. Empirical comparisons have also been performed in some studies, where the selected topic models, or methods were compared against each other. Table 8 provides an overview of, the various metrics that have been found during our study, and also the corresponding research articles in which they were found.

Table 8: Metrics found during the study

Metrics	Articles
Precision and Recall	[8], [61], [85]
Coherence measures	[60], [84]
Topic weights	[89]
variance	[6]
Inverse document frequency (IDF)	[6]
Silhouette index	[83]
Topic distance measure	[56]
Distinctiveness	[74]

Half document perplexity	[72]
N-fold gross validation metric (CV Accuracy)	[72]
Spearman coefficient	[82]
KL-divergence	[82]
Rescaled dot product	[82]
Computational cost	[54]
Number of topics	[54]
Semantic measure	[70]
Co-occurrence measure	[55]
Empirical comparison	[11], [19]
F-measure	[8], [61]

5.2.9 *Selection of topic model quality criteria*

The systematic mapping study helped us in identifying the frequently evaluated quality criteria in case of topic models. Interpretability is one such frequently used quality criteria. As discussed in Chapter 1 of this document, topic models are being used for several tasks such as text classification, text analysis etc., and many more. As a result, the human users are directly exposed to the data that is generated as an output from these tasks [39]. As a result, it is essential that the output generated by the topic models must be very well human interpretable. After all, it is the sole purpose behind implementing topic models, and many other machine learning algorithms i.e., to simplify the tasks for humans, and to reduce the human effort in such tasks. This particular context has been targeted by other researchers as well, such as Lau et al. [39], and Chang et al. [20] etc. Hence, we chose interpretability to be one of the two quality criteria for our experiment. From the results of the mapping study, we have found that generalizability is another frequently used quality criteria to evaluate the topic models (see Section 5.2.7). In general, generalizability can be interpreted as the extent to which a particular topic can be generalized, among a given collection of documents [16]. In the context of topic modeling, generalizability can be seen as the grouping, or clustering of the documents based on their semantic structure [85]. So choosing both generalizability and interpretability as the use-cases/ quality criteria would mean that, one topic model would generate the topics focusing on interpretability (T_I); and the other would generate the topics that are more semantically clustered, or grouped together (T_G). This fact made the researchers curious, to compare and investigate the results of the topic models based on these quality criteria; so as to see which topic model produces the topics, that the subjects find more interpretable.

Apart from the fact that they are amongst the most frequently used quality criteria, other practical factors have also been considered while selecting the quality criteria for the experiment; such as the availability of source code, availability of information regarding the topic model implementation, ease of implementation, and topic model optimization as per the chosen quality criteria. This is because, let's say we chose to optimize the topic models for a quality criterion that does not have much information available, and that is too complex to implement. This would make it very hard for the researchers when issues are encountered during the experiment. As much information is not available, it will be hard for the researchers to solve these issues and find solutions. This may also result in the researchers not being able to submit the thesis on time. In order to avoid this, the above mentioned practical factors were also considered while choosing the quality criteria.

5.3 Answer to RQ1

As discussed in Section 3.1, the results from the systematic mapping study will be used to answer the research question 1, and also serve as a basis for the selection of evaluation methods, and metrics for the experiment.

RQ1) What are the different evaluation methods that can be used to evaluate the quality of topic models?

The research question RQ1 has been divided into three research questions RQ1.1, RQ1.2, and RQ1.3. The results obtained from the systematic mapping study have been used to answer the research questions as follows.

RQ1.1) What is the previous research that has been done in the context of evaluating the topic models?

The research on topic models has picked up pace since the inception of LDA in 2003. The amount of research conducted on topic models has been increasing year after year, with most of the research being published as conference papers (see Section 5.2.1, Figure 11). Amongst this, most of the research has been carried out in the US (Section 5.2.2, Figure 12). Topic modeling, Topic model evaluation, Information retrieval, Text mining, and Clustering techniques are the research areas that are connected to the research being done on topic models (Section 5.2.3, Figure 13). The most frequently used topic model was the Latent Dirichlet Allocation (LDA) topic model (Section 5.2.4, Figure 14). It was also found that textual data has been used in most of the research, as the dataset for the topic models (Section 5.2.5, Table 6). Experimentation was the most frequently used research method, in case of topic models (Section 5.2.6, Figure 15).

RQ1.2) What are the various types of evaluation methods that can be used to evaluate the performance of topic models?

Several evaluation methods were found during our study, for various quality criteria; with the dominant ones being PMI, NPMI, human judgement etc. A list of all the methods that were found during our study have been presented in Table 7, Section 5.2.7. In Chapter 6, of this document, we have described how the results of the mapping study were used to select the appropriate methods and metrics for the experiment, and how the selected methods were implemented in our experiment.

RQ1.3) What are the various quality criteria, and metrics that have been used in the evaluation of topic models?

Several quality criteria have been found in our study along with the methods, and metrics that can be used for those quality criteria. Results show that Generalizability, Interpretability, and coherence are the most commonly evaluated quality criteria in case of topic models (Section 5.2.7, Table 7). We have also uncovered certain metrics that have been used by the researchers to analyze the performance of topic models. They have been presented in Table 8, Section 5.2.8. Results indicate that precision, and recall are the most commonly used metrics along with coherence measures, empirical comparison, and f-measure in the next place.

6 EXPERIMENT RESULTS

This chapter presents the results obtained by conducting the experiment. The results from both training the topic models (Phase-A of the experiment), as well as from the task performed by the subjects have been presented (Phase-B of the experiment), see Figure 7 in Chapter 4. A detailed description of the experimental tasks was provided in Section 4.2, in Chapter 4.

6.1 Generating the Optimal Number of Topics

Based on the metrics obtained from the systematic mapping study, the two topic models have been implemented on the selected dataset to find the optimal number of topics. The original source code of theses topic models was obtained from [39], and has been adapted to the requirements of our task. The topic models have been implemented on the selected dataset with the following parameters. The minimum number of topics that are to be generated are 5, whereas the maximum number of topics must not exceed 255. After each step the number of topics is to be incremented with 10 (following the sequence 5, 15, 25, ….., 255). This process has been repeated for 10 iterations. The total time taken by each of the algorithm to generate the optimal number of topics for the respective quality criteria, and the optimal number of topics that have been obtained as output are described in this section.

6.1.1 Total run time of T_I and T_G

Both the topic models were executed to generate the optimal number of topics that would be ideal to be presented to the subjects. The total run times of both T_I and T_G have been presented in Table 9. The total run time of T_I was 46 hours, and 35 minutes; whereas the total run time of T_G was found to be 3 hours, and 30 minutes, which is surprisingly low when compared to that of the total run time of T_I.

Table 9: Time taken to generate the optimal number of topics

Time	T_I	T_G
Start time	2016-07-14, 6:59 PM	2016-07-17, 2:56 PM
End time	2016-07-16, 5:34 PM	2016-07-17, 6:26 PM
Total run time	46 hours, and 35 minutes	3 hours, and 30 minutes

6.1.2 Optimal number of topics produced by the algorithms

This section describes the optimal number of topics that have been generated by the two selected topic modelling algorithms.

6.1.2.1 Interpretability topic model (T_I)

T_I generates the topics, keeping in mind the topic interpretability as the primary quality criteria. T_I achieves this based on the PMI values for each topic. This has been explained in the Section 6.1.2.1.1.

6.1.2.1.1 Why PMI?

Inferring from the results of systematic mapping study, the most frequently used methods to evaluate the topic models are point wise mutual information (PMI), Normal point wise mutual information (NPMI), and human judgement (See Table 7). PMI is used in 6 articles and NPMI is used in 3 articles. NPMI is one of the variants of PMI. Though, PMI is traditional and NPMI is contemporary, we cannot call any of these measures as state of the art methods. Studies show that, the effectiveness of an evaluation method depends mostly on the task performed [20]. More rigorous research needs to be performed to show NPMI is always as efficient as PMI [91]. Also, in most of the articles, the researchers used human judgement to evaluate the interpretability of

the topic. But, as our goal is to optimize the topic model using automated methods, we have excluded human judgement. So we have chosen PMI over NPMI and human judgement, to optimize the topic model for interpretability. In our experiment, we are considering the human judgement as an interpretability task that will be used to evaluate the topic models.

6.1.2.1.2 Working of PMI

Point wise Mutual Information is a measure of correlation between two events, X and Y. Basically, it is a method which measures the information that is common in both x and y, i.e., reduction of uncertainty of y when we get to know about x [19]. The pointwise in the PMI indicates that the measure considers the specific events, or particular co-occurring events.

Example: x= rights, and y= reserved

PMI is calculated as

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$$

$$PMI(x,y) = log2 \frac{probability(x\ occurs\ with\ y\ under\ p(x,y))}{probability\ (x\ occurs\ with\ y\ under\ p(x)p(y))}$$

Or if we take above two words, word1=rights, word2=reserved,

$$PMI(rights, reserved) = log2 \frac{p(rights\&reserved)}{p(rights)p(reserved)}$$

Here, if we consider an article that is related to copyrights, the probability p(rights & reserved) is the probability that the words "rights" and "reserved" co-occur. If the given two words are independent of each other, then the probability that they occur together is given by p(rights)p(reserved). The ratio between the numerator and denominator is the measure of statistical dependence between two words. The logarithm of base 2 value in the formula helps us to convert the value into bits [92]. Hence, the value of the logarithm of the ratio is the amount of information about the word1 that we achieve, when we observe the word2 [93].

If pmi(x,y) >0, the two events are related. If pmi(x,y) = 0, two events are independent of each other. If pmi(x,y) <0, the two events are complementary distribution.

6.1.2.1.3 Optimal number of topics as per T_I

T_I generates the optimal number of topics by introducing an intruder word in each of the topics. Here, we used the methodology that was presented by Lau et al., in [39]. The proposed method takes the set of topics, including the intruder words. The Point-wise Mutual Information (PMI) score is computed for all the top N words of a topic, and combined with the features of the Support Vector Machine ranking algorithm (SVM_{rank}: joachims) to learn the intruder words. The model precision with which these intruder words have been identified, is automatically calculated by using an R-file. These results have been represented as a graph, in Figure 16. From the graph, it can be observed that the average model precision is high when the number of topics is 35. The average model precision at 35 topics was found out to be 0.7114. Whereas the model precision is very low when the number of topics is 5. The average model precision at 5 topics was found out to be 0.46. the average model precisions for all the number of topics has been presented in Table 10. This high model precision for 35 topics as per the interpretability topic model means that, these topics are the best interpretable, and that any intruder words can be detected easily, thereby providing better interpretability. Hence, T_I produced the topics with best model precision, when the number of topics are 35.

Table 10: Average model precision for interpretability topic model

Number of topics	Average model precision
5	0.46
15	0.6066
25	0.664
35	0.7114
45	0.6866
55	0.6745
65	0.6569
75	0.6640
85	0.6682
95	0.6852
105	0.6371
115	0.6260
125	0.648
135	0.6370
145	0.6365
155	0.6387
165	0.5951
175	0.6097
185	0.6178
195	0.6087
205	0.6039
215	0.6148
225	0.6057
235	0.5787
245	0.5918
255	0.5847

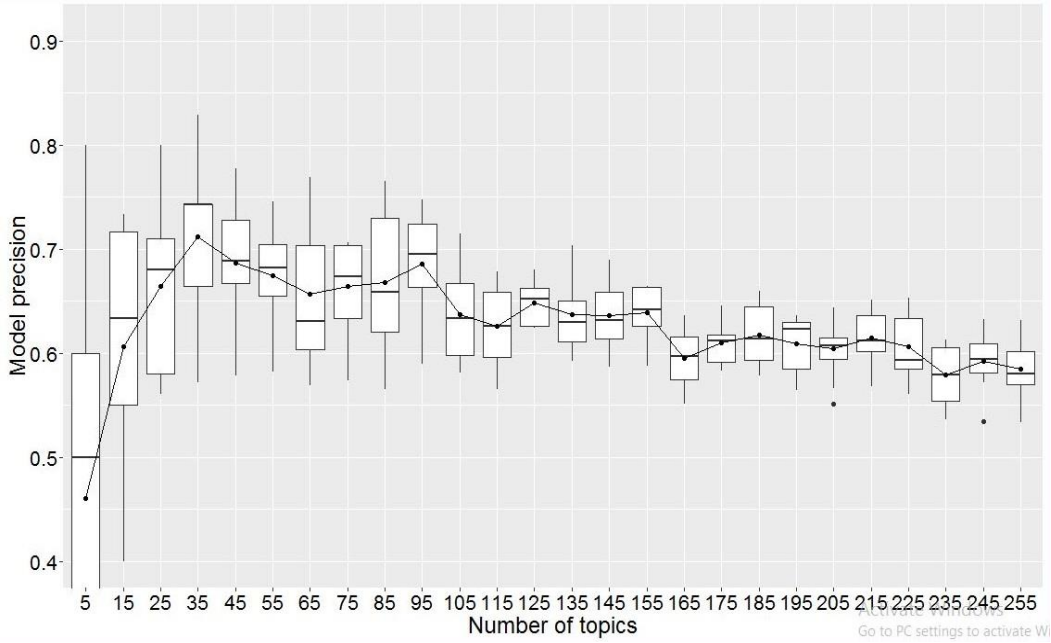

Figure 16: optimal number of topics by interpretability model

6.1.2.2 Generalizability topic model (T_G)

T_G optimizes the topics based on the perplexity of the topics, which has been explained in Section 6.1.2.2.1.

6.1.2.2.1 Why perplexity?

When considering generalizability as the quality criteria, we have found several evaluation methods. Out of these, perplexity measure was used in two articles, and some other measures were also mentioned in other articles. We have observed that in many cases, generalizability is evaluated by involving the log likelihood. Perplexity measure is also a method that uses log likelihood. Many researchers have qualitatively proved that the perplexity measure possesses high generalization ability on unseen data [17]. Thus making the topic model more generalizable. Hence, we have excluded the other methods and chosen perplexity measure to optimize T_G.

6.1.2.2.2 How it works?

As a word, perplexity can be defined as the state of confusion when something is difficult to understand. In text mining, perplexity is a way of evaluating the topic model. Perplexity can be considered as the measurement of, how well a probability distribution predicts a sample [17]. If a topic model has low perplexity, then it is considered to be more generalized topic model, compared to the one that has high perplexity.

Perplexity was proven to be one of the best methods to evaluate the performance of topic model [19]. Initially, the main idea to calculate the perplexity of a topic model is, to measure the log-likelihood of a held-out documents or a test set [20]. Basically it is achieved by splitting the dataset into two parts, one for training, and the other for testing. For the topic model (Latent Dirichlet Allocation), the test set is a collection of unseen documents (wd), and the topic model is described by the topic matrix (ϕ) and the hyper parameter (α), for topic distribution of documents.

$$l(w) = \log p(w|\phi, \alpha) = \sum_{d=1}^{M} logp(wd|\phi, \alpha)$$

The perplexity can be calculated by the formula:

$$Perplexity(test\ set\ w) = \exp\{-\frac{l(w)}{\sum_{d=1}^{m} Nd}\}$$

$$= \quad Perplexity(test\ set\ w) = \exp\{-\frac{\sum_{d=1}^{m} logp(wd)}{\sum_{d=1}^{m} Nd}\}$$

Here,
l(w)=log likelihood function
wd=represents the words in the document d
Nd=number of words in the document
M=total number of documents in the sample

The above mentioned measure is the decreasing function of log likelihood l(w) of unseen documents (wd). It can be interpreted as, the lower the perplexity the better the topic model.

6.1.2.2.3 Optimal number of topics as per T_G

T_G was also implemented under the same conditions, and same parameters; as that of T_I i.e. the minimum number of topics is 5, and the maximum number of topics is 255. The topics are to be incremented in steps of 10, the process must be repeated for 10 iterations. T_G optimizes the topics based on the perplexity measure. The average perplexities for all the number of topics has been presented in Table 11, and have been represented as a graph in Figure 17.

Table 11: Average perplexity for generalizability topic model

Number of topics	Average perplexity
5	277.9294
15	194.7412
25	166.3360
35	152.4651
45	144.2638
55	139.5902
65	134.9902
75	131.2559
85	128.4746
95	126.5746
105	124.6402
115	122.8339
125	121.5383
135	119.9034
145	119.5184
155	117.7392
165	117.5912
175	117.1875
185	116.5026
195	115.6133
205	115.5192
215	115.0194
225	114.5082
235	114.2016
245	113.4794
255	112.7463

The results show that the perplexity decreases as the number of topics increases. Thereby having the lowest perplexity value for 255 number of topics. But in reality, conducting the experimental task with 255 topics from the same topic model is not possible, since the resources are limited. So instead of this, the optimal number of topics were selected by calculating the differences in perplexities between each number of topics from 55 to 105. The lower number of topics, in an observed pair with the least perplexity difference, was considered as the optimal number of topics. The differences between the topic perplexities for the selected range of topics has been presented in Table 12. From Table 12, it can be observed that the difference in the perplexities is less when the number of topics is 85, and 95. Hence we considered 85 to be the optimal number of topics generated by T_G.

Table 12: Perplexity difference among the topics

Topics	Difference in perplexities
(55, 65)	4.5999
(65, 75)	3.7342
(75, 85)	2.7813
(85, 95)	1.8999
(95, 105)	1.9344

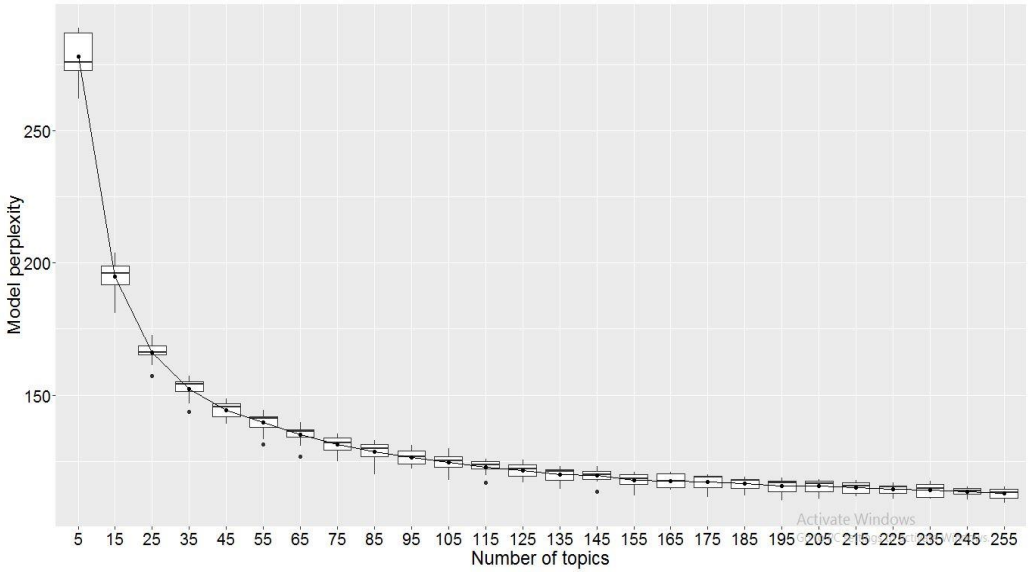

Figure 17: Optimal number of topics by generalizability model

6.2 Experimental task results

For the optimal number of topics obtained for both the topic models, in Section 6.1, the topics have been generated with the number of topics as 35, and 85. These generated topics from both the topic models have been given to the subjects as discussed in Section 4.2.1.2.2. Precision, Recall, and F-measure have been used as the metrics to analyze the performance of the subjects regarding the topics (Section 4.2.2.4).

6.2.1 Precision

As described in Section 4.2.2.4, precision helps us in assessing how well the subjects were able to assign the tags to the topics. It gives us an estimate of, to what extent were the identified tags relevant. The precision values for both of the selected topic models, for all the 20 subjects were as presented in Table 13.

Table 13: Precision results of the experiment

Subject	Interpretability model	Generalizability model
1	0.5413	0.184
2	0.536	0.3786
3	0.4433	0.4026
4	0.4933	0.3633
5	0.6206	0.4793
6	0.638	0.4693
7	0.62	0.398
8	0.602	0.4226
9	0.5806	0.4546
10	0.794	0.614
11	0.58	0.4393
12	0.5833	0.4406
13	0.6706	0.4186
14	0.4933	0.4653
15	0.3326	0.2309
16	0.3430	0.1932
17	0.2615	0.2586
18	0.5093	0.4796
19	0.4714	0.2148
20	0.3267	0.3391
Average	0.5220	0.3823

From the results, we can observe that T_I has better precision, and has outperformed T_G, in case of 95% of the subjects. The average of the precision values of all the subjects per topic model was found out to be 0.522 for T_I, whereas 0.3823 for T_G.

6.2.2 Recall

Recall is another metric that has been used to evaluate the performance of the topic models (Section 4.2.2.4). The recall metric gives us an estimate of the extent to which the relevant tags have been retrieved by the subjects, in the experiment. The recall values for both the topic models have been presented in Table 14.

Table 14: Recall values for the experiment

Subject	Interpretability model	Generalizability model
1	0.4946	0.2226
2	0.5393	0.336
3	0.3733	0.4226
4	0.4733	0.4793
5	0.4686	0.4486
6	0.4213	0.3326
7	0.506	0.4293
8	0.4173	0.4293
9	0.4786	0.4566
10	0.5346	0.4953
11	0.4713	0.5186
12	0.4266	0.4986
13	0.4466	0.408
14	0.2743	0.3631
15	0.5040	0.4219
16	0.3274	0.5765
17	0.3629	0.4677
18	0.2821	0.5093
19	0.4772	0.3612
20	0.4128	0.4427
Average	0.4346	0.4309

In case of recall, the results were a bit consistent with both the topic models. For 55% of the subjects, the recall values of T_I were found to be more than that of T_G. For the remaining 45% of the subjects, T_G outperformed T_I, which is significantly high compared to the 5% of the precision. The mean recall values for both the topics models were also found to be nearly equal, with T_I being slightly higher. The mean recall value for T_I was found to be 0.4346, whereas for T_G, it was found to be 0.4309.

6.2.3 F-measure

The third metric that we have used to evaluate the performance of the topic models, as discussed in Section 4.2.2.4 is, the f-measure. It is a harmonic mean of both precision and recall. The values of the f-measure for both the topic models, have been presented in Table 15.

Table 15: F-measure values of the experiment

Subject	Interpretability model	Generalizability model
1	0.5169	0.2014
2	0.5376	0.3560
3	0.4052	0.4120
4	0.4830	0.4133
5	0.5359	0.4634
6	0.5074	0.3892
7	0.5572	0.4130
8	0.4927	0.4259

9	0.5247	0.4556
10	0.6385	0.5482
11	0.5200	0.4756
12	0.4927	0.4672
13	0.5362	0.4132
14	0.3526	0.4067
15	0.4003	0.2984
16	0.3350	0.2893
17	0.3039	0.3323
18	0.3630	0.4935
19	0.4742	0.2687
20	0.3647	0.3837
Average	0.4670	0.3818

The results indicate that the values of f-measure for T_I were higher than the f-measure values of T_G, for 75% of the subjects. T_G outperformed T_I for the remaining 25% of the subjects. The average f-measure for T_I was 0.4670, and for T_G, it was 0.3818.

6.2.4 Time taken by the subjects

As mentioned in the experiment design (Section 4.2), the time taken by the subjects to complete the entire task has also been noted. The time taken by each of the subjects to complete the task have been tabulated in Table 16.

Table 16: Time taken by the subjects to complete the task

Subject	Time taken (in minutes)
1	85
2	90
3	75
4	80
5	65
6	80
7	90
8	70
9	55
10	100
11	49
12	70
13	60
14	75
15	39
16	72
17	60
18	110
19	55
20	96
Average	73.8

As mentioned in Section 4.2.1.2.2, in Chapter 4, the subjects had no time restriction to complete the task. The fastest time in which a subject was able to complete the given task was 39 minutes. Whereas, the longest time taken by a subject to complete the task was 110 minutes. If we take an average of the total time taken, for all the 20 subjects, on an average it took 73.8 minutes for each subject to complete the task.

Although the task completion time of the subjects was recorded, it was not that useful to us in the analysis. It would have been useful if the topics from both the topic models were given separately to the subjects, and times were recorded accordingly. Hence, a time-based analysis of the results was not possible.

7 ANALYSIS AND DISCUSSION

7.1 Analysis and Discussion

From the results of the experiment, it can be observed that the recall for both the topic models was almost same. But a number of false tags have been assigned to the topics generated by T_G. Hence, the overall precision was quite low for T_G, when compared to T_I. This was also reflected in the f-measure, as the f-measure of T_I was more than the f-measure of T_G. This can be clearly seen in Figure 23 (see Appendix 10.5). As the recall is almost same, and the precision is quite high for T_I, the precison is responsible for the sharp rise in the f-measure for T_I. The precision, recall, and f-measure graphs for both the topic models have been presented in Appendix 10.5. This can be clearly understood with an example.

- **Example:**

Table 17 shows two topics (X and Y), one from T_I, and the other from T_G. These sample topics are a subset of, the set of topics given to a subject. Both the topics were evaluated by the same subject. By observing the topic words and the actual tags for each topic, we can say that both the topics represent the same concept.

Table 17: Analysis example

	Topic_X from T_I	Topic_Y from T_G
Topic	Message, video, file, app, click, tap, media, shown, attached, attachment, content, send, attach, target, save	Message, tap, app, attachment, shown, video, send, content, page, successfully, mms, target, image, notification, create
Original topic tags	mms, message, streaming	Message, sms, mms
Tags assigned	mms, message	Message, mms, notification, camera
Precision	1	0.5
Recall	0.66	0.66
F-measure	0.79	0.56

In case of the Topic_X, the topic clearly explains the test cases related to media attachments, and sending a mms. The actual tags for Topic_X are mms, message and streaming. While performing the task, the subject assigned the tags mms, message. Since no false tags have been assigned to the topic, the precision for the Topic_X is 1.

But in case of Topic_Y, due to the presence of words image and notification, the subject is forced to think about the possibility that other tags may also exist for the topic, along with mms and message. As a result, camera and notification tags have also been assigned to the topic. But the actual tags are message,sms, mms. Due to the two false tags i.e., notification and camera, the precision is low for Topic_Y and calculated as 0.50.

In case of recall, both the topics achieved equall recall of 0.66. In both the cases, the top words clearly explain the minimum concept that the topics are related to i.e., “attaching the video and sending a mms”. We can understand that both the topics are equally sufficent to exaplain about the basic concept of test cases.

In case of f-measure, which is harmonic mean of precision and recall (see Section 4.2.2.4), the scores that we achieved for T_I and T_G are 0.79 and 0.56 respectively. This

shows that, the high precision of T_I, led to the increase in the f-measure of the same, when compared to the T_G.

7.2 Significance of the results

Hypothesis testing is one of the crucial parts in the research to verify the validity of the results. Statistical testing is used to evaluate two mutually exclusive samples of a specific population to determine the best statement that is supported by sample data. Several tests exist, that can test the results against the hypothesis [46].

The motivation behind conducting a statistical test is to find, or determine that we have gathered enough evidence to reject the null hypothesis of the experiment [47], [94]. The statistical hypothesis test in our experiment was conducted based on the data that we have obtained from the subjects. The main challenge in conducting the hypothesis testing is to select the appropriate test that best suits the data.

In statistics, the hypothesis tests can be categorized into two types, they are parametric and non-parametric tests. A parametric test makes assumptions about the parameters of the population distributions, from which the data is taken. Parametric tests involve specific probability distributions, and involve in the estimation of parameters "difference in means" of that distribution. Whereas in a non-parametric test, there are no assumptions, and they are distribution free [95].

So, based on the assumptions provided, one needs to select the test that supports their data and is best fit for his/her experiment.

7.2.1 Test for normality of data

The key assumption in a parametric test is that the sample data is normally distributed. Normal distributions are defined by two parameters namely mean (μ) and standard deviation (σ). The distribution in which the data is symmetrical around its mean is considered as normal distribution. Mean, median and mode of the normal distribution are equal [46]. In order to know if our data possessed normal distribution or not, we have performed the Shapiro-Wilk test [96].

$$W = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Here,

$\mathbf{x_i}$ are ordered random samples,

$\mathbf{a_i}$ are constants generated from the variances, and means of the sample.

When conducting shapiro-wilk test, we have assumed hypothesis as,

- **Null Hypothesis (H_0):** Data follows a normal distribution.
- **Alternate Hypothesis (H_1):** Data do not follow normal distribution.

We took the confidence interval as 5% i.e., alpha=0.05. While testing the hypothesis, there is a risk of committing two types of errors. If the null hypothesis is true, and we reject it, we make a type 1 error. If the alternate hypothesis is true, but is not supported then we make a type 2 error [47]. The alpha value plays a main role in determining the type 1 error, i.e. the probability of not accepting the null hypothesis when it is true, or in other words, the chances that we committed a type 1 error. The alpha value is the level of significance that researchers set for testing the hypothesis. The alpha value of 0.1 states that the researchers are willing to accept that, there is a 10% chance that they are wrong, when rejecting the null hypothesis. To lower this risk, the alpha should be decreased. But, a low alpha value (0.01) makes harder to detect the true difference. So, we have chosen the alpha value as 0.05 in our tests.

By Shapiro-Wilk test, the p value that we got is 0.6043. So, p>0.05. Hence, the null hypothesis can be accepted, and the data can be confirmed as a normal distribution and warranted parametric test.

We have also plotted QQPlot in addition to verify the normality results from Shapiro-Wilk test.

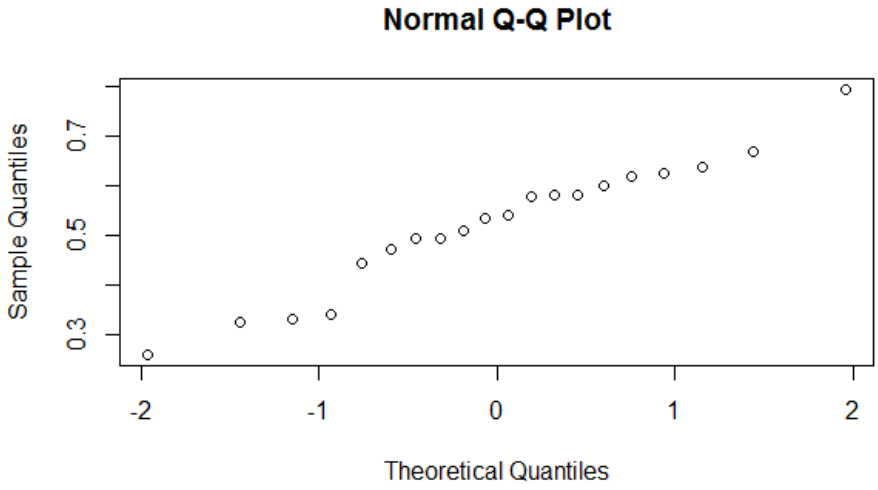

Figure 18: Normal distribution plot

The above plot in Figure 18 shows that, almost all points almost fall on the straight line and accept the sample as normal distribution.

7.2.2 Parametric test (T-test)

In parametric tests, we have T-tests, Z-tests, and Annova tests [47].

- **Annova:** The analysis of variance is used to know to find any statistical differences in means of three or more independent sample groups.
- **Z-test:** It is a parametric test that considers the difference between mean of the variable in a sample and mean of the variable in larger population.
- **T-test:** T-test used to examine the significance of the difference between sample mean and population mean in smaller population.

Annova test is the best method, when we want to know the differences of the means, in three or more sample groups. But here in our data, we only have two independent groups. So, Annova test cannot be used here.

Z-test and T-test are the most widely used methods in statistics. The main rules for using the z-test are:

a) When the sample is large N>30.
b) When the population standard deviation is known.

The sample size of our experiment is only 20, and the standard deviation of the population is unknown. Hence, we can eliminate the z-test, and use the t-test for testing our hypothesis.

T-test is calculated by the formula:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{S_1^2}{N1} + \frac{S_2^2}{N2}}}$$

Here,
$\bar{x}_1$ = Mean of first sample
$\bar{x}_2$ = Mean of second sample
s_1^2 = standard deviation of first sample
s_2^2 =standard deviation of second sample
$N1$=Number of elements in first sample
$N2$=Number of elements in second sample

The numerator in the formula is the difference between the means of two samples and denominator is the measure of variability, or the standard error of difference. T-value is looked up in the significance table, to test whether the ratio is large enough to confirm the difference between the groups, and that it is not by chance. To test the

significance, we need to set the alpha to a particular point, and we compare the value in the table, to the alpha value. If the p value is smaller than alpha, we can reject the null hypothesis and accept that there is a significant difference between the two groups.

The results of the T-test for the metrics precision, recall, and f-measure (see Table 18); between the T_I and T_G, in our experiment are as discussed below:

- T-test value for precision between T_I and T_G is 0.000001824. For 5% confidence interval, the P value is clearly less than 0.05, so we can reject null hypothesis.
- T-test value for Recall between T_I and T_G is 0.89184. For 5% confidence interval, the p-value is greater than 0.05. Hence if we cannot reject null hypothesis, if we consider recall as the main measure.
- T-test value for F-measure between T_I and T_G is 0.00277. For 5% confidence interval, the P value is less than 0.05. Hence, we can reject null hypothesis and accept alternate hypothesis stating that there is significant difference between T_I and T_G when they are used upon interpretability tasks.

7.2.3 Non-parametric test (Wilcoxon Signed-Rank Test)

Non-parametric tests are less powerful compared to the parametric tests when they are used on our experimental results. As non-parametric tests are valid for both non-normalized distribution, and normalized distribution data, we are interested to test the results on one of the non-parametric test that is best supported by our data. There are many non-parametric tests present in statistics. The best alternative test for the t-test (paired) in non-parametric tests is Wilcoxon Signed–Rank Test [47].

- **The Wilcoxon Signed–Rank Test:** This non-parametric test is used to compare two independent samples without making assumptions related to normal distribution. Some main requirements for using this test are:

a) Data should be continuous and distinguishable.
b) The two groups are independent samples.
c) The scale supported are ordinal, nominal, interval or ratio.
d) The approximation to the normal distribution is best when size of two sample groups (Na and Nb) are equal to or greater than 10.

Wilcoxon Signed rank test is calculated by:

$$W = \sum_{i=1}^{N_r} [sgn(x_{2,i} - x_{1,i}).R_i]$$

$$Z = \frac{(W - \mu_w)}{\sigma_w}$$

Here,
W= wilcoxon value
Sgn = sign function
Rank denoted by R_i
Z= Z approximation value
μ_w= Mean of W
σ_w = Standard deviation of W

Here, we use Z-value to take the corresponding value for p from the critical table.

If value of p taken from $Z_{critical}$ table is less than 0.05, then we reject the null hypothesis.

In our experiment,

- The Z-value for the precision of T_I and T_G is -3.48. The corresponding p value is 0.00012. The result is significant at $p<0.05$.
- The Z-value for the Recall of T_I and T_G is 0.17583. The corresponding p value is 0.8493. The result is not significant as $p<0.05$.

- The Z-value for the F-measure of T_I and T_G is -2.8. The corresponding p value is 0.00512. The result is clearly less than 0.005.

Table 18: Significance test results

Significance test	Precision	Recall	F-measure
T-test	0.000001824	0.89184	0.00277
Wilcoxon signed rank test	0.00012	0.8493	0.00512

There is a significant difference between the results of both the topic models. The results show that T_I outperformed T_G.

7.3 Answer to RQ2

RQ2) Among the two topic models that are optimized for different quality criteria, which topic model provides better results, when used for an interpretability task?

Based on the results from the experiment, it can be seen that T_I clearly outperformed T_G; in terms of precision, and f-measure. The average values of precision, recall, and f-measure per topic model are shown in Table 19.

Table 19: Average metrics values of the topic models

Metric	Interpretability	Generalizability
Precision	0.5220	0.3823
Recall	0.4346	0.4309
F-measure	0.4670	0.3818

From Table 19, it can be observed that T_I clearly outperforms T_G. Although the recall values are almost same for both the topic models, the recall of T_I exceeds the recall of T_G by 0.0037. Appropriate significant tests have been conducted to validate the results in Section 7.2. The results from the significance tests (see Table 18) suggest that, the results of T_I were significantly high than T_G, for precision, and f-measure; thereby making our assumption true that a topic model optimized for interpretability performs well for an interpretability task, and allowing us to reject the null hypothesis. However, it was found that the results were not that significant in terms of recall. So, the null hypothesis cannot be rejected in terms of recall. Hence, based on the results from the experiment, and the significance tests, we can say that among the two topic models that are optimized for generalizability (T_G) and interpretability (T_I), the topic model that is optimized for interpretability (T_I) provides better results, when used for an interpretability task.

8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

The main aim of the thesis was to outline various quality criteria, evaluation methods, and metrics that can be used to evaluate topic models; and to generate optimized topic models for the quality criteria found, and evaluaate the performance of topic models in case of test case prioritization. The methods and metrics were then used to compare the performance of those topic models. Our assumpton was that, to achieve better prioritization results, it is important to optimize the topic model for the particular task. In this thesis, we tested this assumption by conducting an experiment. In order to achieve this, we have conducted a systematic mapping study, which was followed by an experiment. The systematic mapping study was used to identify the various quality criteria, evaluation methods, and metrics that can be used to evaluate topic models. Apart from these we have also identified the most frequently used topic model, dataset, and provided an outline on the research that has been conducted on topic models, all of which has been decribed in detail in Chapter 5, answering our RQ1.

We have chosen generalizability, and interpretability to be the quality criteria that we would like to optimize the topic models with. The selection of topic models has been described in detail, in Section 5.2.9. The evaluation methods that have been used to generate the topic models were PMI for interpretability topic model (T_I), and perplexity for generalizability topic model (T_G); whose working has been clearly explained in Section 6.1.2. We have selected the metrics precision, recall, and f-measure to compare the performance of these two topic models, which were discussed in Section 4.2.2.4.

Through this study we have provided the results of a systematic mapping study that provides an overview of the amount of research that has been done on evaluating topic models, in the form of number of research publications per year; key areas related to the research; most frequently used topic models, datasets, research methods; various quality criteria, evaluation methods, and metrics that can be used to evauate the topic models (see Chapter 5). We believe that this is the first systematic mapping study in this research area, to the best of our knowledge. We have also compared the performance of two topic models that were optimized for two different quality criteria (use-cases); using test cases as the dataset.

When compared to the other research that has been done using test cases as the dataset, in [11] Thomas et al. propose a new technique that uses topic models for test case prioritization in the context of black box testing, and compared its performance with the existing test case prioritization techniques. Where as in [97], Unterkalmsteiner et al. proposes a new test case selection approach to guide the domain experts, by exploiting the probabilistic nature of topic models. Our research differs from the research done in [11], [97], in such a way that, we did not propose any new techniques. But rather we have optimized two topic models based on the quality criteria (Generalizability and Interpretability) that we identified from the mapping study, and compared the performance of these topic models using the metrics (Precision, Recall, and F-measure) that were also identified from the mapping study.

Our results showed that T_I exhibits better performance than T_G, for precision (see Section 7.3, Chapter 7). Where as in case of recall, T_G performed almost same as T_I. This high domination in precision, caused T_I to possess high f-measure values, when compared to the T_G; although the recall was almost same for both topic models. Hence, T_I performed better. Apart from the selected metrics, we have also considered the total run time of the topic models i.e., the total time taken by the topic model to generate the optimal number of topics. Table 8 shows the total run time taken by both the topic models. The total run time of T_I is 46 hours, and 35 minutes (2795 minutes). Where as,

the total run time of T_G is 3 hours, and 30 minutes (210 minutes). It can be observed that there is a significant difference in the total run time of both the topic models.

As a result, we would like to conclude saying that, there are certain factors to be considered when choosing a topic model (amongst T_I and T_G) for an interpretability task. If the main goal of the task at hand is, generating topics that are more precise and interpretable, the user should opt for T_I. Furthermore it is better to choose T_I, when time is not a limiting factor; since it has a high total run time. If the main goal of the task at hand is, to generate topics that are partially interpretable and in less time, then user should opt for T_G. This is mainly because of two reasons. The recall of two topic models is almost same. This means that the topics from T_G are sufficent to provide a basic understanding of the underlying concepts of the test cases or any other documents. Furthermore, T_G has a less total run time; which makes it all the more suitable for time critical tasks, where topic quality isn't of high priority.

8.2 What do our results mean for test case prioritization?

From the results of the experiment (see Table 19), it is clear that T_I has better precision, recall, and f-measure than T_G, thereby making our assumption true that task specific optimization of topic models will yield better test case prioritization results. Apart from this, the implications of our research can also be understood from the metrics that we have chosen. The metric precision can be understood as "the extent to which the chosen tags were correct". Whereas, the metric recall can be understood as "the extent to which the correct tags were chosen". There is a significant difference between the meaning of the metrics. This can be better understood with an example. Let us consider two scenarios that involve testing a single feature, and testing an entire system. When the entire focus is shifted towards testing a single feature, we want to test it as rigorously as possible. The tester therefore tries to implement as many test cases as he can get his hands on, which are related to that feature. In terms of the metrics, the tester can choose the topics with better recall. Since recall is the extent of choosing the correct tags, the better the recall, the confident the tester can be in testing the particular feature. In case of the other scenario, i.e., testing an entire system; rigorously testing each and every feature of the system is not feasible at once. Hence, the testers need to prioritize the test cases so that the most crucial test cases are covered. In this case, they can opt for the topics with better precision. Since precision is the extent of the chosen tags being correct, the more the precision, the better is the chance of choosing the relevant crucial test cases.

It can be a bit tricky to understand the correlation between these two metrics, in the context of test case prioritization as they seem to be so relevant to each other, yet they are so far away in what they mean.

8.3 Future work

In our research, we have optimized our topic model (LDA) based on two different use-cases, and observed the difference when they are used for a same particular task. An interesting future work would be to select two different topic models (for example, LDA and HLDA) and optimize them based on a single quality criterion, and study their performances on a particular task. Furthermore, the main focus of our thesis has been on how the two topic models vary in their performance, in terms of generalizability and interpretability. That is, the focus has been more on how the results vary, rather than how good they actually are. So an interesting way would be to investigate the quality of the generated topics.

Also due to limited resources, only 20 subjects were considered for the task. The same task can be extended to a much bigger sample population, to see if there is any difference between the selected metrics.

Since the topics generated from both the topic models have been randomly provided to the subjects in the same document, in equal proportions, we weren't able to perform a time based analysis on the topic models. So, a time based analysis can also be

performed by having two groups of subjects, and having one group to perform the task by giving the topics from one topic model, and vice versa. This way the researchers would be able to know the time taken by the topics to understand the topics by a topic model, and find out which topic model gives topics with a better understandability in less time.

For the dataset, we have used test cases to create topics using LDA topic model. We have observed that topic model produced promising results, when used for test case prioritization. In future, we would like to see the significant use of topic modeling algorithms, for the task of requirement prioritization, where, the linguistic data from the requirements documents that include SRS (description for software systems, functional, non-functional requirements), and case studies from previous similar projects are used.

9 REFERENCES

[1] D. M. Blei, "Probabilistic Topic Models," *Commun ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[3] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[4] Christine Doig, "PyGotham 2015. Introduction to Topic Modeling in Python." [Online]. Available: http://chdoig.github.io/pygotham-topic-modeling/#/2/1. [Accessed: 19-Mar-2016].

[5] A. Ozgür, "Supervised and unsupervised machine learning techniques for text document categorization," Bogaziçi University, 2004.

[6] S. Lee, J. Kim, and S.-H. Myaeng, "An extension of topic models for text classification: A term weighting approach," in *Big Data and Smart Computing (BigComp), 2015 International Conference on*, 2015, pp. 217–224.

[7] V. Krishnan, "Short comings of latent models in supervised settings," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 625–626.

[8] C. Schnober and I. Gurevych, "Combining Topic Models for Corpus Exploration: Applying LDA for Complex Corpus Research Tasks in a Digital Humanities Project," in *Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications*, 2015, pp. 11–20.

[9] L. C. Briand, "Novel applications of machine learning in software testing," in *Quality Software, 2008. QSIC'08. The Eighth International Conference on*, 2008, pp. 3–10.

[10] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, "Software traceability with topic modeling," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, 2010, pp. 95–104.

[11] S. W. Thomas, H. Hemmati, A. E. Hassan, and D. Blostein, "Static test case prioritization using topic models," *Empir. Softw. Eng.*, vol. 19, no. 1, pp. 182–212, 2014.

[12] H. Hemmati, Z. Fang, and M. V. Mantyla, "Prioritizing Manual Test Cases in Traditional and Rapid Release Environments," in *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 2015, pp. 1–10.

[13] M. Unterkalmsteiner, R. Feldt, and T. Gorschek, "Supporting Experts in Test Case Selection with Topic Models," 2015.

[14] J. Woo, "Agile Test Methodology for B2C/B2B Interoperability," *Dep. Ind. Manag. Eng. Pohang Univ. Sci. Technol.*, 2007.

[15] Z. C. Lipton, D. C. Kale, C. Elkan, R. Wetzell, S. Vikram, J. McAuley, R. C. Wetzell, Z. Ji, B. Narayaswamy, C.-I. Wang, and others, "The Mythos of Model Interpretability," *IEEE Spectr.*, 2016.

[16] "ISO 10075-3:2004(en), Ergonomic principles related to mental workload — Part 3: Principles and requirements concerning methods for measuring and assessing mental workload." [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:10075:-3:ed-1:v1:en:term:3.6. [Accessed: 01-Sep-2016].

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J Mach Learn Res*, vol. 3, pp. 993–1022, Mar. 2003.

[18] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia, "How to Effectively Use Topic Models for Software Engineering Tasks? An Approach Based on Genetic Algorithms," in *Proceedings of the 2013 International Conference on Software Engineering*, Piscataway, NJ, USA, 2013, pp. 522–531.

[19] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation Methods for Topic Models," in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, pp. 1105–1112.

[20] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in neural information processing systems*, 2009, pp. 288–296.
[21] K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," *IEEE Trans. Softw. Eng.*, vol. 21, no. 2, pp. 126–137, Feb. 1995.
[22] M. Noorian, E. Bagheri, and W. Du, "Machine Learning-based Software Testing: Towards a Classification Framework.," in *SEKE*, 2011, pp. 225–229.
[23] M. Evett, T. Khoshgoftar, P.-D. Chien, and E. Allen, "GP-based software quality prediction," in *Proceedings of the Third Annual Conference Genetic Programming, volume*, 1998, pp. 60–65.
[24] K. Ganesan, T. M. Khoshgoftaar, and E. B. Allen, "Case-based software quality prediction," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 10, no. 2, pp. 139–152, 2000.
[25] P. Guo and M. R. Lyu, "Software quality prediction using mixture models with EM algorithm," in *Quality Software, 2000. Proceedings. First Asia-Pacific Conference on*, 2000, pp. 69–78.
[26] J. J. Dolado, "A validation of the component-based method for software size estimation," *IEEE Trans. Softw. Eng.*, vol. 26, no. 10, pp. 1006–1021, 2000.
[27] N. E. Fenton and M. Neil, "A critique of software defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 25, no. 5, pp. 675–689, 1999.
[28] T. Dohi, Y. Nishio, and S. Osaki, "Optimal software release scheduling based on artificial neural networks," *Ann. Softw. Eng.*, vol. 8, no. 1–4, pp. 167–185, 1999.
[29] D. Zhang and J. J. P. Tsai, "Machine Learning and Software Engineering," *Softw. Qual. J.*, vol. 11, no. 2, pp. 87–119.
[30] D. Binkley, D. Heinz, D. Lawrie, and J. Overfelt, "Understanding LDA in Source Code Analysis," in *Proceedings of the 22Nd International Conference on Program Comprehension*, New York, NY, USA, 2014, pp. 26–36.
[31] S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein, "Validating the Use of Topic Models for Software Evolution," in *2010 10th IEEE Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2010, pp. 55–64.
[32] A. D. Lucia, M. D. Penta, R. Oliveto, A. Panichella, and S. Panichella, "Using IR methods for labeling source code artifacts: Is it worthwhile?," in *2012 IEEE 20th International Conference on Program Comprehension (ICPC)*, 2012, pp. 193–202.
[33] S. K. Lukins, N. A. Kraft, and L. H. Etzkorn, "Source Code Retrieval for Bug Localization Using Latent Dirichlet Allocation," in *2008 15th Working Conference on Reverse Engineering*, 2008, pp. 155–164.
[34] A. T. Nguyen, T. T. Nguyen, J. Al-Kofahi, H. V. Nguyen, and T. N. Nguyen, "A topic-based approach for narrowing the search space of buggy files from a bug report," in *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2011, pp. 263–272.
[35] E. Linstead, P. Rigor, S. Bajracharya, C. Lopes, and P. Baldi, "Mining Eclipse Developer Contributions via Author-Topic Models," in *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)*, 2007, pp. 30–30.
[36] C. Catal, "The Ten Best Practices for Test Case Prioritization," in *Information and Software Technologies*, T. Skersys, R. Butleris, and R. Butkiene, Eds. Springer Berlin Heidelberg, 2012, pp. 452–459.
[37] P. R. Srivastava, "Test case prioritization," *J. Theor. Appl. Inf. Technol.*, vol. 4, no. 3, pp. 178–181, 2008.
[38] R. K. Saha, L. Zhang, S. Khurshid, and D. E. Perry, "An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes," in *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, Piscataway, NJ, USA, 2015, pp. 268–279.
[39] J. H. Lau, D. Newman, and T. Baldwin, "Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality.," in *EACL*, 2014, pp. 530–539.
[40] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, "Using mapping studies in software engineering," in *Proceedings of PPIG*, 2008, vol. 8, pp. 195–204.

[41] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, 2007.
[42] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering*, 2008, vol. 17.
[43] "Inforum 2002 - papers." [Online]. Available: http://www.inforum.cz/archiv/inforum2002/english/prednaska58.htm. [Accessed: 26-Mar-2016].
[44] K. I. K. Charles W. Knisely, *Engineering communication*. Cengage Learning, 2014.
[45] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*, Springer, 2008, pp. 285–311.
[46] N. Juristo and A. M. Moreno, *Basics of software engineering experimentation*. Springer Science & Business Media, 2013.
[47] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
[48] J. Huang, "Performance Measures of Machine Learning," University of Western Ontario, Ont., Canada, Canada, 2006.
[49] "Environments — MozTrap 1.4 documentation." [Online]. Available: http://moztrap.readthedocs.io/en/latest/userguide/model/environments.html. [Accessed: 06-Jul-2016].
[50] "Test Cases, Suites and Tags — MozTrap 1.4 documentation." [Online]. Available: http://moztrap.readthedocs.io/en/latest/userguide/model/library.html. [Accessed: 06-Jul-2016].
[51] "Histograms." [Online]. Available: http://onlinestatbook.com/2/graphing_distributions/histograms.html. [Accessed: 07-Sep-2016].
[52] H. Davoudi and A. An, "Ontology-Based Topic Labeling and Quality Prediction," in *Foundations of Intelligent Systems*, F. Esposito, O. Pivert, M.-S. Hacid, Z. W. Rás, and S. Ferilli, Eds. Springer International Publishing, 2015, pp. 171–179.
[53] Y. Ding and S. Yan, "Topic Optimization Method Based on Pointwise Mutual Information," in *Neural Information Processing*, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds. Springer International Publishing, 2015, pp. 148–155.
[54] K. L. Caballero, J. Barajas, and R. Akella, "The Generalized Dirichlet Distribution in Enhanced Topic Detection," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2012, pp. 773–782.
[55] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic Evaluation of Topic Coherence," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2010, pp. 100–108.
[56] Y. Wu, Y. Ding, X. Wang, and J. Xu, "A comparative study of topic models for topic clustering of Chinese web news," in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, 2010, vol. 5, pp. 236–240.
[57] K. Mikawa, T. Ishida, and M. Goto, "A proposal of extended cosine measure for distance metric learning in text classification," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 1741–1746.
[58] A. Brahmi, A. Ech-Cherif, and A. Benyettou, "Arabic texts analysis for topic modeling evaluation," *Inf. Retr.*, vol. 15, no. 1, pp. 33–53, Jun. 2011.
[59] H. M. Wallach, "Topic Modeling: Beyond Bag-of-words," in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, pp. 977–984.
[60] S. Spagnola and C. Lagoze, "Word Order Matters: Measuring Topic Coherence with Lexical Argument Structure," in *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, New York, NY, USA, 2011, pp. 21–24.

[61] K. Seshadri, S. Mercy Shalinie, and C. Kollengode, "Design and evaluation of a parallel algorithm for inferring topic hierarchies," *Inf. Process. Manag.*, vol. 51, no. 5, pp. 662–676, Sep. 2015.

[62] H. Chan and L. Akoglu, "External Evaluation of Topic Models: A Graph Mining Approach," in *2013 IEEE 13th International Conference on Data Mining (ICDM)*, 2013, pp. 973–978.

[63] W. Wang, H. Xu, W. Yang, and X. Huang, "Constrained-hLDA for Topic Discovery in Chinese Microblogs," in *Advances in Knowledge Discovery and Data Mining*, V. S. Tseng, T. B. Ho, Z.-H. Zhou, A. L. P. Chen, and H.-Y. Kao, Eds. Springer International Publishing, 2014, pp. 608–619.

[64] L. Shan, C. Sun, L. Lin, M. Liu, X. Wang, and B. Liu, "Evaluating tag quality for blogger modelling via topic models," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 1770–1776.

[65] D. Kelly, F. Diaz, N. J. Belkin, and J. Allan, "A User-Centered Approach to Evaluating Topic Models," in *Advances in Information Retrieval*, S. McDonald and J. Tait, Eds. Springer Berlin Heidelberg, 2004, pp. 27–41.

[66] C. Zou and D. Hou, "LDA Analyzer: A Tool for Exploring Topic Models," 2014, pp. 593–596.

[67] S. Lee, J. Baker, J. Song, and J. C. Wetherbe, "An Empirical Comparison of Four Text Mining Methods," in *2010 43rd Hawaii International Conference on System Sciences (HICSS)*, 2010, pp. 1–10.

[68] D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin, "Evaluating Topic Models for Digital Libraries," in *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, New York, NY, USA, 2010, pp. 215–224.

[69] C. W. Arnold, A. Oh, S. Chen, and W. Speier, "Evaluating topic model interpretability from a primary care physician perspective," *Comput. Methods Programs Biomed.*, vol. 124, pp. 67–75, Feb. 2016.

[70] C. Musat, J. Velcin, S. Trausan-Matu, and M.-A. Rizoiu, "Improving Topic Evaluation Using Conceptual Knowledge.," in *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, vol. 3, pp. 1866–1871.

[71] D. Mimno, "The Details: Training and Validating Big Models on Big Data," *Journal of Digital Humanities*, 08-Apr-2013. [Online]. Available: http://journalofdigitalhumanities.org/2-1/the-details-by-david-mimno/. [Accessed: 31-Mar-2016].

[72] D. Walker, E. Ringger, and K. Seppi, "Evaluating supervised topic models in the presence of OCR errors," in *IS&T/SPIE Electronic Imaging*, 2013, pp. 865812–865812.

[73] J. R. Foulds and P. Smyth, "Annealing paths for the evaluation of topic models," in *Proceedings of the Thirtieth Conference Conference on Uncertainty in Artificial Intelligence*, 2014.

[74] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization Techniques for Assessing Textual Topic Models," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, New York, NY, USA, 2012, pp. 74–77.

[75] J. Chang and D. M. Blei, "Hierarchical relational models for document networks," *Ann. Appl. Stat.*, pp. 124–150, 2010.

[76] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 113–120.

[77] Y. Liu, L. Li, S. Wan, and Z. Gao, "Research on Chinese multi-document hierarchical topic modeling automatic evaluation methods," in *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2014, pp. 444–449.

[78] X. Yi and J. Allan, "Evaluating Topic Models for Information Retrieval," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, New York, NY, USA, 2008, pp. 1431–1432.

[79] E. Alexander and M. Gleicher, "Task-Driven Comparison of Topic Models," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 320–329, Jan. 2016.

[80] J. Foulds and P. Smyth, "Robust evaluation of topic models," *NIPS'13*, 2013.

[81] N. Niraula, R. Banjade, D. Ştefănescu, and V. Rus, "Experiments with Semantic Similarity Measures Based on LDA and LSA," in *Statistical Language and Speech Processing*, A.-H. Dediu, C. Martín-Vide, R. Mitkov, and B. Truthe, Eds. Springer Berlin Heidelberg, 2013, pp. 188–199.

[82] J. Chuang, S. Gupta, C. Manning, and J. Heer, "Topic model diagnostics: Assessing domain relevance via topical alignment," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 612–620.

[83] V. Mehta, R. S. Caceres, and K. M. Carter, "Evaluating topic quality using model clustering," in *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, 2014, pp. 178–185.

[84] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2015, pp. 399–408.

[85] X. Sun, "Textual Document Clustering Using Topic Models," in *2014 10th International Conference on Semantics, Knowledge and Grids (SKG)*, 2014, pp. 1–4.

[86] M. Summary, L. L. Beijing, and Y. Zhong, "A new evaluating method for Chinese text summarization not requiring," in *International Conference on Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009*, 2009, pp. 1–7.

[87] J. Tang, R. Huo, and J. Yao, "Evaluation of Stability and Similarity of Latent Dirichlet Allocation," in *2013 Fourth World Congress on Software Engineering (WCSE)*, 2013, pp. 78–83.

[88] S. Grant and J. R. Cordy, "Examining the relationship between topic model similarity and software maintenance," in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 303–307.

[89] K. Choi, J. H. Lee, C. Willis, and J. S. Downie, "Topic Modeling Users' Interpretations of Songs to Inform Subject Access in Music Digital Libraries," in *Proceedings of the 15th ACM/IEEE-CE on Joint Conference on Digital Libraries*, 2015, pp. 183–186.

[90] K. Farrahi and A. Ferscha, "Topic models towards high performance data mining and analysis," in *2013 International Conference on High Performance Computing and Simulation (HPCS)*, 2013, pp. 692–693.

[91] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proc. GSCL*, pp. 31–40, 2009.

[92] C. E. Shannon, "A Mathematical Theory of Communication," *SIGMOBILE Mob Comput Commun Rev*, vol. 5, no. 1, pp. 3–55, Jan. 2001.

[93] P. Turney and M. L. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," 2002.

[94] A. Arcuri and L. Briand, "A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering," *Softw. Test. Verification Reliab.*, vol. 24, no. 3, pp. 219–250, May 2014.

[95] D. Taeger and S. Kuhnt, "Statistical hypothesis testing," in *Statistical Hypothesis Testing with SAS and R*, John Wiley & Sons, Ltd, 2014, pp. 3–16.

[96] N. M. Razali, Y. B. Wah, and others, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests," *J. Stat. Model. Anal.*, vol. 2, no. 1, pp. 21–33, 2011.

[97] M. Unterkalmsteiner, "Coordinating requirements engineering and software testing," 2015.

10 APPENDICES

10.1 Appendix 1

Table 20: Data Extraction Form Questions

ID	Data Extraction form question	Relevant RQ from Section 4.1.1
1	In which year was the research published?	R1
2	In which country was the research performed?	R1
3	At which venue was the research published?	R1
4	What are the research areas/ key words related to the article?	R1
5	What are the topic models that were used in the article?	R1
6	What are the evaluation methods that were used in the articles?	R2
7	Which dataset has been used in the article?	R1
8	What was the research method used by the researchers in the article?	R1
9	Which quality criteria of topic models has been evaluated, and What are the metrics that have been used by the researchers to evaluate the topic models in the article?	R3

10.2 Appendix 2

Instructions for the task to be performed

- The main objective of this experiment is to evaluate the interpretability of topics generated using a topic modelling algorithms.
- An example has been provided for you, to better understand the task that you need to perform.

Topic Number	Topic Words	Topic Tags
1	Human, genome, dna, genetic, genes, sequence, gene, molecular, sequencing, map, information, genetics, mapping, project, sequences	Genetics, DNA, chromosome
2	Evolution, evolutionary, Species, organisms, Origin, biology, Groups, phylogenetic, Living, diversity, Group, new, common, life, two	Evolution, generation, bio-diversity, chromosome
3	Disease, host, Bacteria, diseases, Resistance, bacterial, New, strains, Control, infectious, Malaria, parasite, Parasites, united, Tuberculosis	Diseases, infection, cough, influenza
4	Computer, models, Information, data, Computers, system, Network, systems, Model, parallel, Methods, networks, Software, new, simulations	Computers, software, memory, processor, simulation

- The above table contains 4 different topics that have been matched with their respective tags.
- You have been given a table consisting of such 30 different (or same) topics, along with a list of 98 tags that have been sorted alphabetically.
- Your task is to carefully study the words in each topic, understand them; and then look at the list of tags that have been given to you. Then, assign the tags that you think are most suitable for that topic.
- Each topic may (or may not) have more than a single tag.
- Several topics may (or may not) contain the same tag.
- If you have any further questions do not hesitate to ask.

Figure 19: Instructions given to the subjects

Topic Number	Topic Words	Topic Tags
1	keyboard character key email tap input displayed select verify written portuguese spanish hold button press	
2	verify view list open app start listed typing order result user task panel scroll shown	
3	button edit delete tap press open deleted contact remove mode select removed app details fields	
4	update updates app device system download check package user notification make connection properly ota install	
5	message mms send open messages app received notification download device thread select verify displayed user	
6	key test field character text select press type hold application keyboard click special input open	
7	sms device send app verify thread test message view user open text conversation received highlighted	

Figure 20: A snapshot of the topics document given to the subjects

10.3 Appendix 3

Table 21: Appendix for research methods

S.No	Research Method	Articles
1	Experimentation	[6], [8], [14], [16], [17], [27]–[41], [43]–[45], [47]–[64]
2	Survey	[82]
3	Case Study	[11], [71]
4	Not Clear	[67], [90]

10.4 Appendix 4

Table 22: Appendix for topic models

S.No	Topic Model Used	Articles
1	LDA	[6], [8], [14], [16], [17], [27], [28], [30], [31], [33]–[35], [38], [39], [41], [42], [44], [45], [46], [48], [49], [51], [53], [55]–[58], [60], [62]–[65]
2	Variants of LDA	[53], [54], [72], [87]

3	HLDA and variants	[59], [61], [63], [77]
4	CTM	[20], [67]
5	LSA, PLSA	[56], [67], [81]
6	Others	[20], [59], [72], [75], [76], [78]
7	Not mentioned	[11], [57], [62], [65], [86]
8	unclear	[68], [79]

10.5 Appendix 5

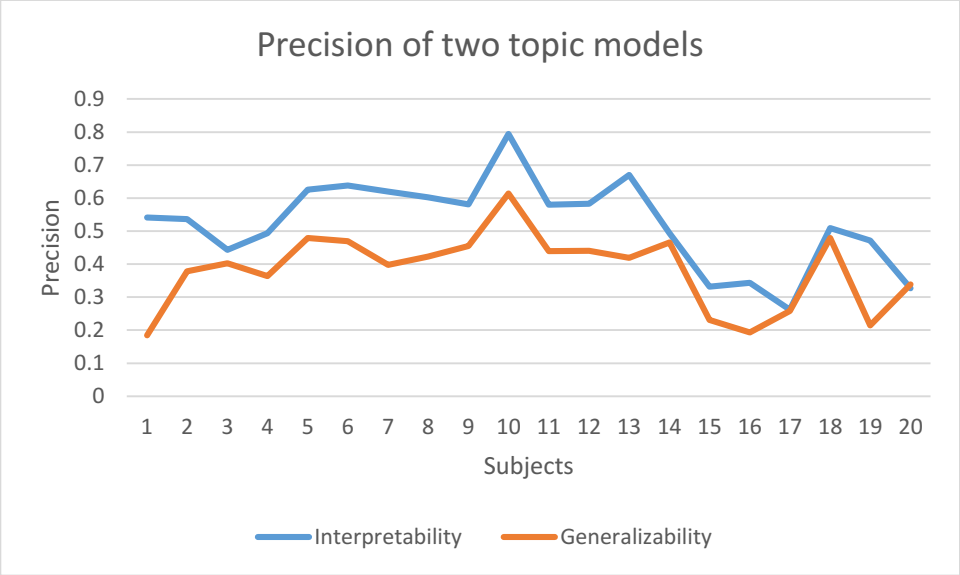

Figure 21: Precision for both topic models

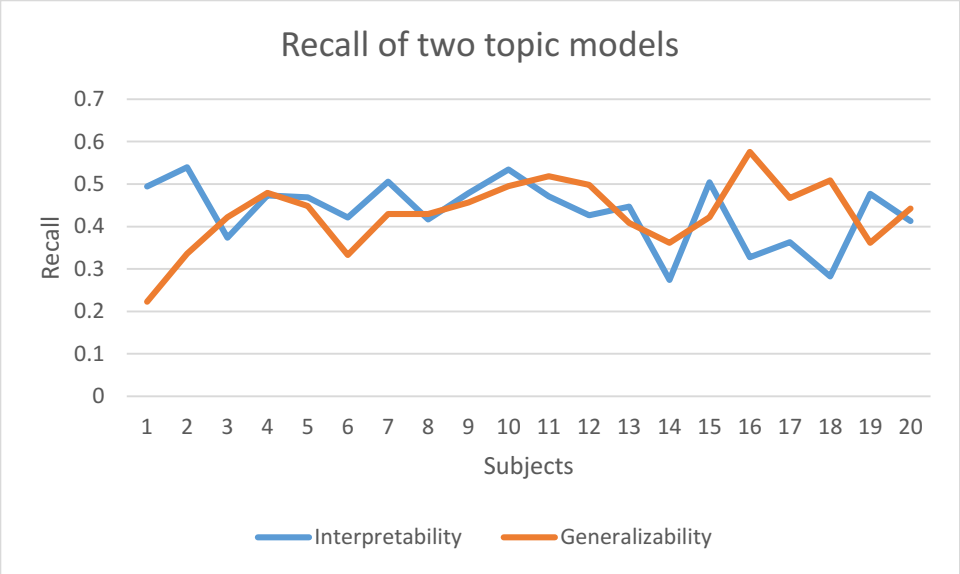

Figure 22: Recall for both topic models

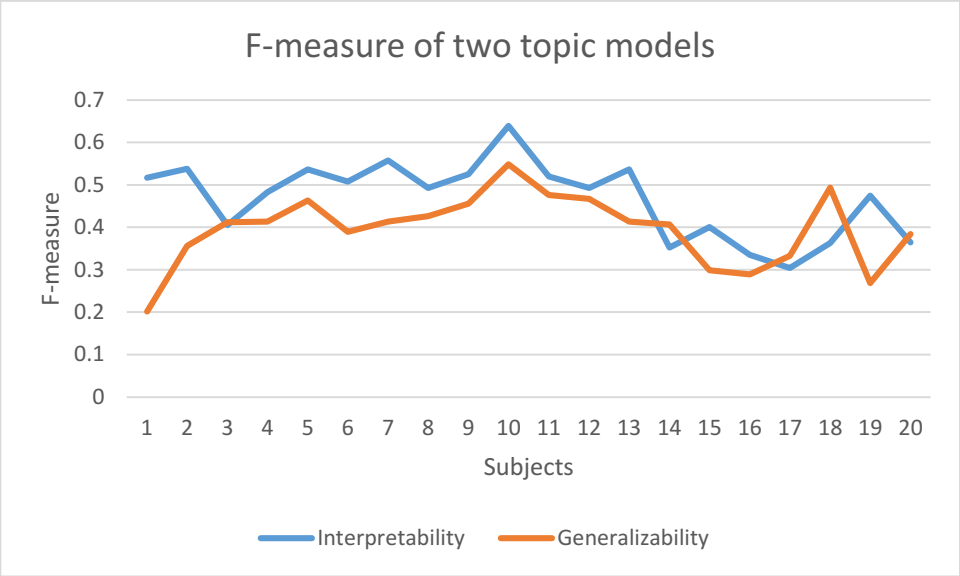

Figure 23: F-measure for both topic models

10.6 Appendix 6

Test case_ID	Test Case name	Description	Steps-instruction	Steps-expected	Tags	product version	case version
97550	Verify the user can browse photos in the Gallery App	It should be possible to navigate the photos in the Grid View of the Gallery App	1)From the Home screen, Open the Gallery App 2) Swipe up 3) Select a thumbnail	1) The Gallery is launched, then Grid View is properly displayed with all onscreen photo thumbnails fully loaded 2) The display scrolls down through the thumbnails in the Grid View 3) Selected thumbnail is displayed in Single Photo View	Gallery	XX	XX
97545	Delete an application by pressing the button "X" from the edit mode	It should be possible to delete an application	1) Press home button to go home if not already there. Long tap (>3 seconds) an app icon to open edit mode. 2) Press "delete" button on an app icon (small X).	1) Edit mode is launched (app icons are vibrating). 2)App is deleted	Home screen	XX	XX
975xx							
975xx							

The total test cases for the input of topic models are: 6417

The linguistic data from the test cases: Name, description, instruction steps, expected steps are given as input to two topic models to generate optimal number of topics for both criteria. The tags are excluded to avoid bias. In the next step, the optimal number of topics are generated by using two topic models. The Doc-topic proportions are also collected form the mallet. By this, we can understand which test cases falls under which documents. The doc-topic proportions threshold are set as 0.5. By this, we can identify the test case which constitutes more than 50% to the particular topic. Test cases which have [(0.4,0,4,0.2),(0.3, 0.3,0.4), or similar] doc-proportions are excluded.

Test case-file-name	Topic	Doc-topic proportion(test case-proportions to the topic)
97550.txt	29	0.7366
e97545.txt	14	0.8935
975xx.txt	12	0.5459
975xx.txt	10	0.9513

After identifying the tags to their corresponding topics in both the topic models, the total number of tags computed as 97.

Topic	Tags
29	Gallery, **,**,**,...
14	Home,**,**,**,...
12	mms, message, sms
10	Contacts, dialer

Now, 30 topics (15 from each topic model) are given along with the 97 identified tags. So assignment is to pick relevant tags from the given list and to assign to the 30 topics.