# Classifying Stack Overflow Questions Based on Bloom's Taxonomy

By

Manisha Dubey

Under the supervision of Dr. Vikram Goyal

Indraprastha Institute of Information Technology Delhi

July, 2016

# Classifying Stack Overflow Questions Based on Bloom's Taxonomy

Manisha Dubey

IIITD-MTECH-CS-GEN-14-014

July, 2016

Indraprastha Institute of Information Technology Delhi

Thesis Advisor: Dr. Vikram Goyal

Submitted in partial fulfillment of the requirements for the degree of Master of Technology

# CERTIFICATE

This is to certify that the thesis titled "Classifying Stack Overflow Questions Based of Bloom's Taxonomy" being submitted by Manisha Dubey to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

DR. VIKRAM GOYAL

DEPARTMENT OF COMPUTER SCIENCE

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY

# ACKNOWLEDGEMENT

After expressing gratitude towards God and my loving parents, I would like to express my sincere gratitude to my advisor Dr. Vikram Goyal for providing excellent guidance and being supportive throughout the journey. Without his patience, motivation and thoughtful insights, this work would never have been completed. A very special thanks to IIITD for providing an excellent infrastructure, environment and a flexible curriculum to carry out my work at a suitable pace. I would like to thank my friends for being so supportive and helping me out with the problems whenever I got stuck. Fun time with you people always made me feel fresh and enthusiastic. This section can't be completed without a vote of thanks to IT department and Bhawani Sir for his help and never ending support. Moreover, I am deeply thankful to all the faculty and admin staff here for their extremely supportive attitude.

# ABSTRACT

Bloom's Taxonomy is a framework which acts as a reference for classification of questions across different cognitive levels such as Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. It can be used to select questions in order to evaluate knowledge and understanding of students. We, in this thesis, work on the problem of knowledge management and try to classify questions asked on popular social networks like Stack Overflow (SO). The motivation for the problem comes from the SO being as one huge source of technical questions and answers which include current trending discussions also. Such a knowledge source can be very useful for the education domain. We first apply LDA to reduce the dimensions of each SO document and then use k-means algorithm on a collection having unlabeled and labelled documents to get the result. We obtain an accuracy of 30.2% with this approach. We further augment other features like score, answer count and view count to the obtained feature set and get an accuracy of 56.33%.

# TABLE OF CONTENTS

List of Tables

List of Figures

**1. Introduction**

1.1 Thesis Structure ……1

1.2 Motivation ……2

1.3 Research Contribution ……2

**2. Literature Review**

2.1 Related Work……3

2.2 Proposed Work ……..4

**3. Background**

3.1 Bloom's Taxonomy ……5

3.2 K-means Clustering ……8

3.3 Topic Modelling and LDA ……9

3.4 Clustering Validation Techniques ……13

**4 Experiment and Results**

4.1 Experimental Settings ……...15

4.2 Results and Analysis ……20

**5 Conclusion and Future Work** **24**

**6 Bibliography** 25

# LIST OF TABLES

3.1 Categories of Bloom's Taxonomy with Illustrative Examples……5

4.1: Structure of Stack Overflow Dataset……15

4.2: Structure of Posts.xml file……16

4.3: Classification of Training data……20

4.4: Precision Calculation……21

4.5 Recall Calculation……21

4.6: Precision Calculation for augmented feature vector……22

4.7: Precision Calculation for augmented feature vector……22

# LIST OF FIGURES

3.1 Bloom's Taxonomy……6

3.2 Intuition behind LDA……10

3.2 Graphical intuition of LDA……12

4.1 Keywords.txt……17

4.2 Composition.txt……18

4.3 Cluster Dumper output……19

4.4 Sequence Dumper output……19

4.5 Comparison of precision for different feature vectors……23

4.6 Comparison of recall for different feature vectors……23

CHAPTER 1

# INTRODUCTION

The real objective of learning and teaching can only be accomplished by having written examination in order to evaluate the learning outcome. Written examination is a standard way of assessment of learners, their knowledge and understanding. Hence, it is very essential to select good quality questions to assess different levels of cognitive. However, developing such questions is always a challenging task since the questions must be provided according to the subject and must be reasonable enough to match the cognitive level of students [10]. Bloom's Taxonomy acts as a framework for the production of examination questions.
The purpose of Bloom's Taxonomy is to provide reference to the educators while designing questions. Bloom's Taxonomy also proves to be useful for evaluating the quality of question paper and automatic questions-answering system. This study has proposed a method to cluster questions in accordance with Bloom's Taxonomy for educational purpose to overcome the problem of question classification.

## 1.1 THESIS STRUCTURE

This chapter details the motivation behind this thesis and research objectives achieved with this work.

Chapter 2 gives the background information. It builds basic foundation on Bloom's Taxonomy. Also, it explains about topic modelling. It explains the basics of k-means algorithm.

Chapter 3 briefly explains the related work done in this regard.

Chapter 4 explains the experiments and results. It elaborates about the 2-step approach to cluster questions according to Bloom's Taxonomy. Subsequently, it discusses the results and analysis of the results in terms of precision and accuracy of the model.

Finally, Chapter 5 discusses the conclusion of the work done. Moreover, it proposes the possibilities for future work.

## 1.2 MOTIVATION

Examinations forms a fundamental component of learning process. It is a method of assessment of learning objective, knowledge and understanding of students. However, it is challenging for academicians to select appropriate questions across different cognitive levels to assess students. The questions to be selected for examinations should be appropriate and good quality to evaluate the cognitive ability of students. To provide a common reference for such a challenging task, Bloom's Taxonomy act as a guide for production of exam questions. Bloom's Taxonomy is a hierarchical type framework for educational objectives [2]. Bloom's Taxonomy is also useful in respect of automating answering systems in forums. It can also be helpful in evaluating the quality and goodness of the question paper. Although this goal has been achieved various techniques like classification, Artificial Neural Networks, Rule-based approach. But, one could also apply unsupervised approach to categorize the questions.

## 1.3 RESEARCH CONTRIBUTION

This model proposed a new method of 2-step approach towards categorization of questions according to Bloom's Taxonomy. The model suggested performs topic modelling to obtain the probabilities of the topics. This acts as a feature vector for clustering. Subsequently, we apply k-means clustering of this feature vector to cluster questions into 6 categories. In this way we can categorize questions according to Bloom's Taxonomy. The model proposed performs better than uniform distribution. Subsequently, we have augmented that feature vector with others extra features which resulted in increase of accuracy of our model. Also, we have analyzed the problems associated with Bloom's Taxonomy. Not all questions can be assigned a discrete category. Moreover, since there is no absolute definition of each category, it may be difficult to assign a single category to each question. Hence, the categories are at times ambiguous. Moreover, we have learnt that it may be efficient to use output of topic modelling as a feature vector to perform clustering. The result of clustering is validated by external validation method in terms of precision and recall.

CHAPTER 2

# LITERATURE REVIEW

This section discusses literature review related to our work. First section describes the literature review related to our work. Next section discusses about our proposed approach.

## 2.1 RELATED WORK

Van Hoeij et al. [3] has evaluated cognitive levels of short essay questions using a simplified classification tool based on Bloom. The study concludes that only moderate-level agreement on the classification of test items. However, given better instructions and an improved classification procedure, this technique may be useful in quality assurance of examination.

Chang and chung [4] applied Bloom's Taxonomy to evaluate and classify English question item's cognition level. Their research included 14 general keywords for Bloom's Taxonomy.

Haris and Omar [6] applied rule based approach to identify various keywords and verbs, which helps to find the cognitive level of question. Moreover, it combines statistical approach by using n-gram. Creating a hybrid technique by using rules for syntactic approach and n-gram for statistical approach helps to perform the categorization in a better way. Also, rules are developed by combining part-of-speech (POS), regular expression and specific keyword that exists in training set. Although rule-based technique are time-consuming and not as dynamic as machine-learning algorithms, they exhibit satisfactory performance.

Yusof and Hui [5] have used Artificial Neural Network to determine cognitive category according to Bloom's Taxonomy. Three different feature sets were proposed and two of them can reduce the dimensionality of feature space to low input. Also, document frequency (DF) feature reduction method offers an interesting combination of classification precision and convergence time.

Abdulhadi et al. [8] has proposed a new method to classify questions according to cognitive levels of bloom's taxonomy by implementing a combination strategy based on voting algorithm that combines three machine learning classifiers. In this work, the classifiers used are Support Vector Machine (SVM), Naïve Bayes (NB) and k-Nearest Neighbour (k-NN) that are used to classify question with feature selection methods namely, Chi-Square, Mutual Information and Odd Ratio. Then a combination algorithm is used to integrate overall strength of all classifiers. This has integrated different feature selection methods and classification algorithms to synthesize classification procedure more accurately.

Barua et al. [7] have tried to perform trend analysis among programmers on Stack Overflow to understand thoughts and needs of developers. They have tried to gain insight over the topics present in developer discussions, their relationship and trends over time.

Allamanis et al [2] have performed topic modelling analysis and tried to gain insight over the concepts, types and code. They have analyzed the categories of programming tasks that are more common in various languages. They have tried to identify the question types that were mostly associated with particular programming constructs or identifiers.

## 2.2 PROPOSED WORK

We have aimed at classifying Stack Overflow questions according to Bloom's taxonomy. Bloom's Taxonomy acts as a guideline to categorize questions into various levels. To categorize questions, we will use unsupervised approach. We have used k-means algorithm to classify questions into 6 categories. Clustering is performed on a feature vector representing the documents. There are various ways to represent a set of documents.

Bag-of-words model represents documents as a series of words, irrespective of order. This bag-of-words may be used as feature vector to perform clustering. But, problem associated with this approach is sparsity and high-dimensionality. To overcome these problems, we have used composition of a document over various topics. This composition includes the topic probabilities over document set. Such composition of documents will reduce the dimension of feature vector. Also, we can overcome the problem of sparsity associated with bag-of-words model.

We can get composition of a document over various topics using topic modelling. Topic Modelling represents topics as distributions over words, and documents as distributions over topics. We have used Latent Dirichlet Allocation to perform topic modelling.

LDA is a Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. Topic modelling can connect words with similar meanings and distinguish between uses of words with multiple meanings using contextual analysis. So, the topic probabilities provide an explicit representation of a document.

Also, we have included some features like Score, Answer Count and View Count. Then we have normalized these features to avoid variance in data. We have augmented these normalized features with topic probabilities. This augmented matrix is treated as new feature vector to represent our set of documents. Now, we have performed k-means clustering on this feature vector. Due to large size of data, we have done clustering using mahout.

Validation of our approach is done using external validation technique. This is done by comparing the results of clustering to externally known results. For this, we have used a test dataset of 500 documents which is used to assign a category to each cluster according to frequency of each category. Then we have manually classified 1500 documents. Now we have calculated accuracy and precision on this set of documents.

CHAPTER 3

# BACKGROUND

## 3.1 BLOOM'S TAXONOMY

Written examination prove to be an integral entity of education system with a goal of assessment of learner's knowledge and understanding. Therefore, a well formed question paper is must for correct assessment of students, thereby making it a challenging task. A well-formed question paper comprises of a perfect blend of questions of different cognitive levels, in correct proportion, in order to assess student's cognitive level and understanding of the concept appropriately. Thus, bloom's taxonomy act as a framework for measuring educational objectives and estimate learner's knowledge.

Bloom's taxonomy consists of 6 cognitive levels, explained as follows:-

1) **Knowledge**
   - It deals with ability to recall information.
   - Sample keyword used for this domain are arrange, recognize, define etc.

2) **Comprehension**
   - It deals with understanding of the concept and the ability to deduce the knowledge.
   - Sample keywords for this cognitive level are classify, describe, discuss etc.

3) **Application**
   - It deals with the ability to use of knowledge in order to apply it to a new and unfamiliar problem.
   - It uses keywords like interpret, solve, use etc.

4) **Analysis**
   - It deals with comprehension of constituent parts and making the relationship between its constituent entities.
   - The keywords majorly used are analyze, differentiate, categorize etc.

5) **Synthesis**
   - It involves integrating the knowledge gained to a complete new problem.
   - Keywords used are arrange, create, design etc.

6) **Evaluation**
   - It involves creation of opinion or decision related to the topic of knowledge.
   - Keywords used are justify, predict, compare etc.

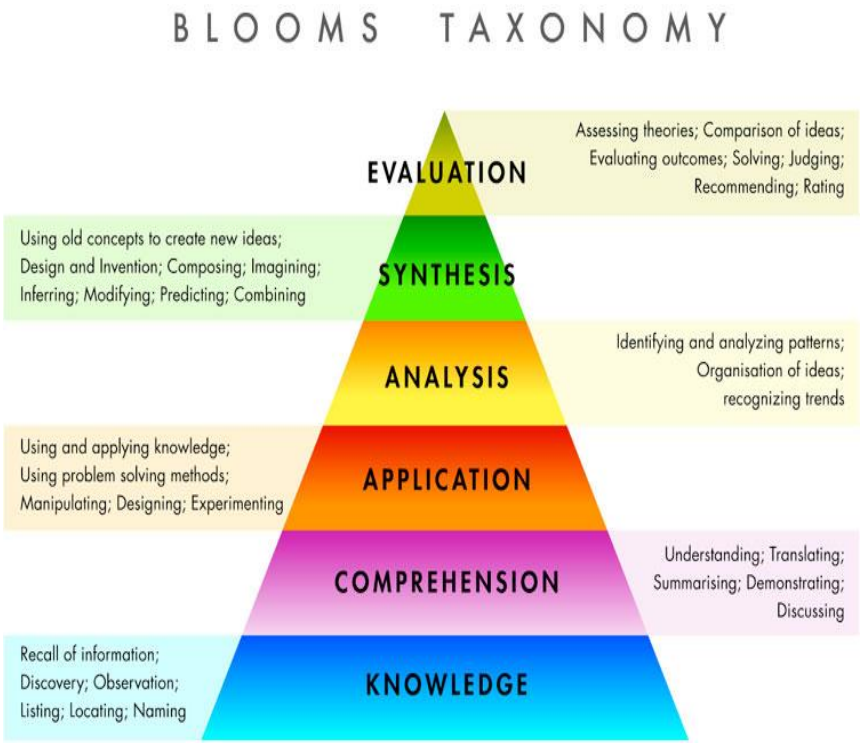


*Figure 3.1 Bloom's Taxonomy*

Bloom's Taxonomy may be helpful for education purposes in the following manner:-

- It can be used to create good quality question papers in order to define the cognitive level of learner.
- It can be used to check the quality of any question paper. So, it can be used in quality control and quality assurance of examination. [3]
- Question classification is an important step towards question answering. So, in order to develop automatic question-answering system, it's very important to classify the questions.

| CATEGORY | SAMPLE KEYWORDS | SAMPLE BEHAVIOUR | SAMPLE QUESTIONS |
|---|---|---|---|
| Knowledge | Arrange, recognize, relate, label, list, memorize, recall, define | Students are able to define the 6 levels of Bloom's taxonomy | Define Inheritance concept. |
| Comprehension | Classify, explain, indicate, locate, recognize, describe, discuss, express. | Students can explain the purpose of Bloom's taxonomy | Explain the structure of a method in the program. |
| Application | Demonstrate, sketch, illustrate, operate, practice, schedule. predict, explain, interpret, employ, solve, use, write. | Students write an instructional objective for each Bloom's taxonomy level. | Demonstrate the relationship of all the packages, classes and methods of the program. |
| Analysis | Analyze, appraise, make a distinction calculate, categorize, differentiate, discriminate, distinguish, examine, list. | Students compare and contrast cognitive and affective domains | List the advantages and disadvantages of using a container class such as ArrayList in place of an array. |
| Synthesis | Arrange, assemble, create, design, collect, compose, develop, set up, propose, write, organize, plan. | Students design a classification method for educational objectives that combine cognitive, affective, and psychomotor domains. | Write a JAVA program to show the Overloading concept. |
| Evaluation | Appraise, judge, predict, assess, attach, compare, defend choose, estimate, rate, compare, Justify | Students judge the effectiveness of Bloom's taxonomy. | Justify the concept of inheritance and give the sample of code to illustrate your answer. |

*Table 3.1: Categories of Bloom's Taxonomy with Illustrative Examples*

## 3.2 K-MEANS CLUSTERING

Clustering is the process of partitioning a group of data points into a small number of clusters. It finds application in browsing or navigation system, since it clusters related information together.

K-means proceeds by selecting k initial cluster centers and then iteratively refining them. We have n data points xi, i=1...n that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to find the positions μi, i=1...k of the clusters that minimize the distance from the data points to the cluster. K-means clustering solves

$$\underset{c}{\operatorname{Arg\,min}} \sum_{i=1}^{k} \sum_{x \in c_i} d(x,\mu_i) = \underset{c}{\operatorname{arg\,min}} \sum_{i=1}^{k} \sum_{x \in ci} \|x-\mu_i\|^2$$

where ci is the set of points that belong to cluster i. The K-means clustering uses the square of the Euclidean distance $d(x,\mu i)=\|x-\mu i\|^2$. This problem is not trivial (in fact it is NP-hard), so the K-means algorithm only hopes to find the global minimum, possibly getting stuck in a different solution.

**Algorithmic steps for k-means Algorithm**

Let $X = \{x_1,x_2,x_3,\ldots\ldots..,x_n\}$ be the set of data points and $V = \{v_1,v_2,\ldots\ldots.,v_c\}$ be the set of centers.

1) Randomly select *'c'* cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, *'$c_i$'* represents the number of data points in *$i^{th}$* cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

So, the algorithm converges when there is no further change in assignment of instances to clusters.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect.

K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors. To perform document clustering, one should be able to represent words in terms of feature vector.

Vector Space Model (VSM) is the most popular method in representing documents. This model assumes that sequence of words appearing in the documents is not important. The model representing documents as series of words irrespective of order is known as bag of words. Problems associated with this model are: sparsity and high dimensionality, because of which performance of clustering algorithms decreases.

In order to improve efficiency of clustering process, dimensionality of feature space is reduced. There are two methods of reducing dimensionality:

1) **Feature Extraction**
   It extracts new set of features from original ones using techniques like Principal Component Analysis. It can also be done by performing word clustering before document clustering [14]. The demerit of feature extraction is that the original features lose their meaning, due to which it becomes difficult to interpret the results of clustering.

2) **Feature Selection**
   This process chooses a subset from the original features. Feature Selection may be supervised or unsupervised, depending on the requirement of class label information.
   Unsupervised feature selection methods include document frequency (DF), Term Contribution (TC).

   It is inappropriate to use only statistical methods like TF-IDF, N-gram to classify the exam questions into Bloom's Taxonomy category since statistical techniques require large data in each document to obtain high accuracy.

   LDA represents topics as distributions over words, and documents as distributions over topics. LDA can be viewed as technique of dimensionality reduction [9].

## 3.3 TOPIC MODELLING AND LDA

A topic comprises of cluster of words that occur often together. A document is a collection of topics where each topic has some particular probability of generating a particular word. So, topic modelling represents documents as mixtures of topics with certain probability and each topic comprises of words related to that topic. Topic modelling can connect words with similar meanings and distinguish between uses of words with multiple meanings using contextual analysis.

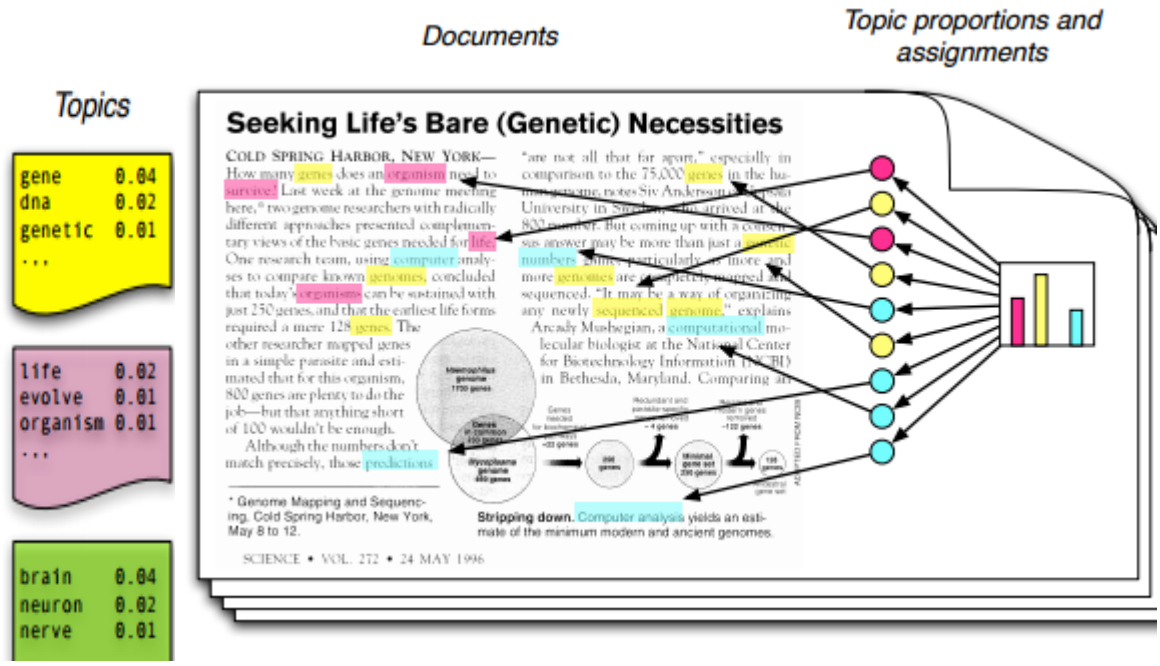


*Figure 3.2: Intuition behind LDA*

The intuition behind Latent Dirichlet Allocation is that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic [16].

We formally define a topic to be a distribution over a fixed vocabulary. We assume that these topics are specified before any data has been generated. Now, for each document in the collection, we generate the words in a two-stage process.

1. Randomly choose a distribution over topics.
2. For each word in the document
   (a) Randomly choose a topic from the distribution over topics in step #1.
   (b) Randomly choose a word from the corresponding distribution over the vocabulary.

This statistical model reflects the intuition that documents exhibit multiple topics. Each document exhibits the topics with different proportion (step #1); each word in each document is drawn from one of the topics (step #2b), where the selected topic is chosen from the per-document distribution over topics (step #2a) [16]

Latent Dirichlet Allocation is a common method of topic modelling. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document [9].

Dirichlet Distribution is a family of continuous multivariate probability distribution. Dirichlet is a distribution specified by a vector parameter α containing some *α*i corresponding to each topic *i*, which we write as Dir(α). The formula for computing the probability density function for each topic vector *x* is proportional to the product over all topics *i* of $x_i^{\alpha_i}$. $x_i$ is the probability that the topic is *i*, so the items in *x* must sum to 1.

The 'Latent' part of LDA comes into play because in statistics, a variable we have to infer rather than directly observing is called a "latent variable". We're only directly observing the words and not the topics, so the topics themselves are latent variables.

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

LDA assumes the following generative process for each document w in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$.

2. Choose $\theta \sim \text{Dir}(\alpha)$.

3. For each of the N words $w_n$:

    (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.

    (b) Choose a word wn from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic $z_n$.

A k-dimensional Dirichlet random variable θ can take values in the (k −1)-simplex (a k-vector θ lies in the (k−1)-simplex if $\theta_i \geq 0$, $\sum^{k}_{1=1} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right) \theta^{\alpha_1 - 1}_{1} \cdots \theta^{\alpha_k - 1}_{k}}{\prod^{k}_{i=1} \Gamma(\alpha_i),}$$

where the parameter α is a k-vector with components αi > 0, and where Γ(x) is the Gamma function.

The Dirichlet is a convenient distribution on the simplex — it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. In Section 5, these properties will facilitate the development of inference and parameter estimation algorithms for LDA.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

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

where $p(z_n \mid \theta)$ is simply $\theta_i$ for the unique i such that $z^i_n = 1$. Integrating over θ and summing over z, we obtain the marginal distribution of a document:

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

Finally, taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left( \prod_{n=1}^{Nd} \sum_{zdn} p(z_{dn} \mid \theta_d)\, p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d$$

**Graphical Representation of LDA**

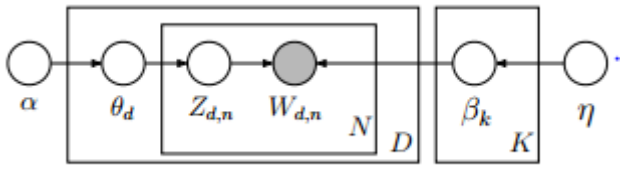


*Figure 3.3: Graphical intuition of LDA*

Each node is a random variable and is labeled according to its role in the generative process. The hidden nodes–the topic proportions, assignments and topics—are unshaded. The observed nodes—the words of the documents—are shaded. The rectangles are “plate” notation, which denotes replication. The N plate denotes the collection words within documents; the D plate denotes the collection of documents within the collection. [17]

**Example:** At a broader level, LDA assumes that each document is generated by –

1) From Dirichlet distribution for k, sample random distribution of topics.
2) For each topic, pick a distribution of words for that topic from the Dirichlet distribution of that topic.
3) For each word in document k,
   a) From the distribution of topics selected for k, sample a topic like "engineering institution".
   b) From the distribution selected for "engineering institution", pick the current word.

   Let's say the document starts with "IIITD is one of the best engineering colleges."

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

So, P (B|A) where B is the event of "IIITD" being generated from current word and A is the event of picking the topic "engineering institutions". P (A) is the probability of "engineering institutions" being selected from document's topic distribution. P (B) is the probability of generation of "IIITD" which is sum over all topic selections A. Now using Bayes theorem, we can find P (A|B), which is the probability that topic A generated word B.

So, we will know probability of each topic per word. Assuming that each word is independent of other in a document, overall probability of a topic throughout a document is product of P(A|B) at each word B.

So, we can find most dominant topic in a document.

## 3.4 CLUSTERING VALIDATION TECHNIQUES

"Clustering Validation techniques can be classified into 4 categories:-

1) **Relative Clustering Validation**
   It evaluates the clustering structure by varying different parameter values for same algorithm.

2) **External Clustering Validation**
   It compares the result of cluster analysis to externally known result, such as externally provided class labels.

3) **Internal Clustering Validation**
   It uses internal information of the clustering process to evaluate goodness of clustering structure without reference to external information.

**4) Clustering Stability Validation**

It evaluates the consistency of clustering result by comparing it with clusters obtained after each column is removed, one at a time."

Some of the measures of external clustering validation are:

**1) Purity**

Each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. Precision and recall can be calculated for this.

**2) Rand Index**

We want to assign two documents to the same cluster if and only if they are similar. A true positive (TP) decision assigns two similar documents to the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters. There are two types of errors we can commit. A (FP) decision assigns two dissimilar documents to the same cluster. A (FN) decision assigns two similar documents to different clusters. The Rand index measures the percentage of decisions that are correct.

$$\text{Rand Index} = \frac{TP + TN}{TP + FP + TN + FN}$$

**3) F-measure**

The Rand index gives equal weight to false positives and false negatives. Separating similar documents is sometimes worse than putting pairs of dissimilar documents in the same cluster. F measure penalize false negatives more strongly than false positives.

$$\text{F-measure} = \frac{TP}{TP + FP}$$

CHAPTER 4

# EXPERIMENTS AND RESULTS

This section consists with two sub-sections. First section deals with experiments. It explains about our dataset considered. It explains overview of the steps taken for topic modelling and clustering. Subsequently, it explains about the augmented matrix considered for experiments. Second section discusses the results and its analysis.

## 3.1 EXPERIMENTAL SETTINGS

Stack Overflow is one of the most popular platforms for programming based Question-Answering activities. It has a huge database of relevant question-answers, which makes it a great option for researchers to perform experiments upon. It is a well-structured Question-Answering website covering a large domain of topics.

Stack Overflow publishes anonymized version of all user-contributed Stack Exchange content. The data is published under Creative Commons BY-SA 3.0 license which makes it possible for us to use it for analysis purpose. Each site can be downloaded individually, and includes an archive with Posts, Users, Votes, Comments, Badges, PostHistory, and PostLinks.

The complete dataset is organized into 8 files, all in XML format.

| FILENAME | DESCRIPTION |
|---|---|
| Badges.xml | Information about user badges |
| Posts.xml | Questions and Answers |
| PostLinks.xml | Related/duplicate post links |
| PostHistory.xml | Edits related to posts |
| Comments.xml | Comments present in posts |
| Tags.xml | Tags assigned to the posts |
| Users.xml | User information |
| Votes.xml | Votes earned by posts |

*Table 4.1: Structure of Stack Overflow Dataset*

Since we need data related to questions and answers, we will be mining Posts.xml file from the data dump. Posts.xml consist of both questions and answers. Structure of Posts.xml is as:-

| FIELD | EXAMPLE |
|---|---|
| Id | “125677” |
| PostTypeId (1 for question, 2 for answer) | “1” |
| ParentId (If PostTypeId is 2) | |
| AcceptedAnswerId (If PostTypeId is 1 ) | |
| CreationDate | “2012-03-06T19:09:38.503” |
| Score | "26" |
| ViewCount | “117” |
| Body | |
| OwnerUserId | |
| LastEditorUserId | |
| LastEditDate | |
| LastActivityDate | |
| Title | "Using ACM or arXiv based tags" |
| Tag | |
| AnswerCount | |
| CommentCount | |

*Table 4.2: Structure of Posts.xml file*

We have extracted ‘Title’ and ‘Body’ from each post and all the possible answers related to each question using SAX Parser on Posts.xml. For each question, a different document is created where the name of document is the Id of the document. Then a corpus is created by compiling all such documents in a folder. MALLET requires the documents to be used as input to be kept in separate files, in a directory that contains no other files.

We have used MALLET to perform topic modelling. MALLET is a **MA**chine **L**earning for **L**anguag**E** **T**oolkit which uses implementation of Gibbs Sampling to perform topic modelling. Before performing topic modelling, we have removed stop-words. Stop-words are generally conjunctions or adverbs which have no contribution towards clustering process, and sometimes have negative influence [13].

We train a topic model with 100 iterations for 150 topics on all questions and their corresponding answers. We have taken large number of topics because it allows formation of fine-grained clusters.

Output of mallet comprises of 2 files:-

1) Keywords
   It consists of top keywords for each topic. The output is shown in the figure 3.1. The first number is the topic, second number indicates weight of the topic followed by the list of top 20 words of that particular topic.

2) Composition
   It indicates the breakdown, in percentage of each topic, corresponding to each input file.

| |
|---|
| 1 0.33333 find solution found problem ve work similar solve solutions answer simple don hope couldn question idea good searching didn works |
| 2 0.33333 button dialog onclick click button builder onclicklistener btn alertdialog show view setonclicklistener clicked dialoginterface alert void findviewbyid cancel code create |
| 12 0.33333 json string response httpclient params jsonobject http catch httppost tostring null execute log getstring client jsonarray gson post httpresponse data |
| 34 0.33333 string amp length append tostring strings equals substring stringbuilder sb trim charat return indexof stringbuffer valueof char format equalsignorecase mystring |
| 63 0.33333 log info debug level logging logger logs console debugging trace appender getlogger debugger warn output information slf severe levels warning |
| 112 0.33333 problem issue works fine ve working fix bug work problems issues doesn worked wrong correctly solve fixed solved reason strange |

*Figure 4.1: Keywords.txt*

```
0   file:/home/shared/manisha_postData/3192109.txt 69   0.35249042145593873   80
0.1743295019157088   87   0.07088122605363985   136   0.0421455938697318   18
0.019157088122605366   145   0.013409961685823755   125   0.013409961685823755   112
0.013409961685823755   58   0.013409961685823755   88   0.007662835249042145 …….

2   file:/home/shared/manisha_postData/14760488.txt   44   0.13562559694364854   89
0.11270296084049666   148   0.10983763132760269   16   0.09264565425023878   33
0.055396370582617   7   0.055396370582617   25   0.038204393505253106   129
0.03533906399235912   63   0.03247373447946514   104   0.026743075453677174……
```

*Figure 4.2: Composition.txt*

The composition of the topics will be used by our k-means algorithm as a feature-vector to perform clustering.

Clustering has been done using Mahout. Apache Mahout provides implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification. Apache Mahout is integrated into Hadoop/HDFS and implements distributed memory algorithms which can be applied to data sets that are much larger than can be handled by other techniques. Hadoop is a framework that allows the processing of certain types of tasks in a distributed environment using commodity machines that allows it to massively scale horizontally. Its main components are the map-reduce execution framework and the HDFS distributed file-system.

Mahout performs k-means clustering on vectors. A vector is a representation of object on which clustering has to be performed. So, all the vectors are written to SequenceFile format, which is read by k-means algorithm in mahout. SequenceFile is a format from Hadoop library that encodes a series of key-value pairs. In our case, the composition file from the output of mallet will be converted to SequenceFile format.

After converting it into SequenceFile format, we have applied k-means clustering. The distance measure chosen is Euclidean. And the documents were clustered into 6 groups.

The output of mahout can be analyzed in 2 ways:-

1) Using SequenceDumper
   The output file consists of ClusterID and bean containing weight, distance and vector representing the document. Weight indicates the probability that the vector is a member of cluster. For k-means, it is 1. Distance indicates 1/(1+distance) where the distance is between the cluster center and the vector using the chosen DistanceMeasure. So, it represents cluster for each file in the corpus.

2) Using ClusterDumper
   This file represents the information about clusters. It consists of clusterID, number of points in each cluster, centroid as a vector, radius as a vector and then each point in the cluster.

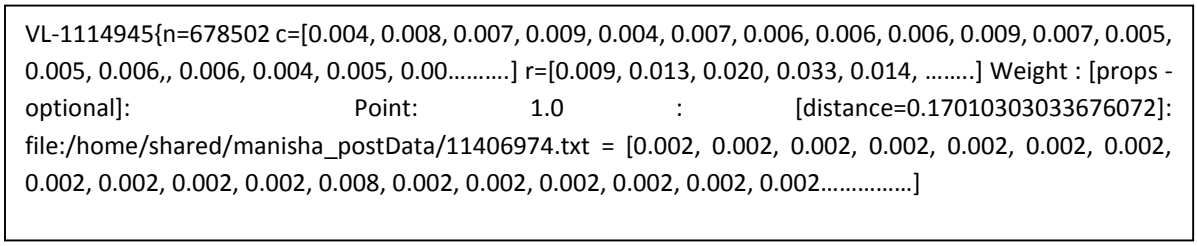


*Figure 4.3: ClusterDumper output*

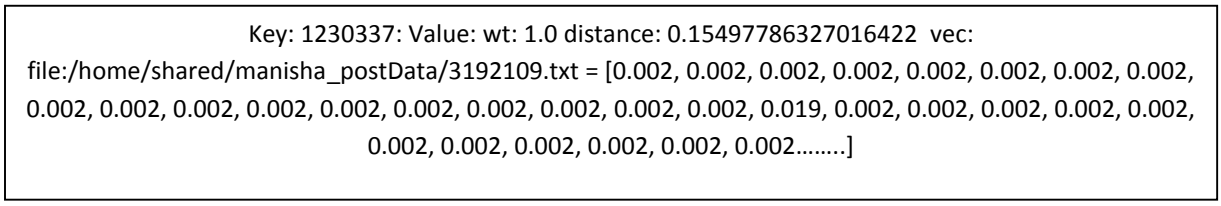


*Figure 4.4: SequenceDumper Output*

In part B of the experiment, we will perform same experiment with augmented feature matrix. For this augmented matrix, we will include Score, ViewCount and AnswerCount. These features are then normalized. Score is normalized by decimal scaling method and AnswerCount and ViewCount are normalized by min-max normalization. This matrix along with the composition matrix is now treated as feature vector. Now, clustering is performed on this vector.

## 3.2 RESULTS AND ANALYSIS

Since there is no definite way to determine optimum allocation of cognitive categories, we will perform external clustering. To evaluate the quality of clustering, we will use purity measure discussed in chapter 2.

500 files were manually classified and each cluster was assigned a category depending on the category of those 500 files. Now, this will act as training data. Training data has been allocated to categories intuitively based on instances of maximum allocation. This allocation has been done to the best of our ability.

Now, we will check efficiency of our clustering on the basis of 1500 files which we have classified for the purpose of evaluation. This set of 1500 files will act as test data. We will compare the category assigned by clustering algorithm and that of manual classification in order to find precision and recall.

| CLUSTERID | NUMBER OF FILES | CATEGORY |
|---|---|---|
| 1098773 | 8 | Synthesis |
| 1114945 | 228 | Knowledge |
| 1230337 | 38 | Application |
| 26806 | 73 | Analysis |
| 471550 | 152 | Comprehension |
| 902566 | 1 | Evaluation |

*Table 4.3: Classification of Training data*

**PRECISION AND RECALL**

Precision represents fraction of retrieved instances that are relevant. So,

$$\text{Precision} = \frac{\text{True Positive}}{\text{True positive + False Positive}}$$

Recall represents fraction of relevant instances that are retrieved. So,

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

So, after performing experiments, we have calculated True Positive and False Positive by comparing manually classified documents and the results which have been obtained after performing experiments. Precision can be tabulated as:

| CATEGORY | TRUE POSITIVE | FALSE POSITIVE | PRECISION (%) |
|---|---|---|---|
| **KNOWLEDGE** | 192 | 495 | 27.94 |
| **COMPREHENSION** | 130 | 261 | 33.24 |
| **APPLICATION** | 96 | 75 | 56.14 |
| **ANALYSIS** | 35 | 192 | 18.22 |
| **SYNTHESIS** | 0 | 12 | 0 |
| **EVALUATION** | 0 | 12 | 0 |
| **OVERALL** | **453** | **1047** | **30.2** |

*Table 4.4: Precision Calculation*

Precision for complete 1500 elements is **30.2%.**

Recall can be tabulated as:

| CATEGORY | TRUE POSITIVE | RELEVANT ITEMS | RECALL (%) |
|---|---|---|---|
| KNOWLEDGE | 192 | 358 | 53.6 |
| COMPREHENSION | 130 | 301 | 43.1 |
| APPLICATION | 96 | 708 | 13.5 |
| ANALYSIS | 35 | 76 | 46.05 |
| SYNTHESIS | 0 | 49 | 0 |
| EVALUATION | 0 | 8 | 0 |

*Table 4.5: Recall Calculation*

Results for part B of the experiment where we have included features like Score, AnswerCount and ViewCount are as follows:

| CATEGORY | TRUE POSITIVE | FALSE POSITIVE | PRECISION (%) |
|---|---|---|---|
| **KNOWLEDGE** | 62 | 43 | 59.04 |
| **COMPREHENSION** | 145 | 97 | 59.91 |
| **APPLICATION** | 560 | 430 | 56.56 |
| **ANALYSIS** | 44 | 32 | 57.89 |
| **SYNTHESIS** | 29 | 46 | 38.66 |
| **EVALUATION** | 5 | 7 | 41.66 |
| **OVERALL** | **845** | **655** | **56.33** |

*Table 4.6: Precision Calculation for augmented feature vector*

| CATEGORY | TRUE POSITIVE | RELEVANT ITEMS | RECALL (%) |
|---|---|---|---|
| KNOWLEDGE | 62 | 358 | 17.31 |
| COMPREHENSION | 145 | 301 | 48.12 |
| APPLICATION | 560 | 708 | 79.09 |
| ANALYSIS | 44 | 76 | 57.89 |
| SYNTHESIS | 29 | 49 | 59.18 |
| EVALUATION | 5 | 8 | 62.5 |

*Table 4.7: Recall Calculation for augmented feature vector*

Precision for this new feature vector is **56.33%**. So, overall precision has increased if we will consider extra features along with topic composition of the documents.

Histograms below show the comparison of precision and recall for normal feature vector and augmented feature vector with respect to different cognitive levels of Bloom's taxonomy.

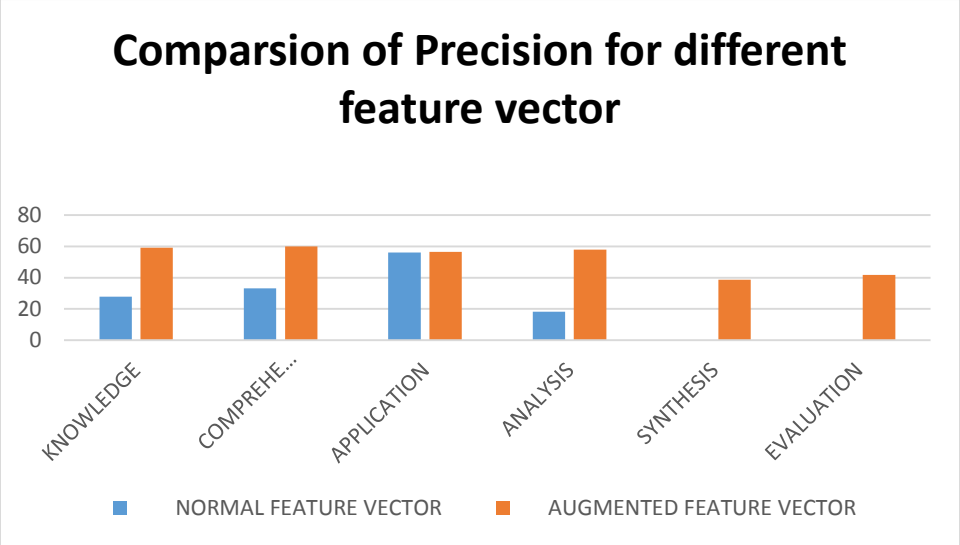


*Figure 4.5: Comparison of precision for different feature vectors*

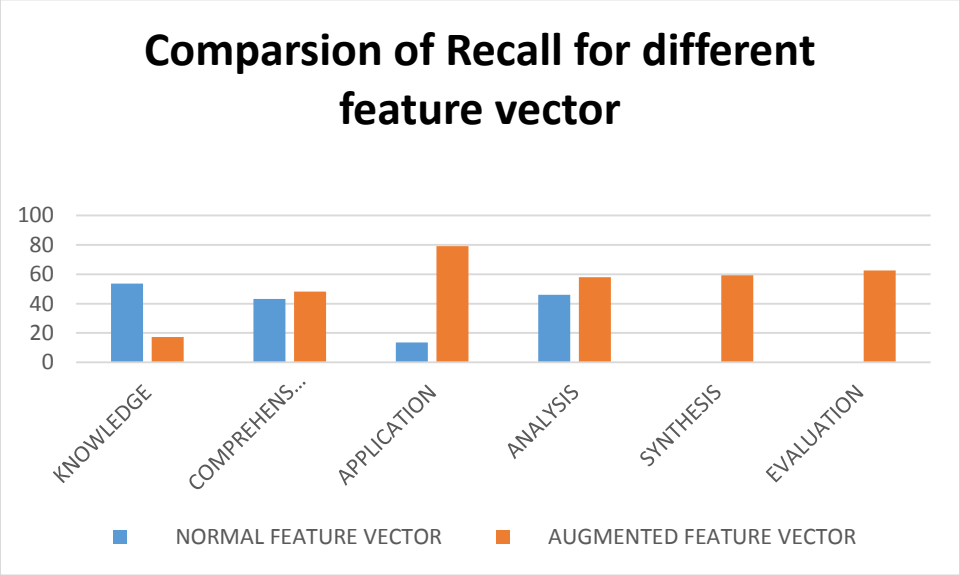


*Figure 4.6: Comparison of recall for different feature vectors*

CHAPTER 4

# CONCLUSION AND FUTURE WORK

We have presented a solution to categorize Stack Overflow questions using Bloom's Taxonomy. We have achieved this by integrating topic modelling and k-means clustering. By considering topic probabilities as feature vector, we have attained accuracy of 30.2%, which is better than random distribution. Also, we have considered more features like Answer Count, Score and View Count along with topic probabilities. This augmented feature vector gives us accuracy of 56.66%. So, we have successfully categorized questions using unsupervised method. So, we have learnt that it may be efficient to use output of topic modelling as a feature vector to perform clustering.

Also, we have analyzed that it's difficult to categorize each question of stack overflow according to Bloom's Taxonomy. Also, we have analyzed the problems associated with Bloom's Taxonomy. Not all questions can be assigned a discrete category. Moreover, since there is no absolute definition of each category, it may be difficult to assign a single category to each question. Hence, the categories are at times ambiguous.

Although, the precision of our technique is better than that of random distribution of categories, it can be increased by considering other feature vector like length of string. Moreover, sometimes it's difficult to assign a discrete category to the question. For such case, fuzzy clustering could be taken into consideration. One could also integrate sematic clustering along with the present model. Also, we can classify questions according to some other promising taxonomies.

# BIBLIOGRAPHY


[1] Bloom, Benjamin Samuel. "Taxonomy of educational objectives: The classification of educational goals ." (1956).

[2] Allamanis, Miltiadis, and Charles Sutton. "Why, when, and what: analyzing stack overflow questions by topic, type, and code." *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013.

[3] van Hoeij, Maggy JW, et al. "Developing a classification tool based on Bloom's taxonomy to assess the cognitive level of short essay questions."*Journal of veterinary medical education* 31.3 (2004): 261-267.

[4] Chang, Wen-Chih, and Ming-Shun Chung. "Automatic applying Bloom's taxonomy to classify and analysis the cognition level of English question items." *2009 Joint Conferences on Pervasive Computing (JCPC)*. 2009.

[5] Yusof, Norazah, and Chai Jing Hui. "Determination of Bloom's cognitive level of question items using artificial neural network." *2010 10th International Conference on Intelligent Systems Design and Applications*. IEEE, 2010.

[6] Haris, Syahidah Sufi, and Nazlia Omar. "BLOOM'S TAXONOMY QUESTION CATEGORIZATION USING RULES AND N-GRAM APPROACH." *Journal of Theoretical & Applied Information Technology* 76.3 (2015).

[7] Barua, Anton, Stephen W. Thomas, and Ahmed E. Hassan. "What are developers talking about? an analysis of topics and trends in stack overflow." *Empirical Software Engineering* 19.3 (2014): 619-654.

[8] Abduljabbar, Dhuha Abdulhadi, and Nazlia Omar. "EXAM QUESTIONS CLASSIFICATION BASED ON BLOOM'S TAXONOMY COGNITIVE LEVEL USING CLASSIFIERS COMBINATION." *Journal of Theoretical and Applied Information Technology* 78.3 (2015): 447.

[9] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Advances in neural information processing systems*. 2001.

[10] Scott, Terry. "Bloom's taxonomy applied to testing in computer science classes." *Journal of Computing Sciences in Colleges* 19.1 (2003): 267-274.

[11] Bloom, Benjamin Samuel. *Taxonomy of educational objectives*. Vol. 2. New York: Longmans, Green, 1964.

[12] Haris, Syahidah Sufi, and Nazlia Omar. "A rule-based approach in Bloom's Taxonomy question classification through natural language processing."*Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*. IEEE, 2012.

[13] Liu, Luying, et al. "A comparative study on unsupervised feature selection methods for text clustering." *2005 International Conference on Natural Language Processing and Knowledge Engineering*. IEEE, 2005.

[14] Slonim, Noam, and Naftali Tishby. "Document clustering using word clusters via the information bottleneck method." *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000.

[15] Owen, Sean, et al. "Mahout in action." (2012).

[16] Blei, David M. "Probabilistic topic models." *Communications of the ACM*55.4 (2012): 77-84.

[17] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.