CAMPUS PLACEMENT DATA ANALYTICS

MINOR PROJECT II

SUBMITTED BY:

PRABUDH KUMAR (20103246) B9
NEHAL JAIN (20103249) B9
KRITANK RISHI GOYAL (20103252) B9

UNDER THE SUPERVISION OF:

DR. NEETU SARDANA

May - 2023

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY UNIVERSITY, NOIDA

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **DR. NEETU SARDANA, Associate Professor**, Dept. of Computer Sciences, Jaypee Institute of Information Technology, India for her generous guidance, help and useful suggestions.

I also wish to extend my thanks to my classmates for their insightful comments and constructive suggestions to improve the quality of this project work. We express our sincere gratitude towards each and every one who helped us towards the completion of the project.

Name of students:

NEHAL JAIN (20103249)
KRITANK GOYAL (20103252)
PRABUDH KUMAR (20103246)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name: Prabudh Kumar
Enrolment No.: 20103246

Name: Nehal Jain
Enrolment No.:20103249

Name: Kritank Rishi Goyal
Enrolment No.:20103252

CERTIFICATE

This is to certify that the work titled "**Campus Placement Data Analytics**" submitted by Nehal Jain, Prabudh Kumar and Kritank Rishi Goyal of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Name of Supervisor:

DR. NEETU SARDANA

Designation:

Professor

Table of contents

ABSTRACT

Topic modeling has gained significant attention in various fields for uncovering latent themes or topics within a collection of documents. In the context of this project, topic modeling serves as a crucial tool for analyzing questions asked in interviews, encompassing both coding and theoretical domains. By identifying underlying topics, we can gain valuable insights into the distribution, frequency, and trends of different question categories. This knowledge aids in understanding the focus areas of companies like Amazon, Google, and Microsoft, enabling candidates to prepare more effectively for interviews.

The need for topic modeling arises from the vast amount of interview questions available, making manual analysis impractical and time-consuming. By automating the process, we can efficiently categorize and cluster questions based on their inherent topics, facilitating targeted analysis and examination of patterns. Through this project, we aim to provide a systematic approach for analyzing interview questions, enhancing the understanding of question distribution and assisting job seekers in their preparation.

Our approach involves collecting and preprocessing interview question data, applying topic modeling algorithms, and evaluating the results to gain meaningful insights. By utilizing topic modeling techniques, we can identify key themes, trends, and common topics across a wide range of interview questions. This analysis enables us to provide valuable information on the distribution and frequency of specific topics, aiding job seekers in focusing their preparation efforts and understanding the expectations of different companies.

SUMMARY

Topic modeling is a machine learning technique that automatically analyzes text data to determine cluster words for a set of documents.

We conducted an extensive analysis of coding questions and theoretical questions asked in interviews conducted by companies such as Amazon, Google, and Microsoft. The project involved the application of both supervised and unsupervised learning techniques to gain insights into these questions and provide valuable information for interview preparation.

In the supervised analysis, we utilized machine learning algorithms to classify coding questions based on relevant tags. We collected a dataset of coding problems and their associated tags, and trained various models:

- **Logistic Regression**
- **Decision Trees**
- **Naive Bayes**
- **KNN**
- **One-vs-Rest Classifier**
- **Random Forest**

to predict the presence of each tag in a given coding problem. We evaluated the performance of each model by calculating accuracy and compared their results.

For the unsupervised analysis, we focused on theoretical questions related to DBMS and networking. Using techniques such as

- **BERT**
- **LDA**
- **GSDMM**

We clustered these questions into potential topics. We evaluated the results using techniques like coherence to ensure their quality. We then used the BERT model to predict the topics matching the questions asked, and represented the frequency of these topics in each company.

Overall, this project aimed to provide valuable insights into the types of questions asked in interviews, enabling job seekers to better prepare for their interviews and focus on important topics. The data-driven analysis and visualizations facilitated decision-making and enhanced understanding of coding and theoretical concepts.

Chapter 1 : INTRODUCTION

1.1 General introduction:

In today's competitive job market, securing a position at top-tier companies like Amazon, Google, and Microsoft is a coveted goal for many job seekers. However, the interview processes at these renowned companies often pose significant challenges, particularly in terms of the coding and theory questions asked. Candidates face the daunting task of preparing for a wide range of technical questions, requiring a deep understanding of programming concepts and theoretical knowledge in areas such as database management systems (DBMS) and networking.
This project aims to address the existing gap in comprehensive analysis and accessible resources for job seekers preparing for interviews with Amazon, Google, and Microsoft. By providing detailed insights into the coding and theory questions asked in these interviews, the project aims to equip candidates with the necessary knowledge and preparation strategies to excel in the interview process.

The project consists of two distinct types of analysis: supervised analysis of coding questions and unsupervised analysis of theory questions on DBMS and networking. The supervised analysis involves the collection of coding problems asked in these companies, primarily sourced from platforms like LeetCode. Through scraping techniques, the problem statements and associated tags are extracted, enabling a thorough understanding of the skills and concepts assessed by these companies. This analysis provides candidates with a clearer understanding of the coding challenges they may encounter during interviews.

On the other hand, the unsupervised analysis focuses on theory questions in the domains of DBMS and networking. By utilizing advanced models such as BERT, LDA, and GSDMM, the project aims to cluster potential topics within the theory question dataset. This clustering facilitates the identification of common themes and areas of emphasis in interviews conducted by Amazon, Google, and Microsoft. By leveraging these unsupervised models, the project enhances the ability to predict relevant topics matching the questions asked and further provides insights into the frequency and importance of these topics in each company's interview process.

The significance of this project lies in its ability to empower job seekers with valuable insights and guidance during their interview preparation. By analyzing the questions asked in interviews conducted by top companies, candidates gain a comprehensive understanding of the specific skills, knowledge areas, and topics that are crucial for success. Armed with this knowledge, candidates can tailor their preparation efforts, focusing on the most relevant concepts and improving their chances of performing well in the interviews.

In conclusion, this project seeks to bridge the gap between job seekers and interview success by providing a comprehensive analysis of coding and theory questions asked in interviews conducted by Amazon, Google, and Microsoft. By understanding the specific requirements of these companies and leveraging both supervised and unsupervised analysis techniques, the project aims to equip candidates with the necessary tools to excel in their interviews and secure coveted positions at these top-tier organizations.

1.2 Problem statement:

Job seekers preparing for interviews with prestigious companies such as Amazon, Google, and Microsoft often face a significant challenge - the lack of comprehensive and easily accessible information on the coding and theory questions asked in these interviews. The absence of structured analysis and insights into the specific types of questions and topics that are commonly encountered creates uncertainty and makes it difficult for candidates to focus their preparation effectively. This knowledge gap hinders their ability to prioritize relevant areas of study, potentially leading to inadequate preparation and decreased chances of success in the interviews. As a result, there is a pressing need for a systematic approach that provides a detailed analysis of interview questions, equipping job seekers with the necessary guidance and resources to enhance their interview performance. By addressing this problem, the project aims to bridge the information gap, ensuring that candidates have access to comprehensive and accurate insights that significantly improve their preparation and increase their chances of succeeding in interviews conducted by these top companies.

1.3 Significance of this project:

The significance of this project lies in its potential to empower job seekers with invaluable guidance and resources for interviews with prominent companies like Amazon, Google, and Microsoft. By conducting a comprehensive analysis of the coding and theory questions asked in these interviews, the project fills the knowledge gap and provides candidates with specific insights into the skills, knowledge areas, and topics that are essential for success. This detailed analysis allows candidates to tailor their preparation strategies, focus on relevant concepts, and allocate their study time more effectively. Ultimately, the project serves as a valuable resource that enhances the confidence and performance of job seekers, enabling them to stand out in the highly competitive interview processes of these top companies. With access to comprehensive information and insights, candidates are better equipped to excel in their interviews, increasing their chances of securing desirable job opportunities.

Chapter 2: LITERATURE REVIEW

2.1 Summary of papers studied:

1. Analyzing BERT for Question-Answering: What Works, What Doesn't, and What Matters? (2019):

   This research paper focuses on analyzing the performance of BERT, a popular language model, in question-answering tasks. The authors investigate various components of BERT, including input representation, fine-tuning techniques, and model architectures. They conduct a comparative analysis to identify the strengths and weaknesses of BERT in question-answering. By examining benchmark datasets, they identify areas where BERT performs well and areas where it falls short. The paper provides valuable insights into the factors that contribute to the success or limitations of BERT in question-answering tasks. The authors suggest future work should explore alternative fine-tuning methods, different BERT variants, and novel model architectures to further enhance BERT's performance.

2. An Analysis of BERT Performance for Reading Comprehension (2020):

   This paper focuses on evaluating the performance of BERT, a pre-trained language model, in reading comprehension tasks. The authors analyze the limitations and failure cases of BERT when applied to reading comprehension datasets. They identify scenarios where BERT struggles to provide accurate answers and propose methods to improve its performance by incorporating additional context and external knowledge sources. The research highlights the need to address the challenges faced by BERT in reading comprehension and suggests exploring ways to handle out-of-domain questions. The paper provides valuable insights into BERT's strengths and weaknesses in reading comprehension tasks and presents potential avenues for enhancing its performance.

3. A Comparison of Machine Learning Algorithms for Student Success Prediction (2017)

   In this study, the authors compare different machine learning algorithms for predicting student success. They explore the performance of logistic regression, decision trees,

random forests, support vector machines, and neural networks using student records as the dataset. The paper presents a comparative analysis of these algorithms, considering evaluation metrics to assess their performance. The results provide insights into the effectiveness of each algorithm for student success prediction. The study suggests future work should involve exploring additional algorithms, incorporating feature engineering techniques, and investigating the interpretability of prediction models. The research aims to contribute to the development of effective student success prediction systems, which can assist in educational decision-making processes.

4. Topic Modelling for Short Text (Mazarura, Jocelyn Rangarirai):
   This research paper focuses on analyzing short text using topic modeling techniques. The author specifically applies Latent Dirichlet Allocation (LDA), a popular topic modeling algorithm, to identify topics in short text documents. By employing LDA, the study successfully extracts meaningful topics from short text data. However, the paper acknowledges the need for further investigation into more advanced topic modeling techniques, such as Dynamic Topic Models (DTM), to improve the analysis of short text. The research highlights the significance of topic modeling in understanding the underlying themes in short textual data, which can have applications in various domains, including education.

5. Analyzing Interview Data: The Development and Evolution of a Coding System:
   This paper discusses the development and application of a coding system for analyzing interview data. The author presents a systematic approach to analyze interview transcripts using a coding system. By employing this coding system, the study enhances the understanding of interview data. The research focuses on the development and refinement of the coding system, leading to a more structured analysis of interview data. The paper suggests future work should involve applying the coding system to additional interviews, exploring inter-coder reliability, and further refining the coding system. The research contributes to the field of qualitative analysis by providing insights into the systematic analysis of interview data and highlights the importance of using coding systems in interview-based research studies.

2.2 Background study:

ID	Paper Name	Authors	Objective	Dataset Used	Techniques Applied	Performance	Results	Future Work
1	Analyzing BERT for Question-Answering: What Works, What Doesn't, and What Matters? (2019)	Rishi Gupta, Bhavitvya Malik, Anirudh Joshi, Radhika Mamidi	Analyze the performance of BERT in question-answering tasks	Benchmark datasets	Investigation of different BERT components (input representation, fine-tuning techniques, model architectures)	Comparative analysis of BERT's strengths and weaknesses	Identified areas where BERT performs well and where it falls short	Further exploration of BERT variants, alternative fine-tuning methods, and model architectures
2	An Analysis of BERT Performance for Reading Comprehension (2020)	Nitish Gupta, Raghav Gupta, Shobhit Bansal, Dan Jurafsky	Examine BERT's performance in reading comprehension tasks	Reading comprehension datasets	Evaluation of BERT's limitations and failure cases	Identified scenarios where BERT struggles to provide accurate answers	Proposed methods to improve BERT's performance through additional context and external knowledge sources	Investigate the impact of incorporating external knowledge sources, explore ways to handle out-of-domain questions
3	A Comparison of Machine Learning Algorithms for Student Success Prediction (2017)	Miao Yu, Kalyan Moy Gupta, Qiang Yang, Tanveer A. Faruquie	Compare machine learning algorithms for student success prediction	Student records	Logistic regression, decision trees, random forests, support vector machines, neural networks	Comparison of algorithm performance using evaluation metrics	Comparative analysis of machine learning algorithm performance for student success prediction	Explore the use of additional algorithms, consider feature engineering techniques, and investigate interpretability of prediction models

4	Topic Modelling for Short Text (2015)	Jocelyn Rangarirai Mazarura	Analyze short text using topic modeling techniques	Short text dataset	Latent Dirichlet Allocation (LDA)	Identification of topics in short text documents	Extracted meaningful topics from short text data	Investigate the use of more advanced topic modeling techniques, such as Dynamic Topic Models (DTM)
5	Analyzing Interview Data: The Development and Evolution of a Coding System (2001)	Weston, C., Gandell, T., Beauchamp	Analyze interview data using a coding system	Interview transcripts	Coding system development and application	Enhanced understanding of interview data through systematic coding	Developed and refined a coding system for interview data analysis	Apply the coding system to additional interviews, explore inter-coder reliability, and further refine the coding system

Chapter 3: PROPOSED METHODOLOGY FOR TOPIC MODELING FOR PLACEMENT DATA

TOPIC DETECTION OF CODING BASED QUESTIONS USING SUPERVISED CLASSIFICATION

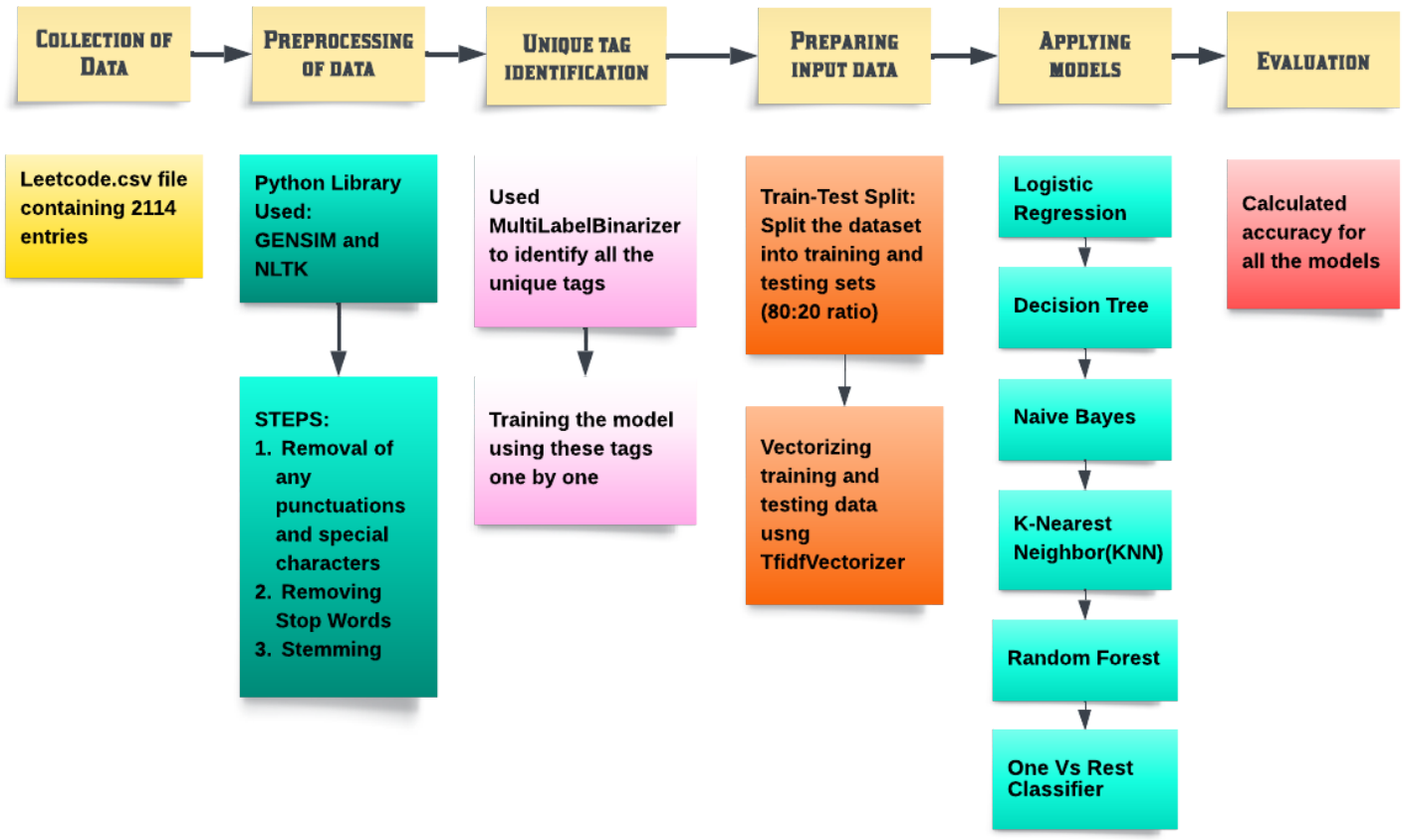


Fig 3.1: TOPIC DETECTION OF CODING BASED QUESTIONS USING SUPERVISED CLASSIFICATION

TOPIC DETECTION OF THEORETICAL QUESTIONS USING UNSUPERVISED CLASSIFICATION

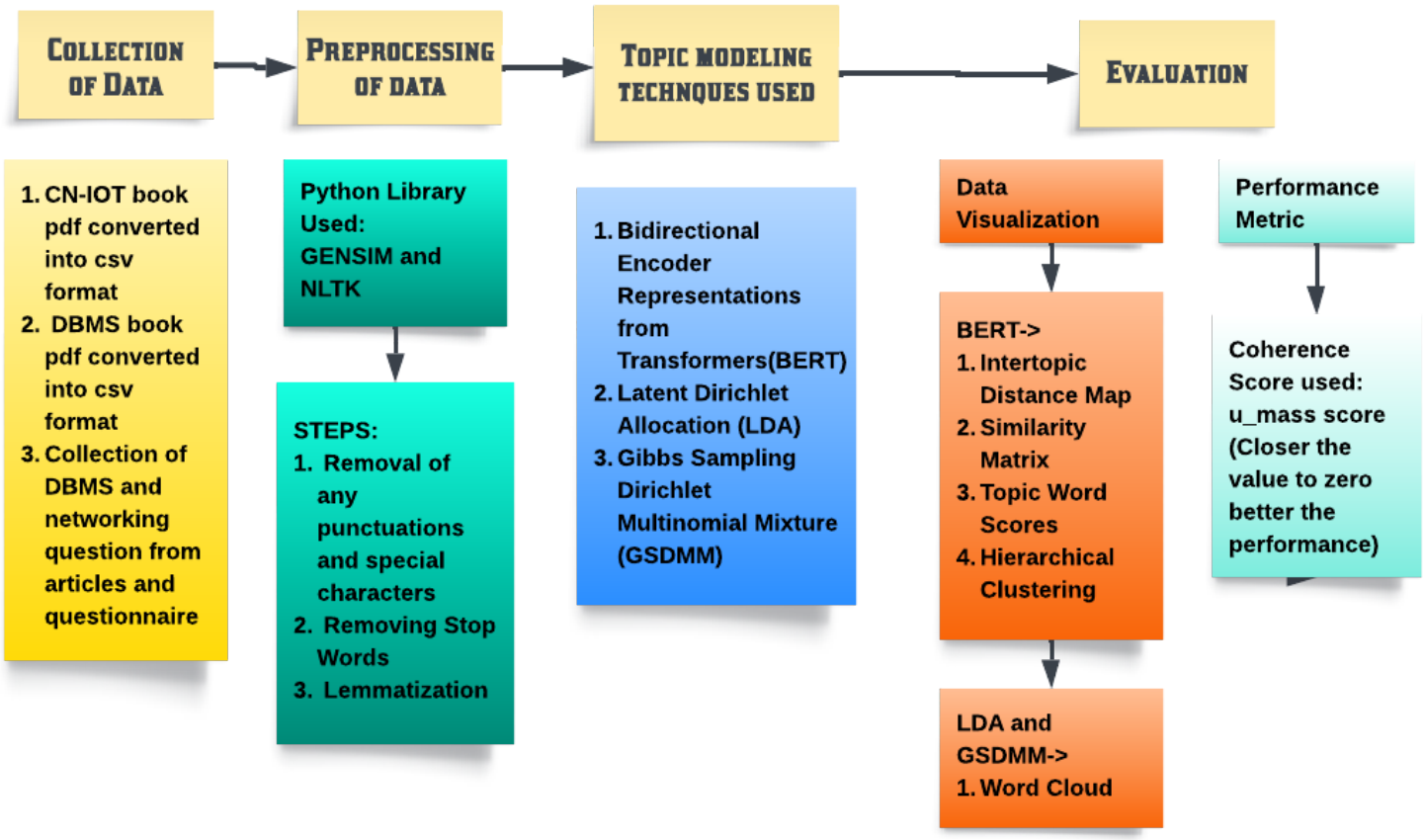


Fig 3.2: TOPIC DETECTION OF THEORETICAL QUESTIONS USING UNSUPERVISED CLASSIFICATION

Chapter 4: Requirement Analysis and Theory of Approaches

4.1 Requirement Analysis

Language Used:

- Python 3

Libraries to be Used:

- pandas
- numpy
- gensim
- nltk
- matplotlib
- bertopic
- umap
- wordcloud
- Torch
- MultiLabelBinarizer
- CountVectorizer
- LogisticRegression
- TfidfVectorizer
- DecisionTreeClassifier
- MultinomialNB

Functional requirements:

1. Data Acquisition:

- ➔ The system should be able to download a file containing links to coding problems asked in the targeted companies.
- ➔ The system should be capable of using the "leetscrape" library to scrape problem statements and associated tags from the provided links.
- ➔ The system should be able to collect and store the scraped data in a CSV format.

2. Data Processing and Analysis:

- ➔ The system should preprocess the collected data by cleaning, removing stop words, and stemming the text.
- ➔ The system should apply the MultiLabelBinarizer technique to deal with the multi-label nature of the tags.
- ➔ The system should convert the text data into numerical vectors using techniques like CountVectorizer and TF-IDF vectorization.
- ➔ The system should split the dataset into training and testing sets for model evaluation.
- ➔ The system should apply various classification algorithms such as Logistic Regression, Decision Trees, Naive Bayes, KNN, and One-vs-Rest Classifier on each class/tag to predict their presence in coding questions.
- ➔ The system should calculate and report the accuracy of each classifier.

3. Unsupervised Analysis:

- ➔ The system should apply different unsupervised models such as BERT, LDA, and GSDMM to analyze the theoretical questions on DBMS and networking.
- ➔ The system should create clusters of potential topics using these unsupervised models.
- ➔ The system should evaluate the clustering results using techniques like coherence analysis.
- ➔ The system should use the trained BERT model to predict the topics matching the questions asked.

➔ The system should represent the predicted topics and their frequency in the particular company.

4. **Visualization and Reporting:**

➔ The system should generate word clouds for each topic to visually represent their frequency and importance.

➔ The system should generate plots and visualizations to provide an analysis of the topics and their distribution.

➔ The system should generate reports or summaries of the findings from the analysis for presentation purposes.

Non functional requirements:

1. **Performance:**
   - The system should be able to handle large datasets efficiently and process them within a reasonable time frame.
   - The data processing and analysis tasks should be performed in a timely manner to provide quick insights and results.
2. **Accuracy:**
   - The classification models used in the supervised analysis should exhibit high accuracy in predicting the presence of tags in coding questions.
   - The unsupervised models should generate meaningful clusters and accurately predict topics matching the questions asked.
3. **Scalability:**
   - The system should be scalable to accommodate an increasing number of coding questions and theoretical questions.
   - It should be able to handle additional companies or data sources without significant modifications to the system.
4. **Reliability:**
   - The system should be reliable and robust, handling exceptions and errors gracefully.

- It should handle unexpected input data or missing values appropriately, providing informative error messages or fallback mechanisms.

4.2 Theory of Approaches:

Supervised Classification

- **Logistic Regression:**

  Logistic regression is a popular linear classification algorithm. It models the relationship between the independent variables and the dependent variable using the logistic function. The logistic function maps the linear combination of the input features to a value between 0 and 1, representing the probability of the instance belonging to a certain class. During training, logistic regression learns the optimal weights for the independent variables by maximizing the likelihood function. In prediction, the model calculates the probabilities and assigns the instance to the class with the highest probability.

  Working:

  ➔ Calculate the weighted sum of the input features and weights.
  ➔ Apply the logistic function (sigmoid) to the weighted sum to obtain the predicted probability.
  ➔ Assign a class label based on a decision threshold (e.g., 0.5). If the probability is above the threshold, classify the instance as one class; otherwise, classify it as the other class.

- **K-Nearest Neighbors (KNN):**

  K-Nearest Neighbors (KNN) is a non-parametric and instance-based classification algorithm. It classifies new instances based on the majority vote of its K nearest neighbors in the feature space. KNN assumes that instances belonging to the same class are close to each other.

  Working:

- ➔ Calculate the distance between the new instance and all training instances using a distance metric such as Euclidean distance.
- ➔ Select the K nearest neighbors with the shortest distances.
- ➔ Assign the class label to the new instance based on the majority vote of the K neighbors. If K=1, the new instance is assigned the label of the nearest neighbor.

- **Naive Bayes:**

  Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with an assumption of independence among the features. It calculates the probability of an instance belonging to a certain class based on the joint probability of the features given the class.

  Working:

  - ➔ Estimate the prior probability of each class from the training data.
  - ➔ Estimate the conditional probability of each feature given the class.
  - ➔ For a new instance, calculate the posterior probability for each class using Bayes' theorem.
  - ➔ Assign the class label with the highest posterior probability to the new instance.

- **Decision Tree:**

  A decision tree is a flowchart-like structure where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a decision. The tree is constructed recursively by selecting the best attribute to split the data based on certain criteria.

  Working:

  - ➔ Select the best attribute to split the data based on a criterion such as information gain or Gini index.
  - ➔ Split the data into subsets based on the attribute values.
  - ➔ Repeat the process recursively for each subset until a stopping condition is met (e.g., reaching a maximum depth or purity).

- ➔ Assign the majority class label of the instances in a leaf node or make a decision based on the leaf node's value during prediction.

- **Random Forest:**

  Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It reduces the variance and overfitting of individual decision trees by introducing randomness.

  Working:

  - ➔ Create an ensemble of decision trees, each trained on a random subset of the training data (bootstrap aggregating or "bagging") and a random subset of features.
  - ➔ During prediction, each tree provides a class label, and the majority vote of the trees determines the final prediction.

- **One vs Rest Classifier:**

  One vs Rest (OvR) classifier, also known as one vs all, is a technique to extend binary classification algorithms to multi-class problems. It trains multiple binary classifiers, where each class is treated as the positive class, and all other classes are treated as the negative class.

  Working:

  - ➔ For each class in the dataset, train a binary classifier to distinguish instances of that class from the instances of all other classes.
  - ➔ During prediction, apply each binary classifier to the input instance and obtain a probability or confidence score for each class.
  - ➔ Assign the class with the highest probability or confidence as the predicted class for the input instance.

Unsupervised Classification:

1. **Bidirectional Encoder Representations from Transformers(BERT):**

   BERT is a pre-trained deep learning model that utilizes a transformer architecture to learn contextualized word representations from large amounts of unlabeled text data. It employs a bidirectional approach, allowing the model to capture the context from both left and right contexts of a word.

   Working:

   The training process of BERT involves two key steps: pre-training and fine-tuning.

   1. Pre-training: BERT is pre-trained on a large corpus of unlabeled text data, such as books, Wikipedia, and web pages. During pre-training, BERT learns to predict masked words in a sentence and understand the relationship between different words. It also learns to differentiate between sentences in a text corpus.
   2. Fine-tuning: After pre-training, BERT is further fine-tuned on specific downstream tasks with labeled data. During fine-tuning, BERT is trained on task-specific datasets, such as sentiment analysis or named entity recognition, to adapt its pre-trained representations for the specific task.

   The architecture of BERT consists of multiple transformer layers. These layers use self-attention mechanisms to capture the relationships between words in a sentence, resulting in contextualized word embeddings that consider the entire sentence context. The contextualized embeddings are then used for various downstream NLP tasks.

2. **Latent dirichlet allocation (LDA):**

Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique that seeks to find a linear combination of features that maximizes the separation between classes. It is commonly used for pattern recognition, feature extraction, and classification tasks.
The goal of LDA is to project high-dimensional data onto a lower-dimensional space while preserving the discriminatory information between classes. It achieves this by maximizing the

between-class scatter while minimizing the within-class scatter. The steps involved in LDA are as follows:

1. Compute the mean vectors for each class in the dataset.
2. Compute the scatter matrices: within-class scatter matrix (Sw) and between-class scatter matrix (Sb).
3. Solve the generalized eigenvalue problem Sw^(-1)Sb to obtain the eigenvectors and eigenvalues.
4. Select the top k eigenvectors corresponding to the largest eigenvalues to form the transformation matrix.
5. Project the data onto the new subspace defined by the selected eigenvectors.

3. GSDMM (Gibbs Sampling Dirichlet Multinomial Mixture Model) is a probabilistic clustering algorithm that is particularly useful for text data. It is based on the Dirichlet Multinomial Mixture Model and utilizes Gibbs sampling to infer the underlying latent clusters in the data.

Theory:

The GSDMM algorithm assumes that each document in the dataset belongs to a specific latent cluster. It assigns each document to a cluster by iteratively sampling the cluster assignments based on the posterior distribution of the cluster membership probabilities. The steps involved in GSDMM are as follows:

1. Initialize the number of clusters and set initial cluster assignments for each document.
2. Iterate through each document and update its cluster assignment by sampling from the posterior distribution of the cluster membership probabilities.
3. After sampling all the documents, update the cluster parameters based on the new cluster assignments.
4. Repeat steps 2 and 3 until convergence, which is typically determined by a maximum number of iterations or a convergence threshold.

Chapter 5: IMPLEMENTATION

5.1 Data collection:

- Supervised Data Collection:

We downloaded a file containing links to coding problems asked in the targeted companies.
We used LeetCode's "leetscrape" library to scrape the problem statements and tags from the provided links.
We collected and stored the data in a CSV format.

- Unsupervised Data Collection:

We gathered theoretical questions related to database management systems (DBMS) and networking from relevant books and resources.
We used these collected questions as the unsupervised data for analysis.

Supervised Classification:

LeetScrape_Output_CSV

	titleSlug	title	difficulty	acceptance	paidOnly	topicTags	categorySlug
2	two-sum	Two Sum	Easy	49.74232	FALSE	array,hash-table	algorithms
3	add-two-numbers	Add Two Numbers	Medium	40.401967	FALSE	linked-list,math,recursi	algorithms
4	longest-substring-without-rep	Longest Substring Wit	Medium	33.83664	FALSE	hash-table,string,sliding	algorithms
5	median-of-two-sorted-arrays	Median of Two Sorte	Hard	36.273797	FALSE	array,binary-search,div	algorithms
6	longest-palindromic-substring	Longest Palindromic S	Medium	32.414763	FALSE	string,dynamic-progran	algorithms
7	zigzag-conversion	Zigzag Conversion	Medium	44.934454	FALSE	string	algorithms
8	reverse-integer	Reverse Integer	Medium	27.469104	FALSE	math	algorithms
9	string-to-integer-atoi	String to Integer (atoi)	Medium	16.620702	FALSE	string	algorithms
10	palindrome-number	Palindrome Number	Easy	53.57827	FALSE	math	algorithms
11	regular-expression-matching	Regular Expression M	Hard	28.016197	FALSE	string,dynamic-progran	algorithms
12	container-with-most-water	Container With Most	Medium	53.997143	FALSE	array,two-pointers,gre	algorithms
13	integer-to-roman	Integer to Roman	Medium	62.071892	FALSE	hash-table,math,string	algorithms
14	roman-to-integer	Roman to Integer	Easy	58.592014	FALSE	hash-table,math,string	algorithms
15	longest-common-prefix	Longest Common Pre	Easy	40.885865	FALSE	string,trie	algorithms
16	3sum	3Sum	Medium	32.617058	FALSE	array,two-pointers,sort	algorithms
17	3sum-closest	3Sum Closest	Medium	45.719728	FALSE	array,two-pointers,sort	algorithms
18	letter-combinations-of-a-pho	Letter Combinations	Medium	56.619536	FALSE	hash-table,string,backt	algorithms
19	4sum	4Sum	Medium	35.876693	FALSE	array,two-pointers,sort	algorithms
20	remove-nth-node-from-end-	Remove Nth Node Fr	Medium	41.157066	FALSE	linked-list,two-pointers	algorithms

After extraction we got leetcode.csv

```
Number of rows in data = 2114
Number of columns in data = 4
```

Sample data:

Out[5]:

	problem_statement	titleslug	title	tags
0	Given an array of integers nums and an integer...	two-sum	Two Sum	Array, Hash Table
1	You are given two non-empty linked lists repre...	add-two-numbers	Add Two Numbers	Linked List, Math, Recursion
2	Given a string s, find the length of the longe...	longest-substring-without-repeating-characters	Longest Substring Without Repeating Characters	Hash Table, String, Sliding Window
3	Given two sorted arrays nums1 and nums2 of siz...	median-of-two-sorted-arrays	Median of Two Sorted Arrays	Array, Binary Search, Divide and Conquer
4	Given a string s, return the longest palindrom...	longest-palindromic-substring	Longest Palindromic Substring	String, Dynamic Programming
5	The string "PAYPALISHIRING" is written in a zi...	zigzag-conversion	Zigzag Conversion	String
6	Given a signed 32-bit integer x, return x with...	reverse-integer	Reverse Integer	Math
7	Implement the myAtoi(string s) function, which...	string-to-integer-atoi	String to Integer (atoi)	String
8	Given an integer x, return true if x is a pali...	palindrome-number	Palindrome Number	Math
9	Given an input string s and a pattern p, imple...	regular-expression-matching	Regular Expression Matching	String, Dynamic Programming, Recursion

Unsupervised Classification:

Computer Networks Data loaded from the book - “Andrew S. Tanenbaum - Computer Networks”:

Out[17]:

	review
0	The th century was the era of the great mecha...
1	The th century was the age of the steam engine
2	During the th century the key technology was...
3	Among other developments we saw the installat...
4	As a result of rapid technological progress t...
5	Organizations with hundreds of offices spread ...
6	As our ability to gather process and distrib...
7	Although the computer industry is still young ...
8	automobiles and air transportation) computers...
9	During the first two decades of their existenc...

DBMS data loaded from the book - “Fundamentals_of_Database_Systems_6th_Edition-1”:

Out[14]:

	review
0	Databases and Database Users Databases and dat...
1	For example if we go to the bank to deposit o...
2	Even purchasing items at a supermarket often ...
3	These interactions are examples of what we ma...
4	In the past few years advances in technology ...
5	New media technology has made it possible to ...
6	These types of files are becoming an importan...
7	Geographic information systems can store and ...
8	Data warehouses and online analytical process...
9	Real time and active database technology is u...

5.2 Data Preprocessing:

One of the most critical steps in any NLP-related project is column-wise preprocessing of all documents. While there are some standard cleaning methods that, in general, can be applied to any corpus, there are also a considerable amount of intuition-driven decisions that we must make.

We have performed the following steps:

- Tokenization: Split the text into sentences and the sentences into words. Lowercase the words, remove punctuation and special characters.
- All stopwords are removed.
- Words are lemmatized — words in third person are changed to first person and verbs in past and future tenses are changed into present.
- Stemming

Supervised Classification: Lowered the text, Clean HTML, Punctuation, Stopwords Removal, Stemming

Out[126]:

	problem_statement	titleslug	title	tags	Array	String	Hash Table	Dynamic Programming	Math	Tree	...	Heap (Priority Queue)	Backtracking	Simulation	Prefix Sum	Gr
0	given array integ num integ target return indi...	two-sum	Two Sum	['Array', 'Hash Table']	1	0	1	0	0	0	...	0	0	0	0	
1	given non empti link list repres non negat int...	add-two-numbers	Add Two Numbers	['Linked List', 'Math', 'Backtracking']	0	0	0	0	1	0	...	0	1	0	0	
2	given string find length longest substr withou...	longest-substring-without-repeating-characters	Longest Substring Without Repeating Characters	['Hash Table', 'String', 'Sliding Window']	0	1	1	0	0	0	...	0	0	0	0	
3	given sort array num num size n respect return...	median-of-two-sorted-arrays	Median of Two Sorted Arrays	['Array', 'Binary Search', 'Divide and Conquer']	1	0	0	0	0	0	...	0	0	0	0	
4	given string return longest palindrom substr s	longest-palindromic-substring	Longest Palindromic Substring	['String', 'Dynamic Programming']	0	1	0	1	0	0	...	0	0	0	0	

5 rows × 29 columns

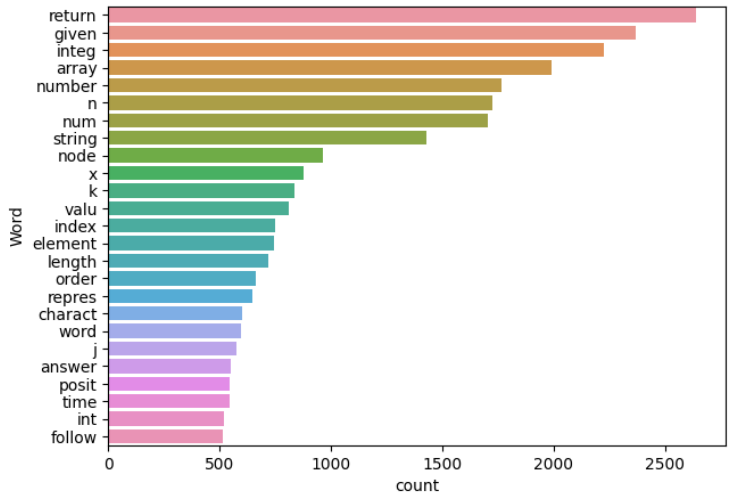


Fig 5.1: most frequent words

Unsupervised Classification: Gensim Simple preprocessing(lower, removing unwanted symbols), Stopwords Removal, Lemmatized

Computer Networks data after preprocessing:

```
In [23]: content.head(10)
```

Out[23]:

	review
0	th century era great mechanical system accompa...
1	th century age steam engine
2	th century key technology information gatherin...
3	development saw installation worldwide telepho...
4	result rapid technological progress area rapid...
5	organization hundred office spread wide geogra...
6	ability gather process distribute information ...
7	industry young compared industry
8	automobile air transportation computer spectac...
9	decade existence system highly centralized usu...

DBMS data after preprocessing:

```
In [17]: content.head(10)
```

Out[17]:

	review
0	database database user database database syste...
1	example bank deposit withdraw fund hotel airli...
2	purchasing item supermarket automatically upda...
3	interaction example traditional database appli...
4	past year advance technology led exciting new ...
5	new medium technology possible store image aud...
6	type file important component multimedia database
7	geographic information system store analyze ma...
8	data warehouse online analytical processing sy...
9	real time active database technology control i...

5.3 Application of algorithms:
Supervised Classification:

Distribution_of_tags

```
Array    :    1150
String    :    519
Hash Table    :    389
Math    :    371
Dynamic Programming    :    366
Sorting    :    271
Greedy    :    263
Depth-First Search    :    204
Binary Search    :    191
Breadth-First Search    :    173
Matrix    :    160
Tree    :    157
Bit Manipulation    :    143
Two Pointers    :    142
Binary Tree    :    126
Heap (Priority Queue)    :    122
Stack    :    116
Prefix Sum    :    107
Simulation    :    107
Graph    :    95
Counting    :    89
Design    :    87
Backtracking    :    78
Database    :    77
Sliding Window    :    72
```

Splitting data for training and testing (80% : 20%)

2.4. Train-Test Split

In [127]:
```
from sklearn.model_selection import train_test_split

train, test = train_test_split(data, random_state=42, test_size=0.20, shuffle=True)

print(train.shape)
print(test.shape)
```

```
(1691, 29)
(423, 29)
```

Vectorised the data before fitting it into the model:

2.5. TF-IDF

In [33]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(strip_accents='unicode', analyzer='word', ngram_range=(1,3), norm='l2')
vectorizer.fit(train_text)
vectorizer.fit(test_text)
```

Out[33]:
```
TfidfVectorizer(ngram_range=(1, 3), strip_accents='unicode')
```

In [34]:
```
x_train = vectorizer.transform(train_text)
y_train = train['Array']

x_test = vectorizer.transform(test_text)
y_test = test['Array']
print(x_test)
```

```
  (0, 26408)    0.06364060260998565
  (0, 26407)    0.06364060260998565
  (0, 26375)    0.04058767820262016
  (0, 25909)    0.06364060260998565
  (0, 25908)    0.06364060260998565
  (0, 25757)    0.02735457878022888
  (0, 25727)    0.06364060260998565
  (0, 25726)    0.1272812052199713
  (0, 25725)    0.17874355237853165
  (0, 25673)    0.10672211183052387
  (0, 24345)    0.06364060260998565
  (0, 24344)    0.06364060260998565
  (0, 24343)    0.06364060260998565
  (0, 22516)    0.06364060260998565
  (0, 22512)    0.05670098087559433
  (0, 22446)    0.03203347882366712
  (0, 21003)    0.06364060260998565
  (0, 21002)    0.06364060260998565
  (0, 20981)    0.04415862137672426
  (0, 20650)    0.06364060260998565
  (0, 20649)    0.06364060260998565
  (0, 20535)    0.03182269718861194
  (0, 20297)    0.06364060260998565
  (0, 20296)    0.06364060260998565
  (0, 20266)    0.04009920289291582
  :     :
```

Unsupervised Classification

1. BERT:

Got the clusters/list of topics after applying BERT on the preprocessed data

Computer Networks:

```
In [26]: topic_model.get_topic_info()
```

Out[26]:

	Topic	Count	Name
0	-1	9117	-1_time_bit_packet_error
1	0	609	0_telephone_phone_mobile_voice
2	1	248	1_dns_domain_record_query
3	2	199	2_site_web_search_unpopular
4	3	189	3_band_ghz_frequency_spectrum
...	...	...	...
328	327	10	327_let_unrealistic_example_concrete
329	328	10	328_fairness_efficiency_defining_min
330	329	10	329_byte_tcp_entity_reconstructs
331	330	10	330_nics_ethernet_address_unique
332	331	10	331_queue_queued_quietly_tail

333 rows × 3 columns

DBMS:

```
In [20]: topic_model.get_topic_info()
```

Out[20]:

	Topic	Count	Name
0	-1	4858	-1_attribute_relation_employee_database
1	0	292	0_diagram_uml_notation_rose
2	1	268	1_index_field_block_file
3	2	203	2_dbms_software_application_ibm
4	3	186	3_relationship_entity_binary_ratio
...	...	...	...
217	216	10	216_spatial_essentially_city_join
218	217	10	217_engineer_technician_secretary_hourlyemployee
219	218	10	218_gpa_science_majoring_major
220	219	10	219_collection_related_implicit_coherent
221	220	10	220_project_involve_manager_worker

222 rows × 3 columns

2. Latent Dirichlet Allocation (LDA):

Computer Networks:

LDA made the numbers of cluster of similar words as specified for networking

```
(0, '0.242*"mean" + 0.216*"reason" + 0.081*"demand" + 0.065*"frequency" + 0.052*"upload"')
(1, '0.424*"new" + 0.070*"purpose" + 0.067*"part" + 0.050*"maintain" + 0.045*"construct"')
(2, '0.445*"server" + 0.076*"good" + 0.074*"small" + 0.054*"think" + 0.045*"play"')
(3, '0.209*"record" + 0.123*"problems" + 0.089*"help" + 0.088*"apply" + 0.080*"view"')
(4, '0.202*"lose" + 0.128*"transmission" + 0.086*"appear" + 0.078*"queue" + 0.058*"transform"')
(5, '0.300*"authentication" + 0.151*"place" + 0.124*"home" + 0.052*"mobile" + 0.041*"location"')
(6, '0.233*"protocol" + 0.146*"program" + 0.130*"type" + 0.123*"packet" + 0.059*"define"')
(7, '0.163*"look" + 0.144*"way" + 0.130*"include" + 0.118*"like" + 0.086*"connections"')
(8, '0.173*"happen" + 0.148*"see" + 0.107*"select" + 0.092*"best" + 0.068*"examples"')
(9, '0.320*"public" + 0.087*"real" + 0.087*"reduce" + 0.076*"download" + 0.055*"consist"')
(10, '0.182*"algorithm" + 0.154*"size" + 0.087*"rfc" + 0.076*"fetch" + 0.063*"parameters"')
(11, '0.168*"give" + 0.129*"servers" + 0.122*"tell" + 0.087*"carry" + 0.080*"word"')
(12, '0.311*"frame" + 0.229*"send" + 0.055*"successor" + 0.043*"versions" + 0.032*"style"')
(13, '0.240*"allow" + 0.194*"base" + 0.060*"use" + 0.055*"authenticate" + 0.055*"additional"')
(14, '0.322*"fig" + 0.190*"show" + 0.129*"media" + 0.122*"form" + 0.082*"group"')
(15, '0.414*"address" + 0.122*"sit" + 0.101*"private" + 0.050*"range" + 0.040*"represent"')
(16, '0.244*"design" + 0.105*"character" + 0.079*"procedure" + 0.073*"expect" + 0.071*"current"')
(17, '0.196*"site" + 0.165*"return" + 0.074*"modify" + 0.065*"acknowledgement" + 0.062*"acknowledge"')
(18, '0.459*"web" + 0.080*"domain" + 0.056*"talk" + 0.048*"simply" + 0.035*"person"')
(19, '0.193*"video" + 0.184*"set" + 0.133*"audio" + 0.105*"add" + 0.081*"space"')
(20, '0.173*"different" + 0.136*"problem" + 0.117*"control" + 0.104*"multiple" + 0.077*"later"')
(21, '0.134*"simple" + 0.120*"flow" + 0.108*"kb" + 0.103*"main" + 0.078*"software"')
(22, '0.214*"break" + 0.189*"input" + 0.128*"transmit" + 0.067*"error" + 0.049*"decide"')
(23, '0.173*"applications" + 0.150*"read" + 0.096*"channel" + 0.093*"class" + 0.078*"yes"')
(24, '0.429*"internet" + 0.125*"clients" + 0.040*"network" + 0.037*"area" + 0.035*"architecture"')
(25, '0.201*"window" + 0.184*"suppose" + 0.174*"reply" + 0.071*"kinds" + 0.054*"send"')
```

DBMS:

LDA made the numbers of cluster of similar words as specified for DBMS

```
(0, '0.322*"discuss" + 0.226*"section" + 0.092*"relate" + 0.044*"concepts" + 0.037*"type"')
(1, '0.090*"ssn" + 0.067*"employee" + 0.057*"workson" + 0.050*"sex" + 0.041*"figure"')
(2, '0.154*"relations" + 0.106*"introduce" + 0.082*"usually" + 0.060*"small" + 0.056*"ways"')
(3, '0.121*"classification" + 0.108*"salary" + 0.095*"employee" + 0.083*"encryption" + 0.061*"average"'
(4, '0.197*"item" + 0.104*"size" + 0.068*"server" + 0.066*"web" + 0.057*"depend"')
(5, '0.297*"access" + 0.144*"web" + 0.132*"base" + 0.041*"have" + 0.041*"limit"')
(6, '0.128*"multiple" + 0.086*"problems" + 0.072*"multimedia" + 0.065*"store" + 0.054*"enforce"')
(7, '0.409*"rule" + 0.197*"different" + 0.064*"determine" + 0.035*"appropriate" + 0.032*"represent"')
(8, '0.281*"update" + 0.078*"possible" + 0.062*"produce" + 0.050*"modify" + 0.048*"copy"')
(9, '0.180*"language" + 0.146*"program" + 0.117*"associate" + 0.050*"data" + 0.041*"query"')
(10, '0.108*"element" + 0.096*"directly" + 0.088*"command" + 0.051*"difference" + 0.042*"set"')
(11, '0.195*"mean" + 0.111*"properties" + 0.049*"linear" + 0.046*"automatically" + 0.042*"explain"')
(12, '0.091*"certain" + 0.080*"particular" + 0.068*"applications" + 0.058*"data" + 0.056*"organization'
(13, '0.233*"disk" + 0.133*"memory" + 0.124*"main" + 0.057*"construct" + 0.046*"higher"')
(14, '0.313*"type" + 0.125*"specific" + 0.062*"mechanism" + 0.055*"define" + 0.037*"relationship"')
(15, '0.154*"cluster" + 0.142*"control" + 0.140*"department" + 0.072*"work" + 0.069*"project"')
(16, '0.147*"refer" + 0.102*"new" + 0.093*"list" + 0.077*"attribute" + 0.072*"component"')
(17, '0.165*"pattern" + 0.117*"version" + 0.090*"sort" + 0.047*"time" + 0.033*"consume"')
(18, '0.262*"node" + 0.153*"address" + 0.067*"begin" + 0.028*"number" + 0.026*"mark"')
(19, '0.215*"transaction" + 0.103*"schedule" + 0.095*"transactions" + 0.083*"read" + 0.065*"write"')
(20, '0.243*"condition" + 0.093*"tuples" + 0.066*"satisfy" + 0.056*"selection" + 0.039*"horizontal"')
(21, '0.120*"sit" + 0.100*"statistical" + 0.089*"feature" + 0.081*"deal" + 0.044*"topic"')
(22, '0.131*"oracle" + 0.071*"build" + 0.070*"scheme" + 0.065*"database" + 0.046*"learn"')
(23, '0.114*"typically" + 0.094*"collection" + 0.088*"view" + 0.087*"schema" + 0.055*"select"')
(24, '0.166*"variable" + 0.126*"return" + 0.080*"hand" + 0.061*"versions" + 0.057*"query"')
(25, '0.215*"issue" + 0.055*"affect" + 0.046*"earlier" + 0.044*"mention" + 0.039*"survey"')
(26, '0.160*"assume" + 0.139*"single" + 0.095*"overview" + 0.043*"attribute" + 0.042*"brief"')
```

3. Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM):

Computer Networks:

GSDMM made some cluster of similar words for Networking

```
In stage 0: transferred 21133 clusters with 100 clusters populated
In stage 1: transferred 11928 clusters with 96 clusters populated
In stage 2: transferred 5839 clusters with 53 clusters populated
In stage 3: transferred 4801 clusters with 35 clusters populated
In stage 4: transferred 4512 clusters with 33 clusters populated
In stage 5: transferred 4364 clusters with 24 clusters populated
In stage 6: transferred 4313 clusters with 24 clusters populated
In stage 7: transferred 4389 clusters with 25 clusters populated
In stage 8: transferred 4328 clusters with 21 clusters populated
In stage 9: transferred 4266 clusters with 20 clusters populated
Number of documents per topic : [   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0  101 4685    0    0    0    0    0    0    0    0    3    0
 1731 1484    0    0    0    6    0    0    0    0    0 4429 1252    0
    4    0    0 1387    0    1    0    0    0    0    0    2    0
    0    1    0    0    0    0    0    0    0   29    4    0    0
    0 4153    0    0    2    0    0    0    0  998    0    0    0    0
    0    0    0 1299    0    0    0    0    0    0    0    0    0    0
    0   80]
Most important clusters (by number of docs inside): [16 39 71 28 29 45 87 40 79 15 99 66 33 67 42]

Cluster 16 : {'frame': 0.22, 'send': 0.15, 'time': 0.13, 'packet': 0.12, 'data': 0.1, 'network': 0.1, 'packets': 0.1, 'number':
0.08}

Cluster 39 : {'layer': 0.21, 'network': 0.15, 'address': 0.14, 'ip': 0.12, 'message': 0.1, 'server': 0.1, 'protocol': 0.1, 'sen
d': 0.09}

Cluster 71 : {'network': 0.36, 'internet': 0.12, 'telephone': 0.1, 'service': 0.09, 'switch': 0.08, 'cable': 0.08, 'different':
0.08, 'use': 0.08}

Cluster 28 : {'page': 0.25, 'web': 0.22, 'server': 0.11, 'browser': 0.1, 'content': 0.08, 'user': 0.08, 'program': 0.08, 'htm
l': 0.08}

Cluster 29 : {'bite': 0.23, 'bits': 0.19, 'error': 0.12, 'block': 0.12, 'key': 0.11, 'message': 0.08, 'code': 0.08, 'cipher':
0.07}

Cluster 45 : {'signal': 0.25, 'channel': 0.16, 'frequency': 0.13, 'fiber': 0.11, 'band': 0.1, 'time': 0.09, 'send': 0.09, 'bit
s': 0.08}

Cluster 87 : {'rout': 0.21, 'router': 0.17, 'network': 0.14, 'routers': 0.12, 'path': 0.11, 'packet': 0.1, 'tree': 0.08, 'tabl
e': 0.08}

Cluster 40 : {'key': 0.22, 'bob': 0.17, 'alice': 0.16, 'message': 0.16, 'public': 0.09, 'send': 0.08, 'trudy': 0.07, 'certifica
te': 0.05}

Cluster 79 : {'people': 0.16, 'network': 0.15, 'standards': 0.13, 'government': 0.12, 'use': 0.12, 'internet': 0.11, 'work': 0.
11, 'company': 0.1}

Cluster 15 : {'et': 0.38, 'al': 0.37, 'information': 0.08, 'example': 0.05, 'additional': 0.04, 'clark': 0.03, 'saltzer': 0.03,
```

DBMS:

GSDMM made some cluster of similar words for DBMS

```
In stage 0: transferred 13517 clusters with 100 clusters populated
In stage 1: transferred 5568 clusters with 78 clusters populated
In stage 2: transferred 2560 clusters with 49 clusters populated
In stage 3: transferred 2010 clusters with 34 clusters populated
In stage 4: transferred 1902 clusters with 29 clusters populated
In stage 5: transferred 1782 clusters with 29 clusters populated
In stage 6: transferred 1745 clusters with 32 clusters populated
In stage 7: transferred 1703 clusters with 27 clusters populated
In stage 8: transferred 1713 clusters with 26 clusters populated
In stage 9: transferred 1671 clusters with 26 clusters populated
Number of documents per topic : [   0    0    0    9 2530    0   10    0    0    0    0 1155    0    0
    3    2  610    0    0    0    0    0    0    0    0    4    0
    0    2    1    0    0    0    0    1   90    0    0    0    0
    0    0    0    0    0    0    0 1342    0    5    0    0    0  174
  715 1436    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0
 1510    0    6    0 1461    8   17    0    1    0  245    0 2365    0
    0  149]
Most important clusters (by number of docs inside): [ 4 96 84 88 57 49 11 56 16 94 55 99 36 90  6]

Cluster 4 : {'database': 0.24, 'data': 0.14, 'model': 0.12, 'object': 0.12, 'query': 0.1, 'design': 0.1, 'relational': 0.09, 'p
rogram': 0.09}

Cluster 96 : {'query': 0.17, 'attribute': 0.15, 'employee': 0.15, 'value': 0.12, 'relation': 0.12, 'tuples': 0.1, 'department':
0.1, 'tuple': 0.1}

Cluster 84 : {'record': 0.22, 'file': 0.18, 'index': 0.16, 'block': 0.14, 'disk': 0.1, 'value': 0.07, 'field': 0.07, 'search':
0.07}

Cluster 88 : {'data': 0.28, 'database': 0.21, 'distribute': 0.09, 'dbms': 0.09, 'databases': 0.08, 'systems': 0.08, 'process':
0.08, 'storage': 0.08}

Cluster 57 : {'data': 0.16, 'information': 0.16, 'document': 0.14, 'web': 0.14, 'search': 0.12, 'security': 0.1, 'database': 0.
09, 'user': 0.09}

Cluster 49 : {'transaction': 0.25, 'lock': 0.16, 'schedule': 0.15, 'write': 0.1, 'transactions': 0.1, 'database': 0.08, 'item':
0.08, 'commit': 0.08}

Cluster 11 : {'type': 0.23, 'attribute': 0.17, 'entity': 0.16, 'relationship': 0.14, 'object': 0.08, 'figure': 0.08, 'class':
0.07, 'model': 0.06}

Cluster 56 : {'relation': 0.21, 'dependencies': 0.16, 'key': 0.12, 'attribute': 0.12, 'functional': 0.11, 'form': 0.1, 'desig
n': 0.09, 'normal': 0.08}

Cluster 16 : {'course': 0.16, 'student': 0.16, 'database': 0.15, 'number': 0.14, 'example': 0.13, 'figure': 0.09, 'attribute':
0.09, 'time': 0.08}
```

Chapter 6: EVALUATION

6.1 Diagram/Graphs:

InterTopic Distance Map: A visual representation of the distances between different topics in a corpus.

Similarity matrix: A table or matrix that displays the similarity between pairs of objects or variables.

Topic word score: A measure that indicates the relevance of a particular word to a given topic.

Hierarchical Clustering: A method of clustering data points into a tree-like structure based on their similarities or distances.

Word Cloud - A visualization technique that displays the frequency of words in a corpus as a collection of words arranged in a cloud-like shape, where the size of each word represents its frequency.

- **BERT:**
  - **Computer Networks:**

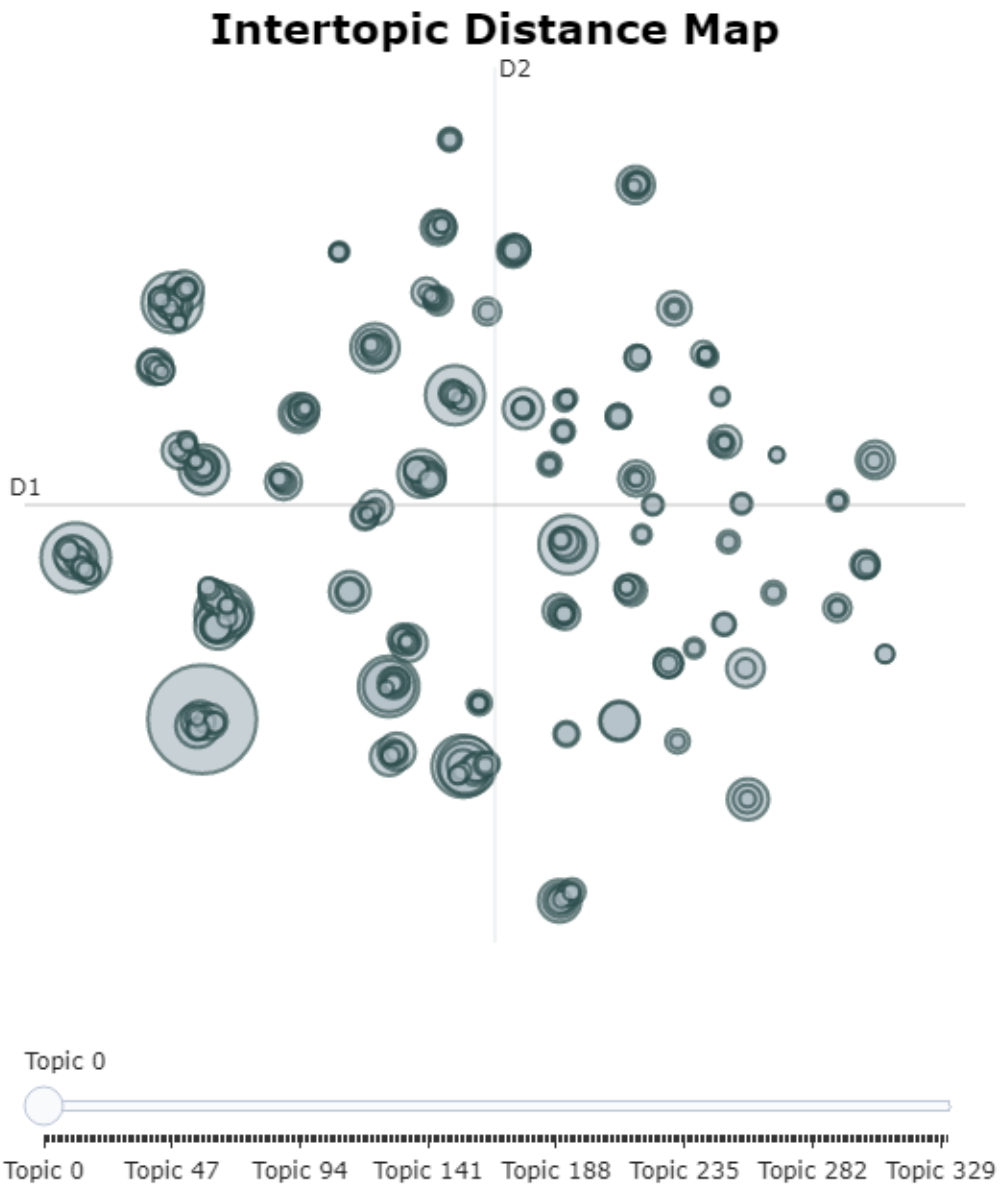


Fig 6.1 Intertopic Distance Map of topics identified by the BERT model for Computer Networks subject

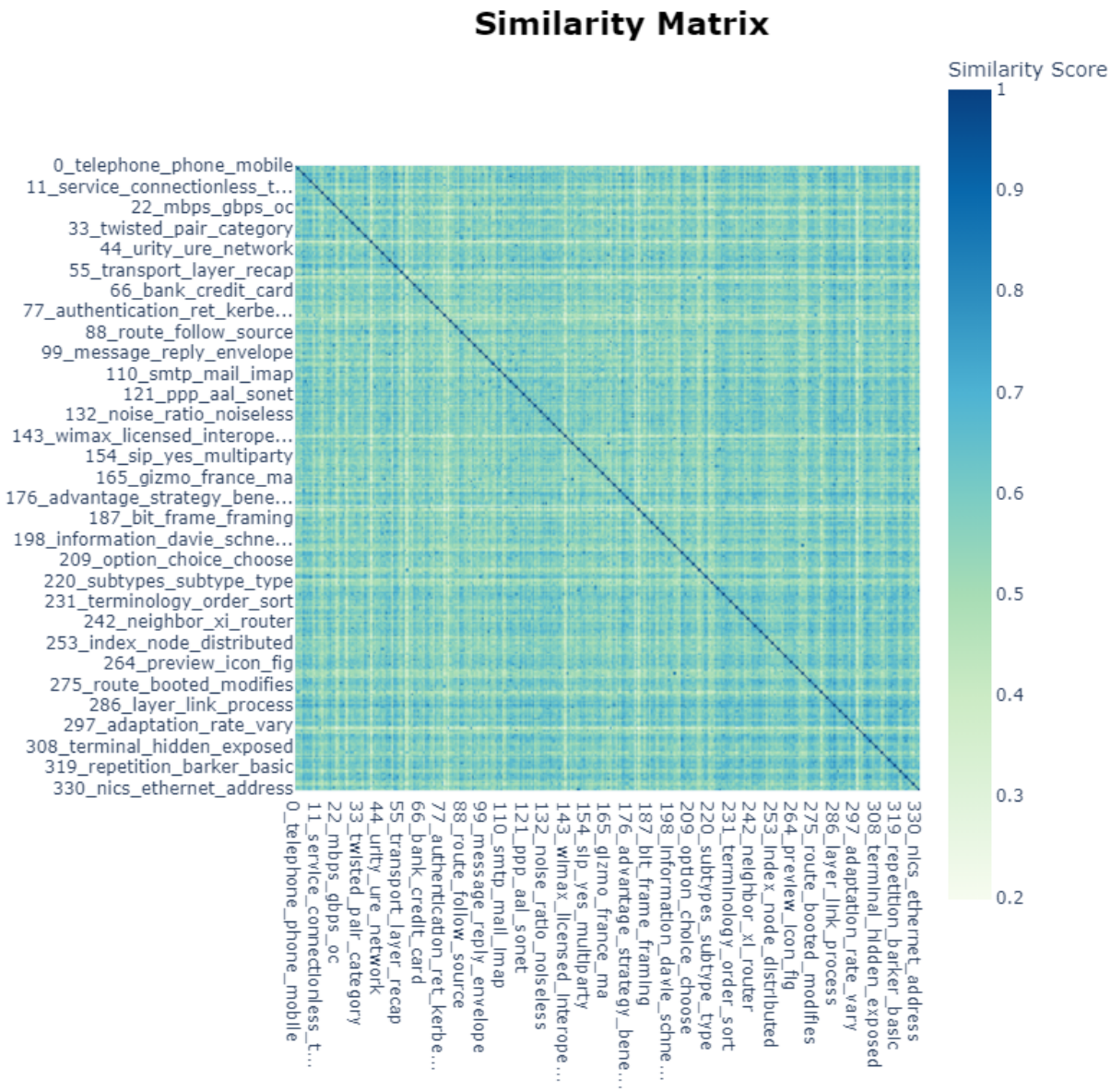


Fig 6.2: Similarity Matrix of topics identified by the BERT model for Computer Networks subject

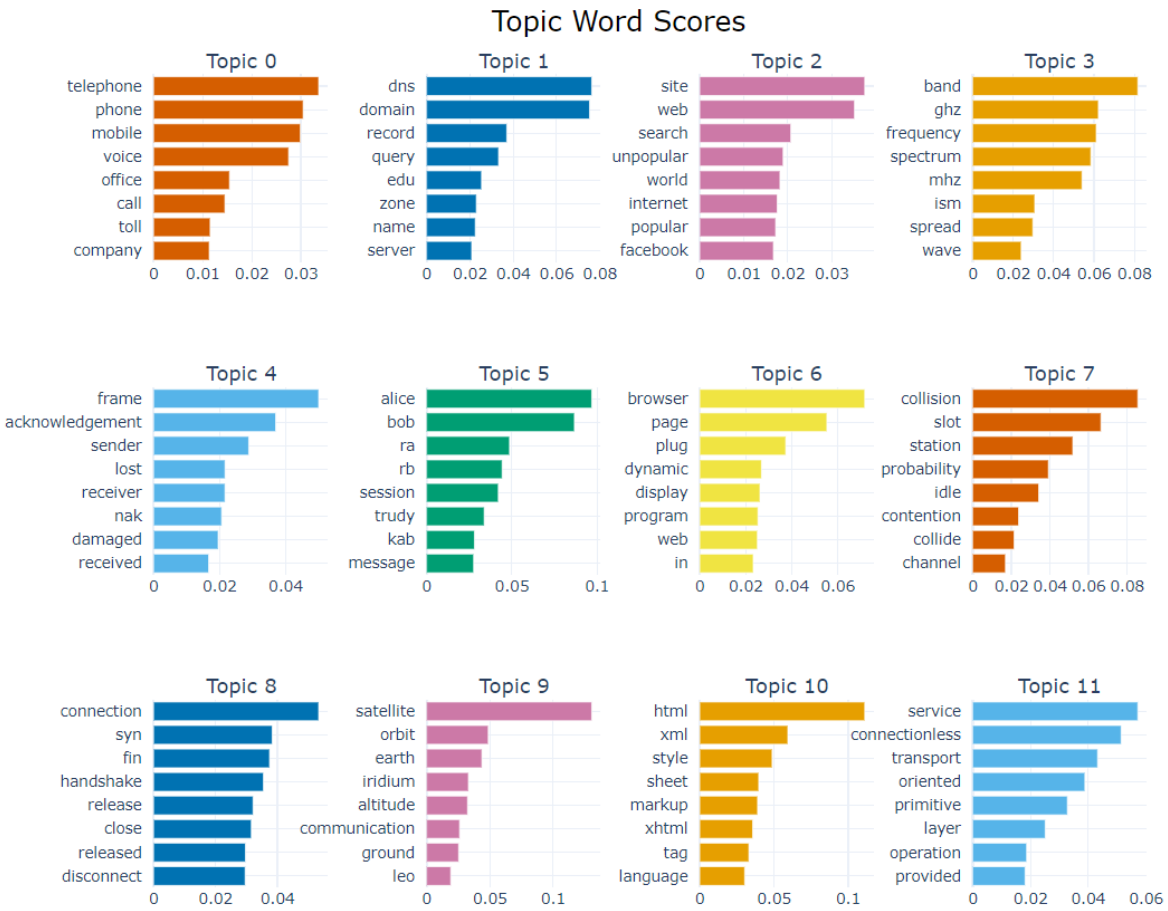


Fig 6.3: Topic Word Score of topics identified by the BERT model for Computer Networks subject

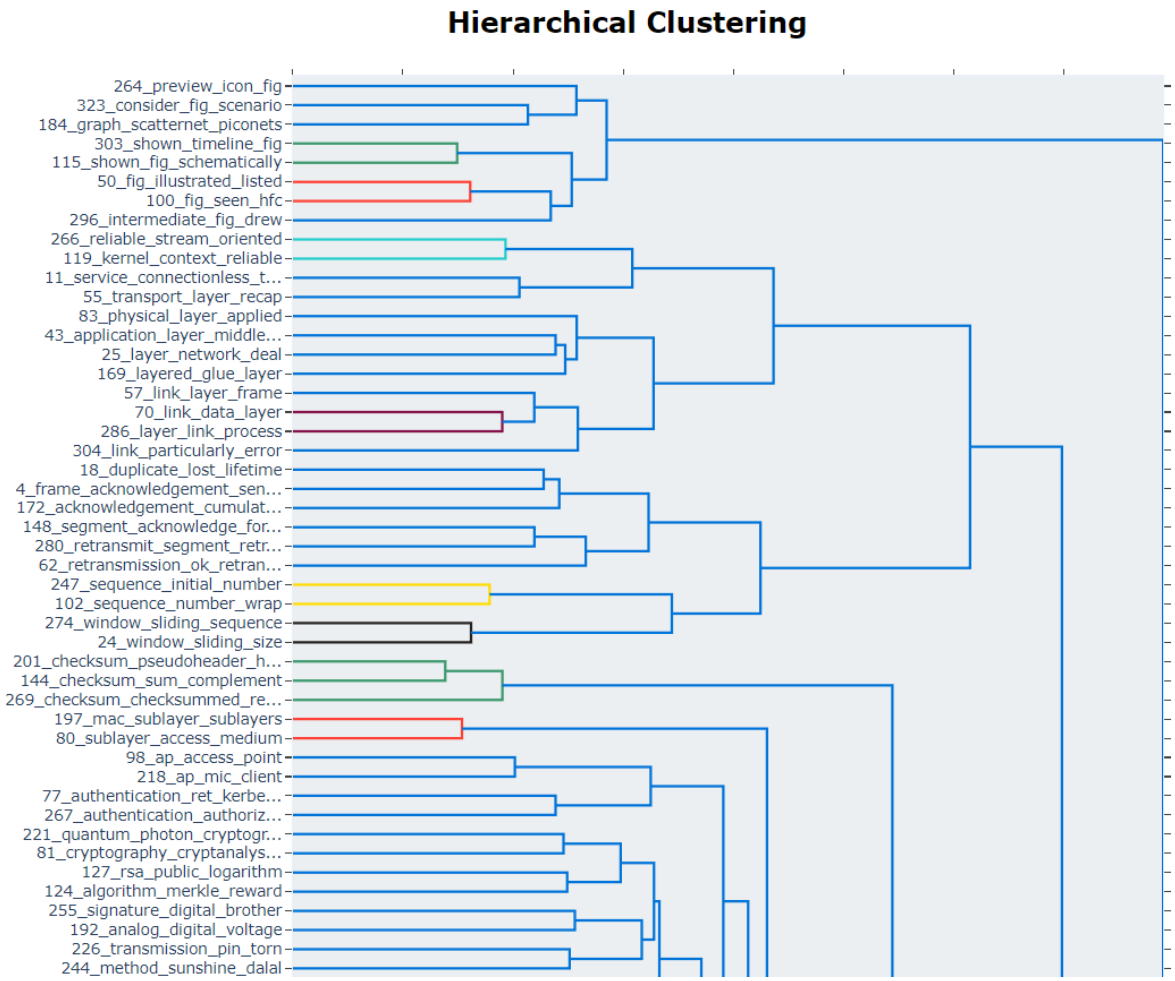


Fig 6.4: Hierarchical Clustering of topics identified by the BERT model for Computer Networks subject

➢ **DBMS:**

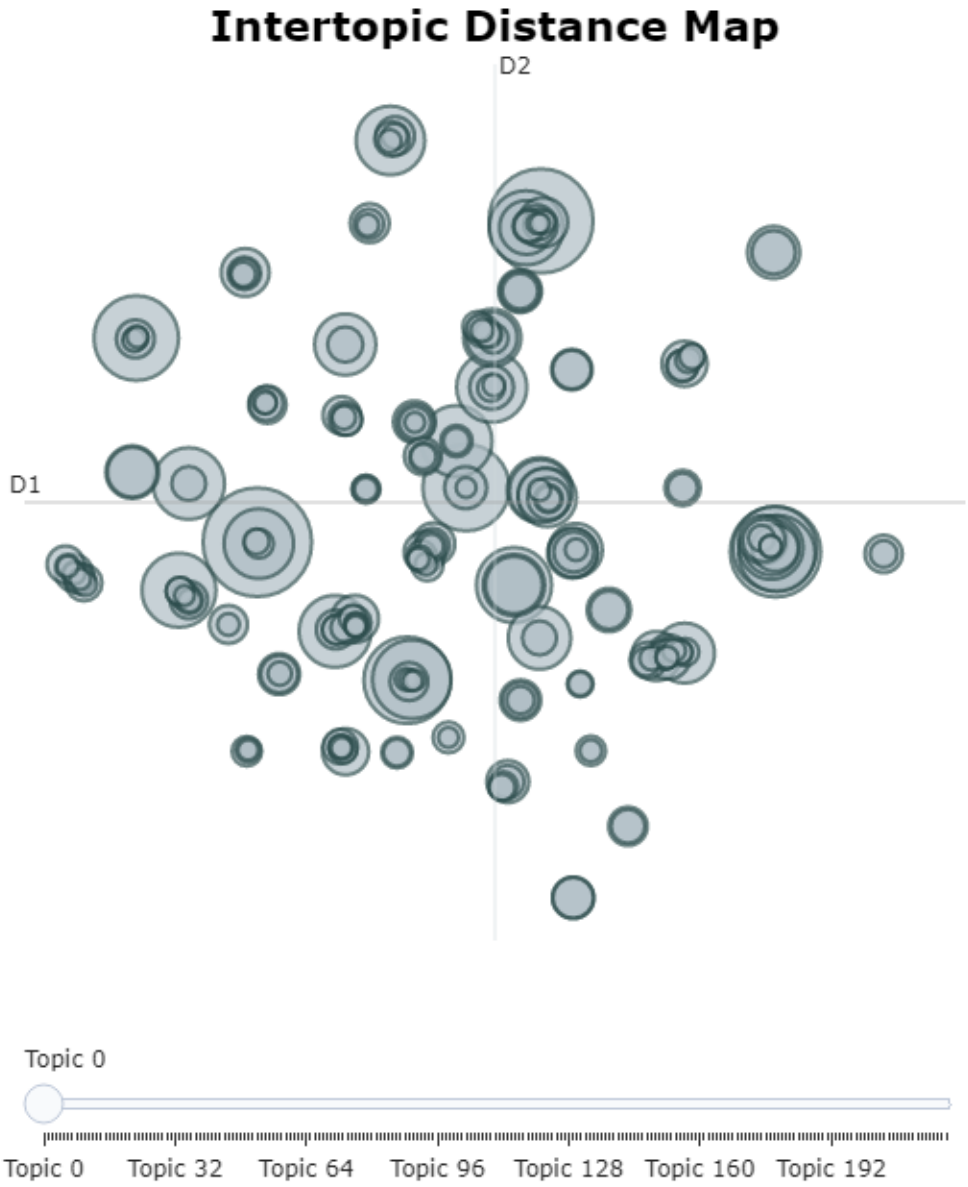


Fig 6.5: Intertopic Distance Map of topics identified by the BERT model for DBMS subject

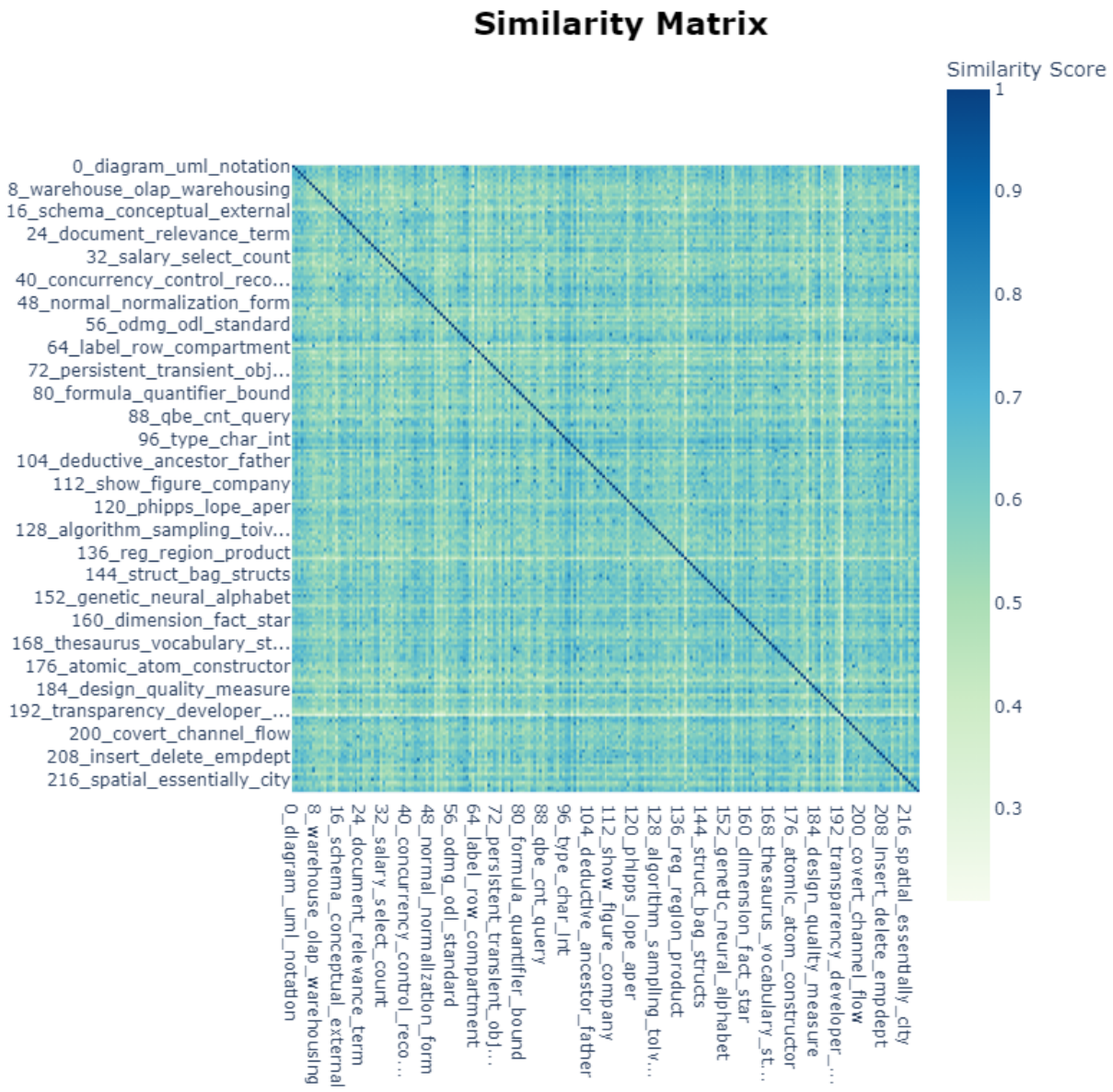


Fig 6.6: Similarity Matrix of topics identified by the BERT model for DBMS subject

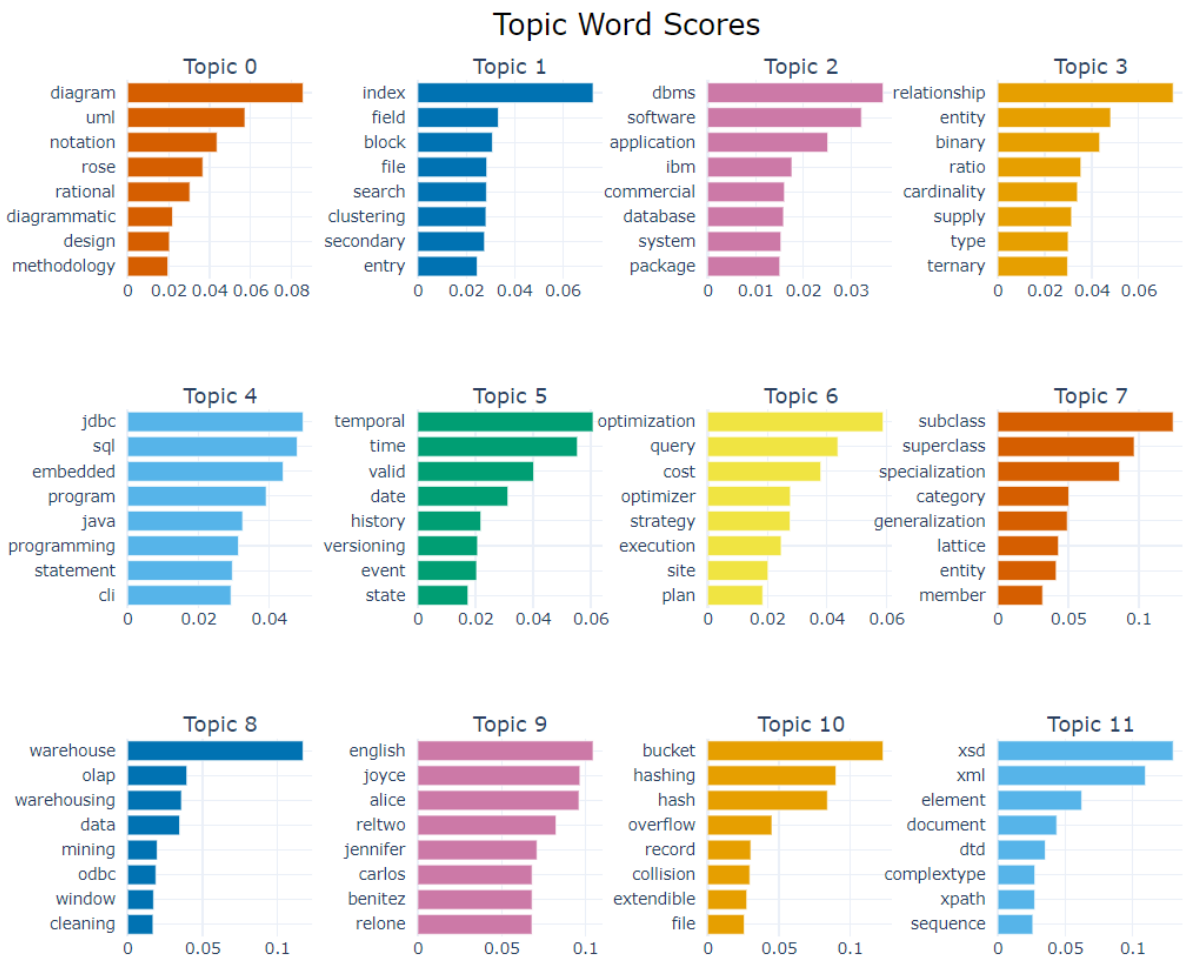


Fig 6.7: Topic Word Score of topics identified by the BERT model for DBMS subject

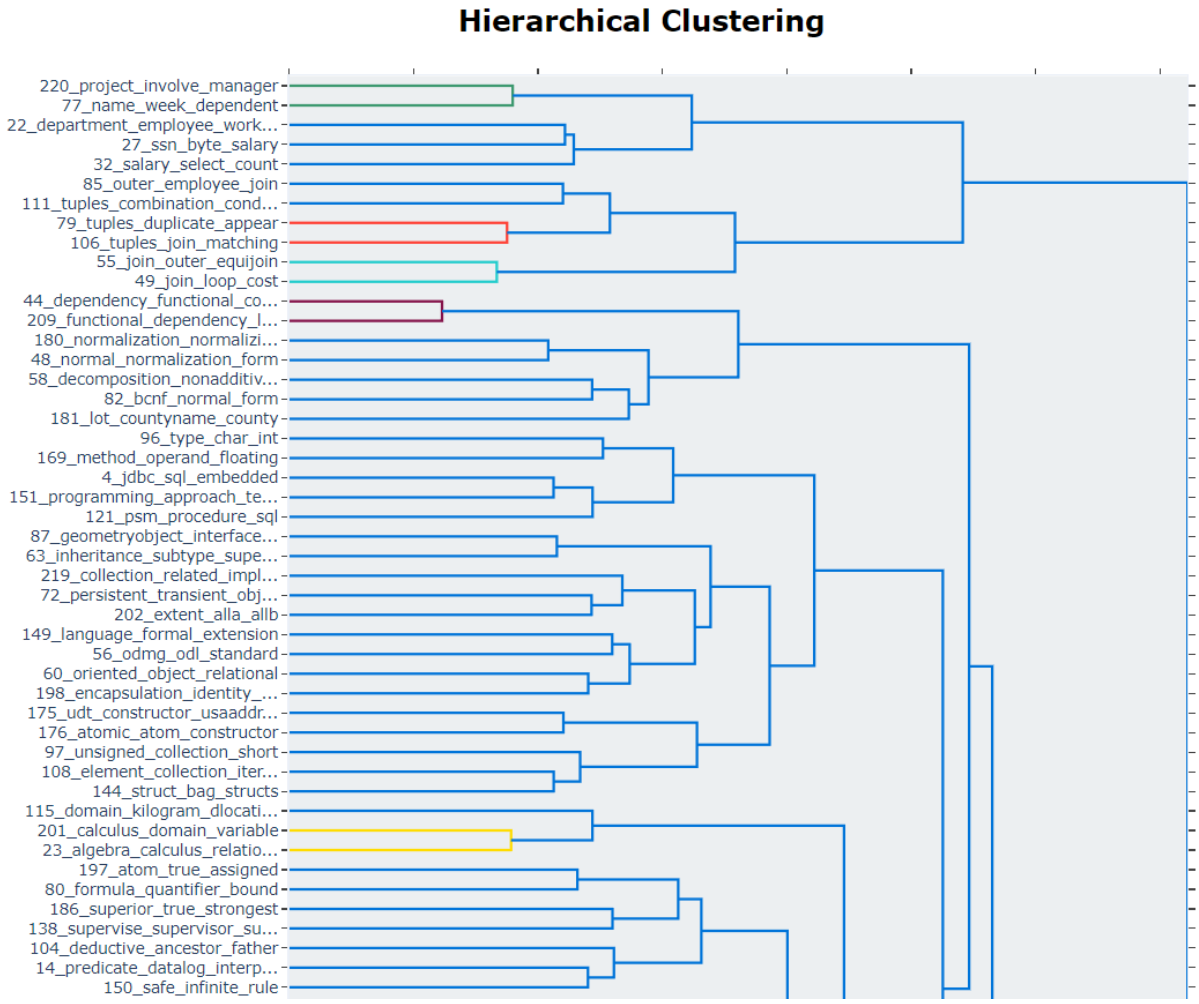


Fig 6.8: Hierarchical Clustering of topics identified by the BERT model for DBMS subject

❖ LDA:

➢ Computer Networks

Fig 6.8:LDA for Computer Networks

➢ **DBMS:**

Fig 6.9:LDA for DBMS

GSDMM:

Computer Networks

Fig 6.10: GSDMM for computer networks

DBMS:

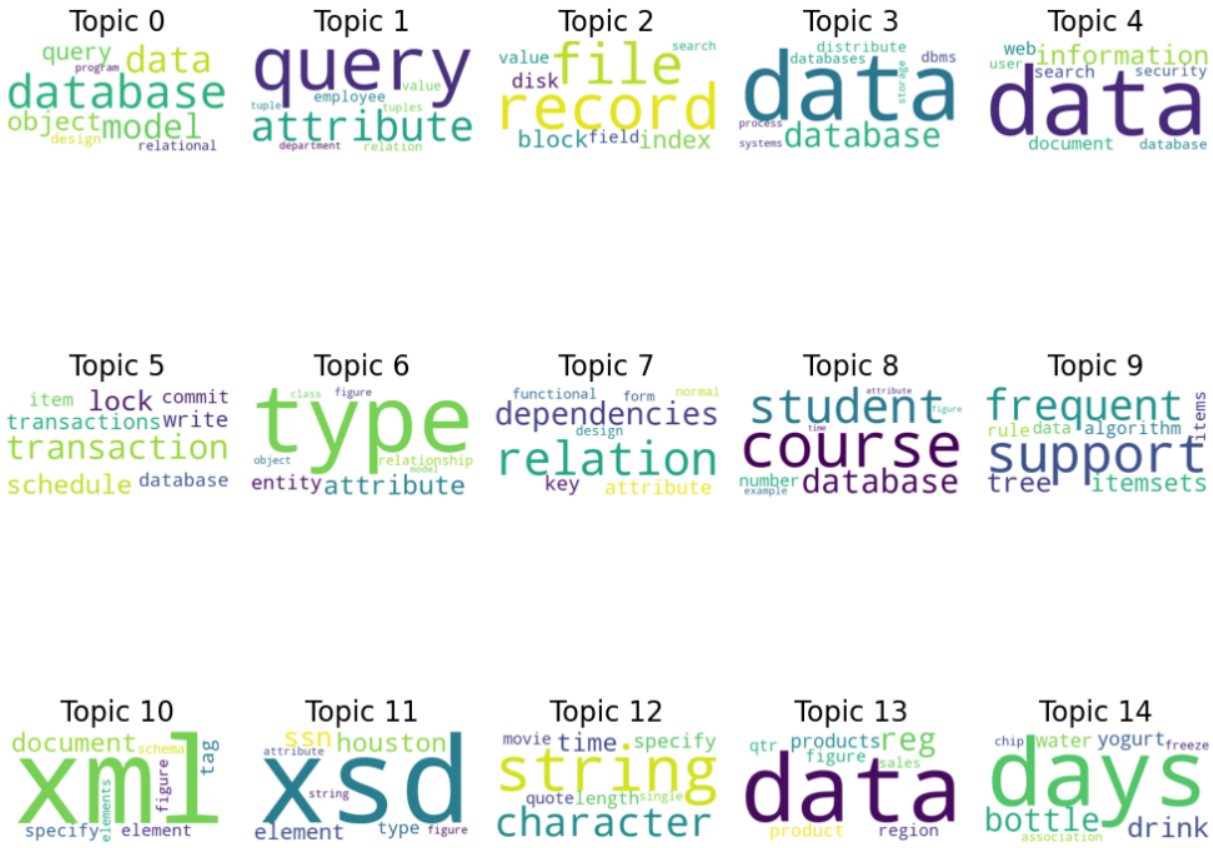


Fig 6.11: GSDMM for DBMS

6.2 Coherence:

The Topic Coherence score is an objective measure which is rooted in the distributional hypothesis of linguistics: words with similar meanings tend to occur in similar contexts. Topics are considered to be coherent if all or most of the words are closely related.
Performance metric used here is Coherence(**u_mass**). u_mass coherence captures the optimal number of topics by giving the interpretability of these topics a number called coherence score. u_mass closer to value 0 means perfect coherence and it fluctuates either side of value 0 depending upon the number of topics chosen and the kind of data used to perform topic clustering.

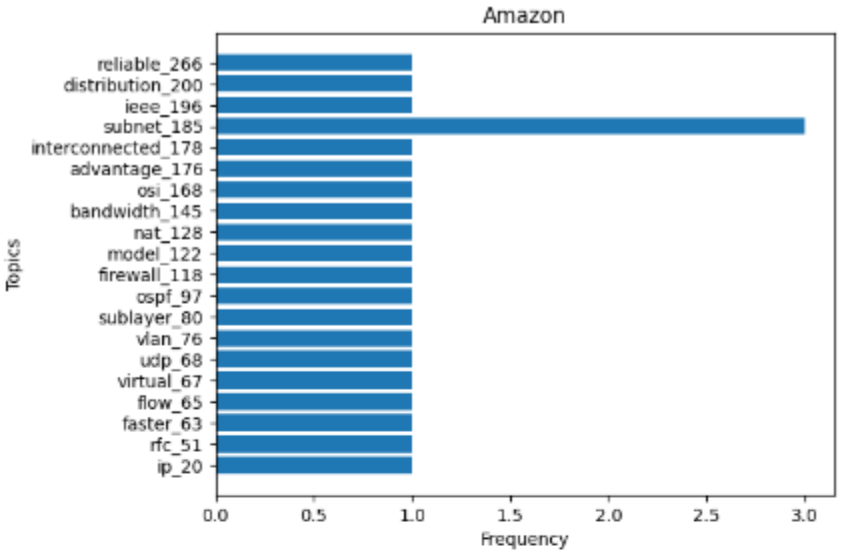
- **u_mass** is always negative.

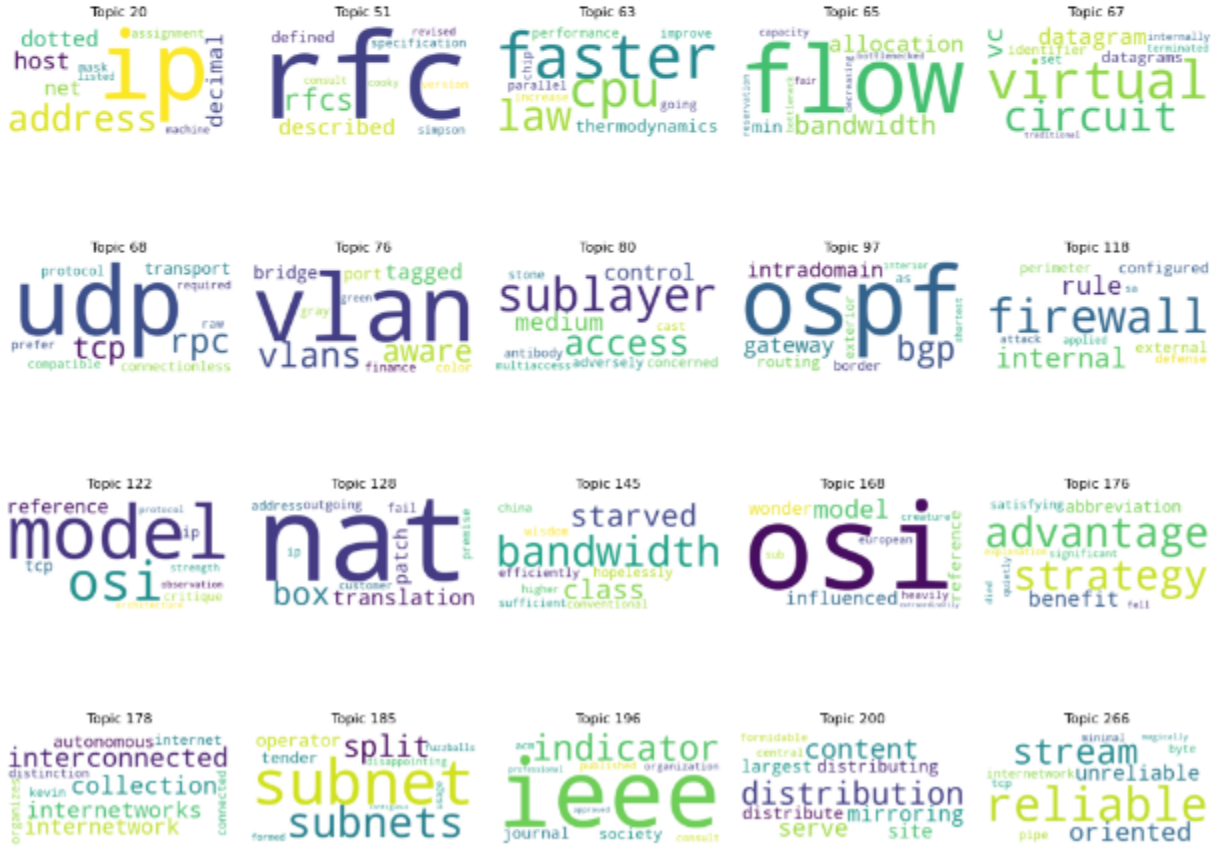
We have also shown a graph of change in coherence with change in number of topics along with coherence of the whole document after applying LDA and GSDMM.

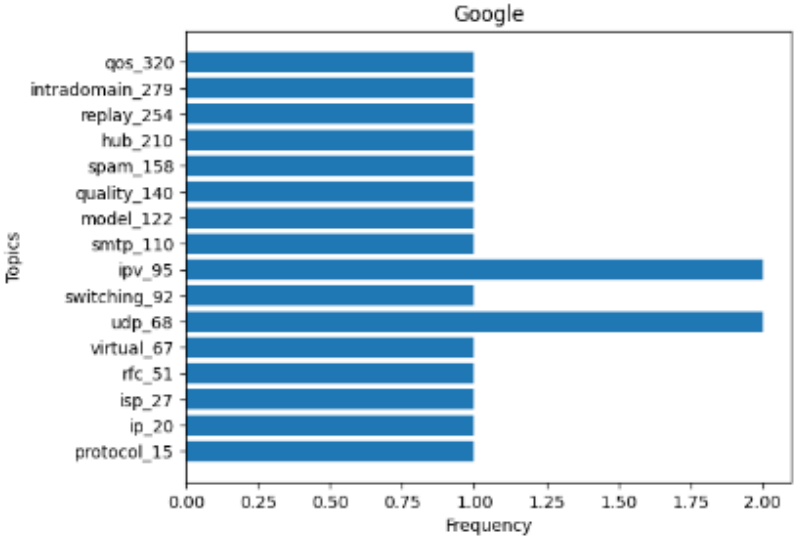
6.3 Frequency of topics:

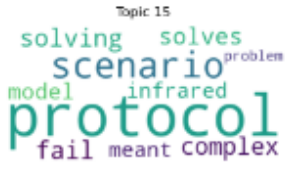
We have analyzed the frequency of topics from which the question is being asked in the different companies

Computer Networks:






Google
qos_320
intradomain_279
replay_254
hub_210
spam_158
quality_140
model_122
smtp_110
ipv_95
switching_92
udp_68
virtual_67
rfc_51
isp_27
ip_20
protocol_15
Topics
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Frequency

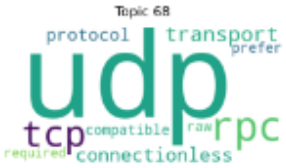

Topic 15
solving
solves
problem
scenario
model
infrared
protocol
fail
meant
complex


Topic 20
mask
decimal
net
dotted
machine
ip
assignment
host
listed
address


Topic 27
backbone
pay
customer
isp
isps
ixps
pop
connect
internet
provider

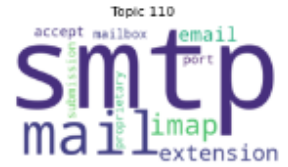

Topic 51
consult
specification
defined
cooky
version
revised
rfcs
rfc
simpson
described

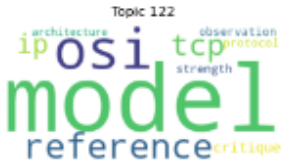

Topic 67
set
terminated
internally
virtual
vc
traditional
identifier
circuit
datagram
datagrams


Topic 68
protocol
transport
prefer
udp
tcp
compatible
raw
rpc
required
connectionless


Topic 92
switch
switched
npl
charging
packetswitched
switching
comprehending
equipment
circuit
packet


Topic 95
flexible
island
deployed
major
ipv
ip
ietf
proved
deployment
version


Topic 110
accept
mailbox
email
port
smtp
submission
proprietary
mail
imap
extension


Topic 122
architecture
observation
ip
osi
tcp
protocol
strength
model
reference
critique


Topic 140
requirement
mechanism
enforces
integrated
differentiated
idling
quality
service
capability
clearest

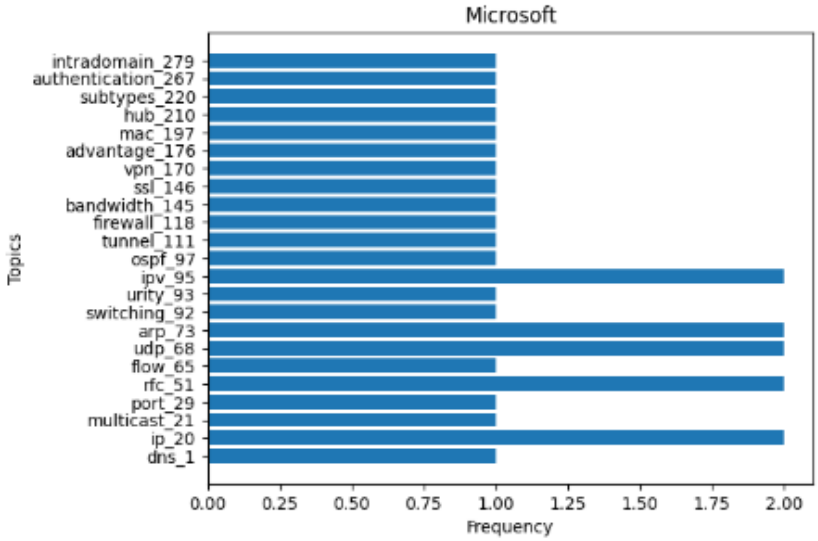

Topic 158
spam
forged
spammer
electronic
mcafee
junk
mail
generated
unwanted
probably


Topic 210
switch
advent
specie
hub
amplify
gateway
changed
repeater
bridge
endangered


Topic 254
attack
repudiate
handledintegrity
onds
concocted
replay
replayed
recy
challenge


Topic 279
exterior
externally
participate
intradomain
interdomain
metz
routing
gateway
interior
protocol


Topic 320
qos
seek
guarantee
quality
lumped
shaping
traffic
reserving
service
admission


Microsoft
intradomain_279
authentication_267
subtypes_220
hub_210
mac_197
advantage_176
vpn_170
ssl_146
bandwidth_145
firewall_118
tunnel_111
ospf_97
ipv_95
urity_93
switching_92
arp_73
udp_68
flow_65
rfc_51
port_29
multicast_21
ip_20
dns_1
Topics
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Frequency


Topic 1
query
domain
dns
server
name
zone
record
edu

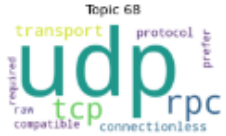

Topic 20
dotted
decimal
mask
listed
ip
machine
address
host
net
assignment


Topic 21
unicast
spanning
multicast
anycast
group
multicasting
member
tree
routing


Topic 29
sockaddr
identify
sizeof
inet
fatal
tcp
port
failed
socket
struct


Topic 51
defined
revised
version
rfc
simpson
consult
cooky
rfcs
specification
described


Topic 65
allocation
min
flow
bandwidth


Topic 68
transport
protocol
udp
prefer
required
tcp
rpc
ran
compatible
connectionless


Topic 73
arp
cache
host
address
dhcp
gratuitous
configuration
mapping
broadcast

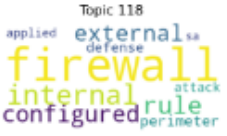

Topic 92
switching
switch
switched
npl
charging
circuit
packet


Topic 93
encrypted
encrypt
urity
encryption
integrity
cryptographic


Topic 95
major
deployment
ipv
proved
deployed
ietf
island
ip
flexible
version


Topic 97
intradomain
ospf
shortest
border
interior
gateway
routing
bgp


Topic 111
london
tunnel
tunneling
parisipv


Topic 118
applied
external
defense
firewall
internal
attack
configured
rule
perimeter

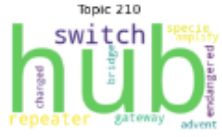

Topic 145
hopelessly
china
higher
bandwidth
conventional
starved
wisdom
sufficient
class
efficiently

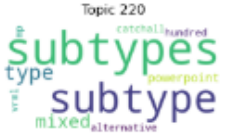

Topic 146
version
ssl
tl
ure
src
http
incompatibility


Topic 170
vpn
vpns
private
virtual


Topic 176
strategy
abbreviation
advantage
benefit
satisfying


Topic 197
mac
layer
upper
date
sublayer
land
physical

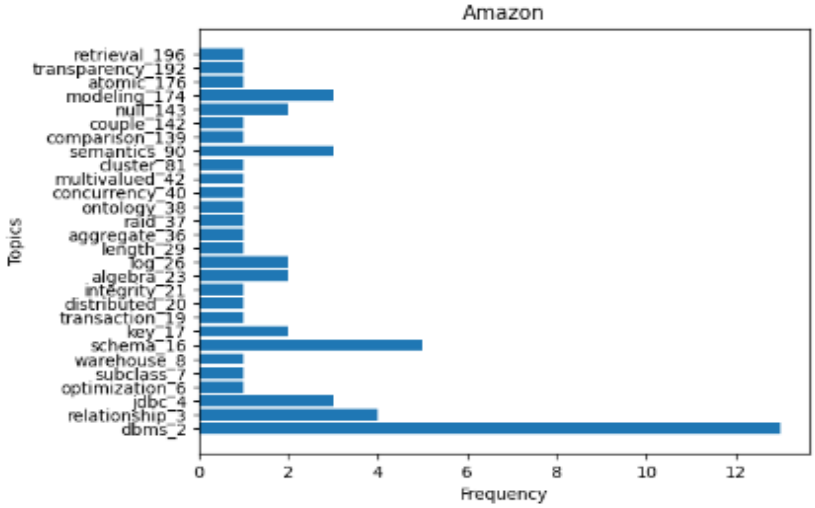

Topic 210
switch
hub
repeater
gateway
advent

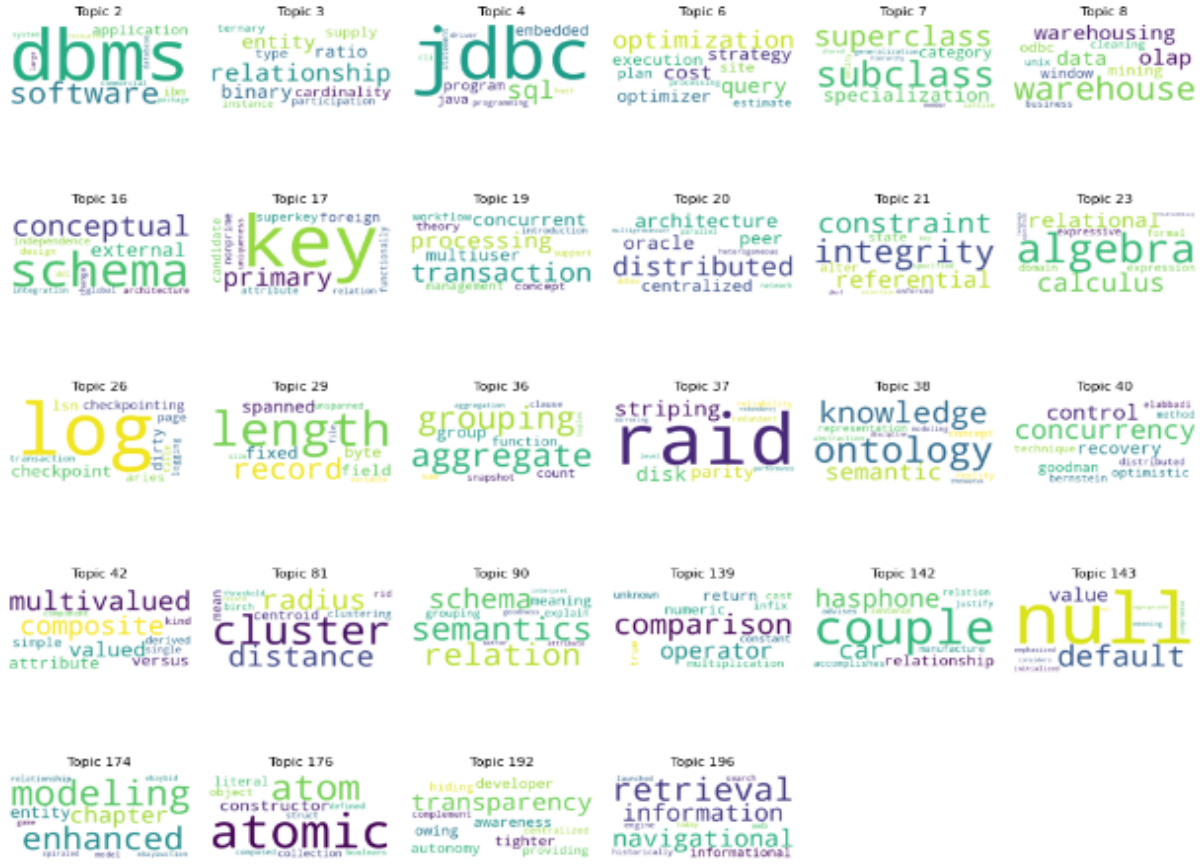

Topic 220
subtypes
type
subtype
mixed

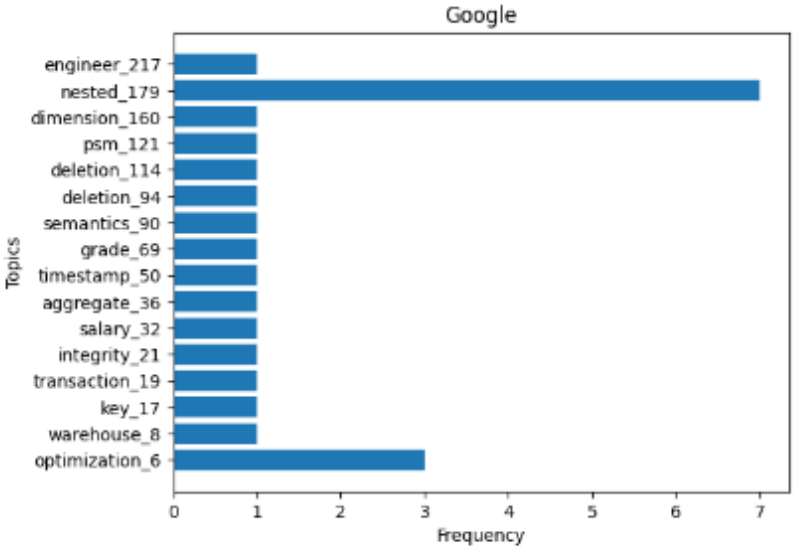

Topic 267
authentication
answered
affirmative
authorization
scott
moral
impersonating
confuse


Topic 279
interdomain
interior
gateway
intradomain
exterior
routing

DBMS:


Amazon
retrieval_196
transparency_192
atomic_176
modeling_174
null_143
couple_142
comparison_139
semantics_90
cluster_81
multivalued_42
concurrency_40
ontology_38
raid_37
aggregate_36
length_29
log_26
algebra_23
integrity_21
distributed_20
transaction_19
key_17
schema_16
warehouse_8
subclass_7
optimization_6
jdbc_4
relationship_3
dbms_2
Topics
0
2
4
6
8
10
12
Frequency

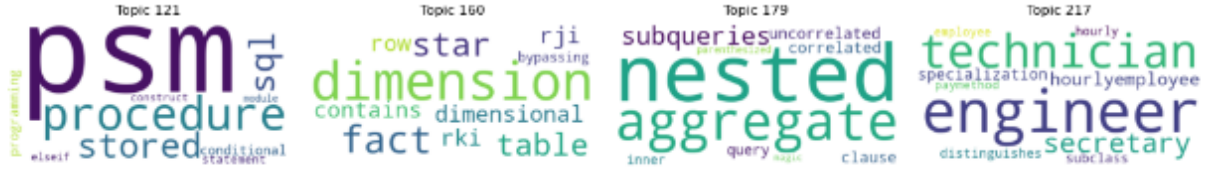

Topic 2
dbms
application
software
Topic 3
entity
supply
ratio
type
relationship
binary
cardinality
Topic 4
jdbc
embedded
program
sql
java
Topic 6
optimization
execution
strategy
plan
cost
query
optimizer
Topic 7
superclass
category
subclass
specialization
Topic 8
warehousing
odbc
data
olap
window
mining
warehouse
Topic 16
conceptual
external
schema
Topic 17
superkey
foreign
key
primary
Topic 19
concurrent
processing
multiuser
transaction
Topic 20
architecture
oracle
peer
distributed
centralized
Topic 21
constraint
integrity
referential
Topic 23
relational
algebra
calculus
Topic 26
log
lsn
checkpointing
checkpoint
dirty
Topic 29
spanned
length
fixed
byte
record
field
Topic 36
grouping
group
function
aggregate
count
Topic 37
striping
raid
disk
parity
Topic 38
knowledge
ontology
semantic
Topic 40
control
concurrency
recovery
goodman
bernstein
optimistic
Topic 42
multivalued
composite
simple
valued
attribute
Topic 81
radius
centroid
cluster
distance
Topic 90
schema
meaning
semantics
relation
Topic 139
return
numeric
comparison
operator
Topic 142
hasphone
couple
car
relationship
Topic 143
value
null
default
Topic 174
modeling
entity
chapter
enhanced
Topic 176
literal
atom
constructor
atomic
Topic 192
hiding
developer
transparency
awareness
owing
tighter
autonomy
Topic 196
retrieval
information
navigational
informational


Google
engineer_217
nested_179
dimension_160
psm_121
deletion_114
deletion_94
semantics_90
grade_69
timestamp_50
aggregate_36
salary_32
integrity_21
transaction_19
key_17
warehouse_8
optimization_6
Topics
0
1
2
3
4
5
6
7
Frequency


Topic 6
query
site
processing
optimizer
cost
plan
strategy
execution
optimization
estimate
Topic 8
cleaning
business
warehouse
odbc
olap
data
unix
warehousing
mining
window
Topic 17
superkey
foreign
key
attribute
uniqueness
nonprime
primary
relation
functionally
candidate
Topic 19
theory
multiuser
workflow
processing
management
support
introduction
transaction
concept
concurrent


Topic 21
specified
constraint
referential
enforced
key
assertion
integrity
dknf
state
alter
Topic 32
employee
lname
department
count
salary
select
average
Topic 36
function
count
aggregation
aggregate
snapshot
grouping
group
sum
tuples
clause
Topic 50
ordering
protocol
timestamp
successfully
unique
date
transaction
timestamps
multiversion
generated


Topic 69
student
grade
course
graderepport
prerequisite
Topic 90
interpret
better
attribute
relation
grouping
goodness
explain
semantics
meaning
schema
Topic 94
algorithm
half
node
deletion
outline
tree
insertion
overflow
simulation
redistribution
Topic 114
deleted
space
wasted
record
deletion
garbage
reorganization
deleting
marker
periodic


Topic 121
psm
tbsql
procedure
construct
module
programming
stored
elseif
conditional
statement
Topic 160
row
star
rji
bypassing
dimension
contains
dimensional
fact
rki
table
Topic 179
subqueries
uncorrelated
correlated
nested
aggregate
inner
query
clause
Topic 217
employee
hourly
technician
specialization
hourlyemployee
engineer
distinguishes
secretary
subclass

Chapter 7: RESULTS

Coherence score table

SUPERVISED CLASSIFICATION:

Logistic Regression	0.9169739952718676
Decision Tree	0.9197163120567378
Naive Bayes	0.9139479905437352
KNN	0.921513002364066
Random Forest	0.9324822695035462
One VS Rest	0.8712056737588654

UNSUPERVISED:

COMPUTER NETWORKS

No. of Topics	LDA	GSDMM
100	-2.3450868071558157	-4.024005395437741
150	-2.9554905147933064	-4.132307746130759
200	-5.4940617850492615	-4.228884278530324
250	-7.90196257192071	-4.403659263149508
300	-9.182522990767493	-4.8253443473075714

DBMS

No. of Topics	LDA	GSDMM
100	-4.775388724311264	-2.9621299147556317
150	-5.249026706228136	-2.731512709061364
200	-5.725387710055674	-3.4099775715402094

BERT

CN (333 topics)	-1.2875920040861317
DBMS (222 topics)	-1.1656034655139083

CONCLUSIONS

Supervised Classification:
After preprocessing and refining the data we tried implementing many models to classify multi tags one by one. However there is not much significant difference among the models but among all the models Random Forest performed well.

Unsupervised Classification:
In the end , we have been able to obtain different varieties of clusters of topics for the same data. These clusters of topics are somewhat meaningful in sense but after analyzing different approaches we conclude that using LDA and GSDMM we could not extract significant topics

Further we have analyzed that the **BERT works better in extracting topics than LDA and GSDMM in our dataset** as it also predicts the topic/cluster very well.

We have extended our study to further link working of this project with DBMS subject, where also we have seen similar outcomes of BERT working better.

Chapter 8: FUTURE POSSIBILITIES

- Explore fuzzy modeling techniques to handle uncertainty and improve topic predictions.
- Investigate transfer learning methods to leverage knowledge across different question domains.
- Continuously refine and fine-tune machine learning models to enhance accuracy.
- Expand the question bank by collecting and incorporating new questions from diverse sources.
- Integrate with learning management systems for real-time analysis and personalized feedback.
- We will try to further use N-gram in BERT to improve predictions.

REFERENCES

[1] R. Gupta, B. Malik, A. Joshi, and R. Mamidi, "Analyzing BERT for Question-Answering: What Works, What Doesn't, and What Matters?" in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 2493-2502.

[2] N. Gupta, R. Gupta, S. Bansal, and D. Jurafsky, "An Analysis of BERT Performance for Reading Comprehension," in 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 1688-1697.

[3] Mazarura, J. R. (2015). *Topic modelling for short text* (Doctoral dissertation, University of Pretoria).

[4] M. Yu, K. M. Gupta, Q. Yang, and T. A. Faruquie, "A Comparison of Machine Learning Algorithms for Student Success Prediction," in 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT), 2017, pp. 292-296.

[5] Weston, C., Gandell, T., Beauchamp, J., McAlpine, L., Wiseman, C., & Beauchamp, C. (2001). Analyzing interview data: The development and evolution of a coding system. *Qualitative sociology*, *24*, 381-400.

[6] Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2020). Short text topic modeling techniques, applications, and performance: a survey. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1427-1445.