Expert Recommendation Model for Stack-Overflow

Sayali Kudale
Computing and Software System
University Of Washington
Bothell, WA, USA
sayalik@uw.edu

Dong Si
Computing and Software System
University Of Washington
Bothell, WA, USA
dongsi@uw.edu

ABSTRACT

Community Question Answering (CQA) is a widely used platform by information and knowledge seekers to satisfy their quest. Stack Overflow (SO) is one such a most popular community with more than ten thousand posts per day. However, such a huge amount of data puts forward new challenges of redirecting new questions to the correct group of experts. Consequently, many questions remain unanswered or get delayed receiving answers. In this project, we are proposing the answerer prediction model for SO where a group of experts who are most active when a question is posted will be recommended to the new question based on matching the skills required for answering the question and quality answers given in the past. We have implemented Latent Dirichlet Allocation (LDA) topic modelling technique and K-Means clustering to find the similar clusters of questions. The test data is matched with the created clusters and clusters with highest similarity is extracted. We further perform filtering based on the Creation Date of the question in the selected cluster to get users who are most active when the question is posted. We have further implemented ranking on the filtered cluster data based on the quality score of the accepted answers. Finally, top k experts are recommended to the question.

CCS CONCEPTS

• Natural Language Processing • TF-IDF • Topic Modelling • K-Means Clustering • Ranking

KEYWORDS

Natural language processing, data cleaning, text mining, LDA Topic modelling, k-Means clustering, stack overflow

1 INTRODUCTION

In recent times, community question answering (CQA) platforms such as Stack-Overflow (SO), Quora have become immensely popular. Specifically, SO is used by many engineers and programmers to discuss the questions related to programming languages like Java, C++, Python among others.

Over a period of time, SO has earned the trust of users to provide correct answers within a short period of time. In this platform community users can ask, answer and participate in the discussion of the answers to the posted question, the asker of the question can mark answers as accepted once a satisfactory solution is received. This not only benefits the user who has posted the questions but also the million other computer programmers who are facing the similar issues.

As of March 2021, SO has nearly 14 million registered users, more than 21 million questions, and more than 31 million answers [1]. As the rate of asking questions increases, the rate of open questions also increases. We can visualize from Figure 1 the rate at which the number of questions increases till year 2016. This trend continues to grow in subsequent years as well.

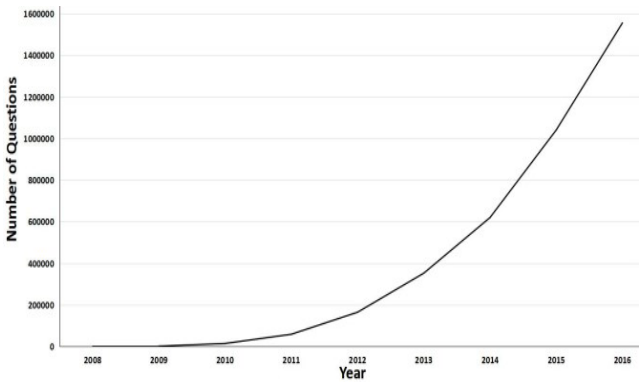

Figure 1: Number of open questions [2]

By considering the popularity of SO and at the same time the rate at which open question increases over this platform, it becomes utmost important to introduce automated models in existing systems to increase efficiency. This paper aims to resolve this problem by providing a model to recommend the set of users who possess the skills to provide the correct answer to the posted question. Moreover, while implementing we also consider the quality of the answers posted by the user as well as the activeness of the user over this platform when a question is posted. With this approach and by evaluating the results we claim that our model can accurately predicts the users who are currently active in answering the questions over the community and has right skills to answer the posted question.

The remainder of the paper is organized as follows: In Section 2, we will discuss the previous research and studies conducted for the CQA Expert finding system. In Section 3, we will explore the dataset features and data pre-processing steps. In Section

4, we discuss the approaches and methodology of the expert finding system. In Section 5, we will present the results and evaluations of the experiment. Finally, we will conclude in section 6 and discuss the future scope of the project.

2 RELATED WORK

Considerable efforts have been taken to find the experts in the question and answering community. In [3], the researchers have delivered a comprehensive framework to categories the studies done on the expert recommendation system for CQA between January 2008 to March 2019, which evaluated gaps and various approaches of the studies. Their study suggests the need to explore the different aspects of the dataset features using a comprehensive model for better results. Additionally, in [3] which is published in April 2018, researchers have reviewed the current progress of expert finding in CQA by conducting a survey and concluded two major areas needed to consider while developing the expert recommendation model for any CQA. Firstly, find users who are most likely to have expertise to answer the question and secondly have the willingness to accept the invitation to answer the question. Our proposed model is designed based on these two major areas.

Ayushi along with other researchers [4], proposed an expert recommendation system for stack-overflow using K-mean clustering and social network technique by achieving 60% of accuracy. In [5], the author used feature-based and social network-based prediction approach. In [6], researcher recommended experts for specific skills such as Java, Python using page rank and HIITS algorithm on the social network. Nan with his colleagues [7], proposed a novel expert finding system by developing the Exp-Rank algorithm which is basically a link-based ranking algorithm based on the quality of user post and authority of users in the stack overflow community. Although this approach gave the accuracy score of 0.67, this algorithm entirely based on the existing tags given by the user while posting the question which may not be always correct.

In [2] Bin along with his colleagues developed the answerer recommendation system using the Latent Dirichlet Allocation (LDA) topic modelling technique. After getting the clusters of topics they have implemented the Random Forest classifier using features of the questions and user features to get the list of probable answers. However, they have only considered the traditional features of the question and the answer information and they ignored the probable answerer's profile information. Along the same line, in [8] the researchers compared the performance of LDA and Segmented Topic modelling (STM) without considering the other metadata available in the dataset such as score of the questions and answers, user badges and reputation, etc.

We have also studied the research done on other CQA platforms such as Yahoo for expert recommendation. In [9], researchers have proposed a concept of question routing to appropriate users for the Yahoo platform by considering both users' expertise and users' availability.

In most of the answerer recommendation models proposed for SO platform so far, the recency attribute of the user has been neglected by many researchers, hence this project aims to consider not only the quality of the user's post but also the recent activity of the user on the stack-overflow community.

3 DATASET ANALYSIS

3.1 Dataset Generation

The dataset used in this project is prepared by gathering the data from the data explorer of the Stack exchange website [10]. The Stack exchange data explorer is a platform for data science professionals and researchers where one can execute the *mysql* queries against the data from various question answering sites in the stack exchange network. The Stack overflow database has over 21 million questions, 31 million answers, and 60,000 tags.

The stack overflow database consists of Posts, Tags, Users, Comments, Badges, etc. Tables. The query has been executed on the Posts table to retrieve the Questions and Answers information. We collected a total of 3,01,013 high-quality questions that were posted between January 2020 to January 2021. The users prefer to answer questions that have good readability. The readability of questions is determined by the score attribute which is the difference between upvotes and downvotes received for the question.

The original Posts table has a total 23 columns including *Body, Title, ViewCount, CommentCount, FavouriteCount, LastActivityDate, OwnerId, OwnerName, AcceptedAnswerId* etc. However, only relevant columns are selected while querying on the table, and data is gathered. We have identified the relevant features of the given data and selected columns accordingly hence most of the feature selection activity is covered during this stage of data collection.

The *PostTypeId* classifies the post into question or answer, such that if the value is 1 then the post is question type else if the value is 2 then the post is answer type. The objective of this study is not to analyze the entire dataset but to find the users' who can potentially give the acceptable answer to the question. Hence only questions with accepted answers are selected. Also, to maintain high quality and readability, questions with positive scores are selected. The generated result in stack-overflow data explorer is available to download in CSV format. At a time only 50,000 query records are available for viewing and downloading, hence month wise batches are created to collect more data and all the csv files results are combined for further data preprocessing.

The attributes selected while gathering data from Stack overflow is shown in Table 1. This table gives details about the description of the features in the gathered dataset as well explains how these features are used while actual implementation of the algorithm. Some features of the dataset such as *OwnerUserId*, *Score* of the question, are not used in further implementation and will be removed in later stages of data preprocessing activity.

Data Attribute	**Description**	**Use**
Id	A unique identifier of a question	Unique identification
Title	Title of a question	Input to the model to create similar clusters
Body	Description of a post with code samples (if any)	Input to the model to create similar clusters
Tags	Original Tags of question	Validate the created clusters by matching the tags and top frequent words in cluster
Creation Date	Creation date of a question	Used to fetch the active user when this question is posted
Score	Upvotes - downvotes	Question score is not used and will be removed in further processing
OwnerUserId	Unique identifier of a user who posted question	Question score is not used and will be removed in further processing
Accepted AnswerId	A unique identifier of accepted answer to this question	Answer unique identification
AnswerOwnerId	Unique identifier of a user who posted answer	This Id will be used while recommending users to new question
AnswerCreationDate	Date when answer of this question is posted	To check recency of user while recommendation
AnswerScore	Upvotes – downvotes of answer	To check quality of the answers given by user's

Table 1: Data attributes and description

3.2 Data Filtering

Over the SO platform questions of vast variety of skills are posted every day. While posting a question, the user gives a tag to identify the skill. Currently SO has 61,000 unique tags [1]. After analyzing the dataset, we have found that there is imbalance in the data because of rarely used tags or wrong selection of tags by users. Also, the questions of most famous skills such as python, JavaScript are asked more frequently compared to the rare skills questions such as *Xamarin, instagram-api*, etc. This imbalance increases bias in the data hence to minimize the bias and make processing faster we have further reduced the size of the dataset by taking the most frequently occurring tags. We have taken the records of the top 1000 frequent tags from the dataset. Figure 2 shows the top 30 most frequent tags from the selected list.

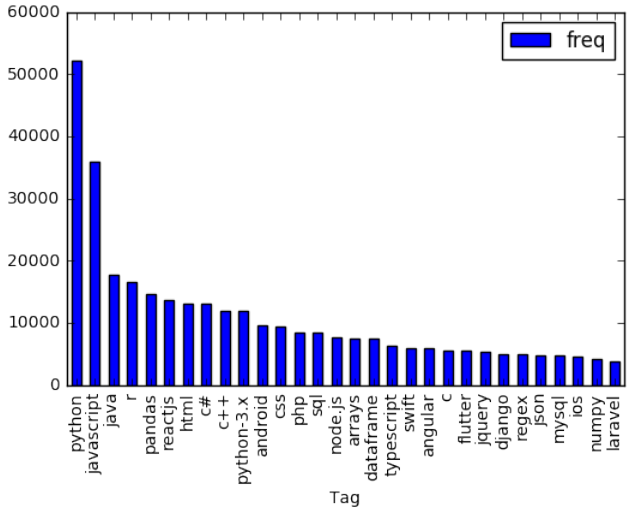

Figure 2: Top 30 most frequently used tags

Table 2 gives the information of the data sample selected for further analysis of this experiment.

Total number Questions	**Unique Answerers**
83156	24433

Table 2: Selected Sample of Dataset

3.3 Data Pre-processing

In the stack overflow dataset, each question is considered a distinct document. Each document attribute has more textual information in *Title* and *Body*. Hence data cleaning and data preprocessing activity have been performed on the textual data. Below activities are performed for data cleaning and pre-processing:

1. Removal of Null/Empty data: The null or empty data check activity is performed on all columns of the dataset and we have found that column OwnerUserId and AnswerOwnerUserId contained some null values. The values in these columns are very critical for the further implementation of the project. Hence these documents are removed from the dataset.

2. Code block removal from Body: While doing data analysis we found that the code mentioned in any post is enclosed within the <code> </code> tag. This code block doesn't add meaningful value while taking the decision of a finding potential expert, because the question may contain a similar problem but different code implementation and syntax. We are aiming to find the similarity between questions based on the question explanation and title of the question. So, the code blocks within the body of the question have been removed.

3. Html Tag Removal: The question-answer information in stack overflow is stored inside the *Html* tags. These tags have been deleted while data cleaning using *BeautifulSoup* python library.

4. Hyperlinks Removal: Hyperlinks inside the question body and title have been removed.

5. Regular Expression and Normalization: A number of activities have been performed for text data normalization, including the lower-case conversion, space regularization, punctuation and abbreviation removal, removal of extra spaces. While removing punctuations special care has been taken to retain the tags such as *c++, c#, .net,* etc. The built-in Python functions and regular expression library functions are utilized to successfully carry out these tasks.

6. Stop Word Removal: Some of the English common words such as 'is', 'are', 'a', 'an', 'the' etc. excluded from the title and body attribute using NLTK library, so that more focus can be given to those words which constitute the decision making.

8. Part of Speech (POS) Tagging: POS tagging is a very famous natural language processing (*NLP*) technique which is performed at the token level. This method assigns the part of speech to individual words in a sentence. Many verbs and adjectives such as performed, analyzed, used, extremely, kindly, appreciate etc are very common in most of the questions on the SO platform and this may lead to incorrect formation of clusters hence by selecting the singular and plural noun forms from the input text such words are removed. After implementation of POS tagging method, we have seen a huge improvement in the performance of models.

9. Extended Stop words removal: We have taken additional steps to remove the most common words in SO question data to further improve the accuracy of the clusters. Some words such as input, output, error etc. are very frequent in each SO question and getting rid of such words also becomes important to form well defined clusters of similar topics. Hence by extensive data analysis and multiple cluster evaluation we have identified the extended words to be removed from the SO input text data.

7.Tokenization and Lemmatization: To further reduce the complexity of the text data, Lemmatization has been performed on the documents. Lemmatization uses the lexical knowledge base like *WordNet* to obtain the correct base forms of words.

8.Float to Int type conversion: The *OwnerId* and *AnswerOwnerId* columns contained data in the decimal format. These column data have been converted into integer format.

9. Tag Data cleaning: The *Tags* column which will be used for evaluating the accuracy of clusters needed some data preprocessing. Tags where enclosed within the <, > brackets hence those brackets were removed from the data and extra spaces between the tags is also removed. The resulting tag data is then ready for comparison with the cluster keywords.

10. Combining Title and Body: The *Title* of the question holds the important information about the question along with the *Body*. Hence, both of these columns are combined into one as *QuestionText*. This *QuestionText* will be the final column after applying all the NLP data cleaning techniques and this column is passed to the model as input.

3.4 Data Visualization

This section provides the visualization of the sample data objects after pre-processing to obtain a better perspective of the dataset.

Figure 3: data object on stack-overflow platform [11]

Data Attribute	Sample Data Object
Id	59550804
Title	melt column substring column name panda python
Body	data frame want melt data frame look meaning melt-based delimiter column name best way
CreationDate	2020-01-01 07:44:54
Score	2
OwnerUserId	6057371
AcceptedAnswerId	59550967
AnswerOwnerId	11232091
AnswerCreationDate	2020-01-01 08:19:30
AnswerScore	1

Table 3: Data attributes and sample data object after data pre- processing

Figure 3 shows an example of the post in the stack-overflow platform, where the data is stored with Html tags, code blocks and non-ascii characters. On the other hand, Table 3 illustrates the same data object after applying all the pre-processing techniques.

4 METHODOLOGY

4.1 Project System Design and Algorithm

4.1.1 System Flow

We propose a new expert finding model by evaluating the performance of users on the SO platform. Figure 4 illustrates the proposed system design of the Expert Recommendation model.

After performing data processing and in the next phase of the project we are planning to perform the below steps using 5-fold cross validation:

1 Clustering of similar questions: Using Machine learning clustering techniques clusters of the similar questions created.

2. Finding similar clusters: For the question in test data, the closest matching cluster is predicted from the created model.

3.Filtering based on Created Date of question: From the closest matched cluster the data is filtered to select only those users' who are most active when question is posted. Users' who have posted answers within the two months of the timeframe when the question is posted are selected.

4.Ranking Based on the Score: In this step, the clustered data is ranked according to the highest score of the answers given by users.

5.Recommending top K Answerer: After getting a list of ranked users, we are recommending top 50 users as a potential answerer to the question.

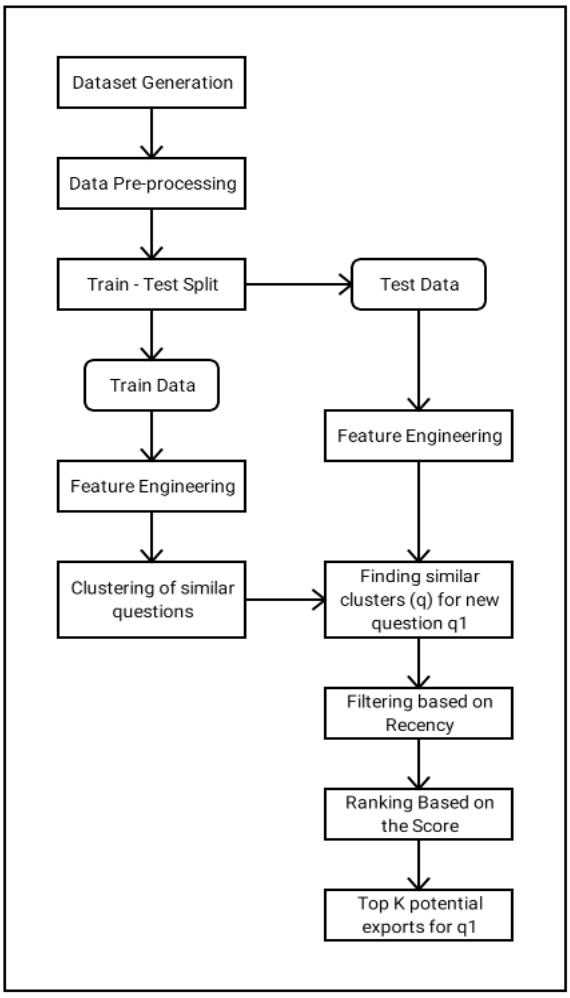

Figure 4: Proposed flow diagram for expert recommending system

4.1.2 Proposed Algorithm

Find below the algorithm of the expert recommendation system:

Input: Training Set(t)
Output: Top Experts (userIds[])
Function getRecommendedAnswerers()

C **=ClusteringAlgo**(QuestionsText)

getTopTerms = getTopTermsInCluster(C)

For each Q in Testset
BestCluster=**C.Predict(**Q_{Body}**)**

FilteredQstns=**getFilteredData**($Q_{CreationDate}$)

SortedCluster=**SortOnScore**(FilteredQstns)

$Q_{recommendedAnswerer}$=**GetTopAnswerers**(SortedCluster)

4.2 Feature Engineering

In this section we will discuss the aspects of feature engineering carried out during implementation of this project. After performing the data preprocessing activity as discussed in section 3.3, we have acquired the raw clean data in text format. However, for any machine learning model it is difficult to interpret the unstructured text data. To build intelligent machine learning models we need to feed the structured and numerical vectorized data by removing the noise from original unstructured textual data. This process of transforming textual data into numerical vectors is one of the feature engineering techniques. There are various feature engineering strategies available such as the Bag of Words model, Bag of N-Gram model and TF-IDF model. In this project we worked with the TF-IDF model.

4.2.1 TF-IDF Vectorization

In the SO dataset, we have combined the Title and Body information into a single column as *QuestionText*. When we combine all the *QuestionText* data it creates a very large corpus. Moreover, many terms in the corpus are frequently occurring across all the documents. If we employed the Bag of word model in such a type of corpus it is highly possible that frequently occurring terms overshadow the other features in the dataset [12]. Hence, we decided to work with the TF-IDF model which resolves this issue.

TF-IDF stands for Term Frequency-Inverse Document Frequency and mathematically represented as TF-IDF = *tf x idf.* The detail formula for TF-IDF calculation is given in the Figure 5:

$$tfidf(w,D) = tf(w,D) \times idf(w,D) = tf(w,D) \times log\left(\frac{C}{df(w)}\right)$$

Figure 5: TF-IDF calculation formula [12]

Here (*w,D*) denotes word w in document D.

4.3 Clustering models

This section will explore details of the clustering and topic modelling methods applied to create clusters of the similar questions.

4.3.1 K-Means Clustering

The textual content of the questions is investigated to determine the similarity between the questions and group the most similar questions together. Here, we have applied the K-Means Clustering technique to form the clusters of the question.

K-Means clustering is an Unsupervised learning algorithm, used to perform clustering of the data. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to only one group that has similar properties [13]. Here, k represents the number of clusters to find in the data and is passed as an input parameter to the model. K-Means clustering starts with selecting random k points as centroid and assigning each point to the closest centroid to form the k clusters. Further, it calculates the variance of each cluster and decides the new centroid and reiterates the step one again until the well-defined clusters are not formed.

The performance of the K-Means clustering is highly dependent on the number of clusters it forms. Hence choosing the right value of k is very essential to achieve good results with the K-Means clustering. There are various methods to decide the value of k, However, we have used the most appropriate method which is the Elbow method.

The concept behind the Elbow method is, it executes the K-Mean clustering for a range of k values. For each k value the K-Means model gives the Sum of squared error (SSE) in the inertia_ parameter. These SSE values are collected for each value of k and then plotted on the graph. The sharp point of bend which looks like an arm is considered as the best value of K. As seen in Figure 6, we have run the Elbow algorithm for 10-40 values of k. Here, we can see at a value of k=25 there is a slight bend in the curve. Hence, we have used the value of k to find the best results of K-Means clustering which have been determined through Elbow method.

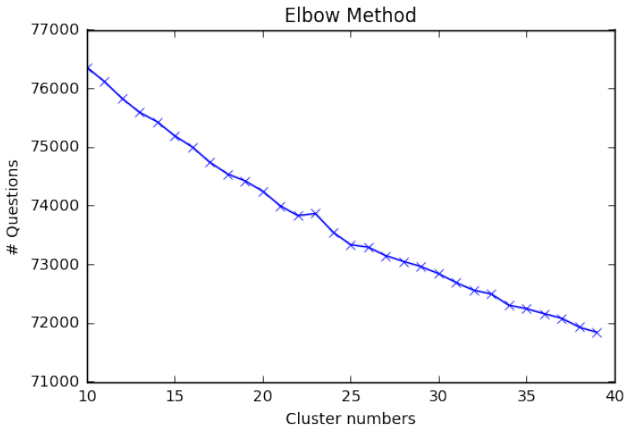

Figure 6: Elbow method to determine best clusters

4.3.2 Latent Dirichlet Allocation (LDA)

LDA is a one of the most popular techniques of the topic modelling in Machine learning. Topic modelling is an Unsupervised machine learning technique which is used to find out the different abstract topics in a text document.
LDA is a statistical generative model, which says each word in a document comes from a topic and the topic is selected from a per-document distribution over topic [14]. We implemented the LDA model using scikit-learn library to find the topics of SO questions. Along with finding topics of the input data LDA model gives the probability distribution of each topic to every

document. The topic which has dominant probability is selected as the topic label for that document. Figure 7 illustrates the topic probability distribution for a single document d_1. From the plotted graph we can visualize that topic 10 has the highest probability for this document and hence document d_1 is clustered into topic 10 cluster.

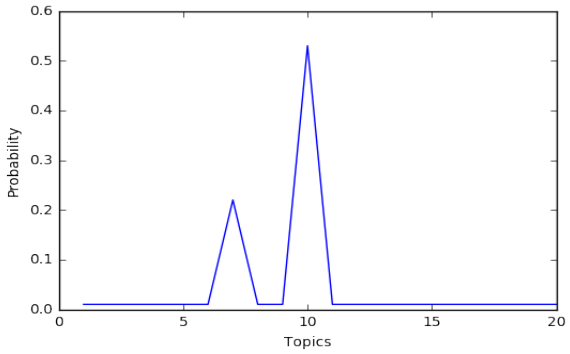

Figure 7: Topic probability distribution for a document

Along the same lines of K-Mean clustering LDA model also needs to give a number of topics as an input. To get the distinct set of topics choosing the right value of topics is an important step. SO, how to choose the correct number of topics? To decide the optimal value of k we have performed hyperparameter tuning using *GridsearchCV*. We pass the predefined values of hyperparameters to the *GridsearchCV* and it tries all combinations of values passed in the parameter dictionary and evaluates the model for each combination using the Cross-Validation method. Using the results of the *GridsearchCV* we can choose the best model.

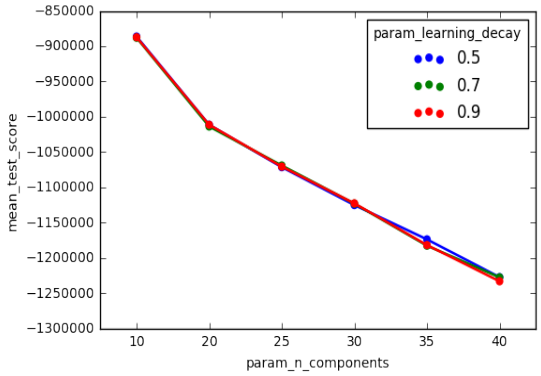

Figure 8: GridsearchCV results

We can see the plotted results of *GridsearchCV* in Figure 8, where the model shows best results when the value of K=10. However, if cluster size is reduced in the expert finding model then the number of users unique in the cluster increases which may cause the low accuracy while recommending the appropriate users hence, we have experimented expert finding model using LDA with choosing number of topics as 10,20,30.

5 EXPERIMENTAL RESULTS AND DISCUSSION

In this section we will present the results of our experiment for the different parameter settings. Firstly, we will present the results we got by applying the K-Means Clustering and then by LDA topic modelling technique. Furthermore, we will compare the results of both the methods and discuss the evaluation metrics and results received by other researchers.

5.1 K-Means Clustering Results

We have implemented the K-means clustering for the best value of cluster i.e., K=25 found using Elbow method. After implementation of K-Means we have got 25 clusters of topics. Some examples of topics extracted using K-means clustering are shown in Table 5.

To visualize how effectively K-means clustering algorithm recommends the user to post a question we have created a word cloud representation of one of the topics formed by K-means algorithm in Figure 9. The word cloud gives pictorial representation of most dominating words in a topic. From Figure 9, we can classify the topic as SQL server database because the most frequent words are *server, select, database, table, query, mysql, sql, etc.*

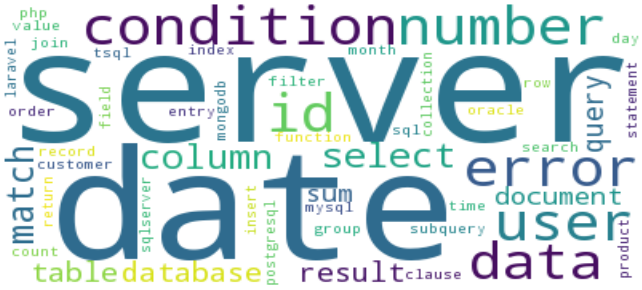

Figure 9: Word-cloud representation of topic 1 in LDA

Recommended Answerer	Tags
161127428	sql sql-server tsql
1491895	mysql,php, javascript
11565629	sql oracle
1509264	sql oracle pl sql
9097906	oracle pl sql
1144035	sql postgresql
17389	php mysql
1422451	sql ms-access
12232340	php mysql

Table 4: Top-10 recommended users for a question of topic 1

We have taken the sample question which belongs to the topic SQL server and listed down the top 10 recommended users to that question along with the actual tags of the previous questions from training data which are answered by the recommended users. The results are plotted in table 4. From this table we can claim that almost all the top 10 recommended users had knowledge of the topic SQL-Server database and they can answer the posted question.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
data	button	query	regex	file	cell	document	class	plot	date
frame	click	table	match	json	sheet	mongodb	method	matplotlib	day
column	javascript	sql	character	csv	googlesheets	collection	instance	ggplot2	month
panda	html	mysql	string	python	vba	field	java	bar	column
value	radio	result	word	directory	column	query	function	data	panda
dataframe	jquery	column	expression	folder	excel	aggregationframework	constructor	graph	data
row	change	record	pattern	text	value	mongodbquery	object	color	format
python	page	data	group	script	formula	mongoose	cplusplus	label	time
time	function	join	number	error	google	array	property	point	dataframe
json	css	value	python	line	row	aggregation	type	python	value

Table 5: Top-10 Keywords of topics in K-Mean Clustering

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
regex	django	date	query	kera	terraform	column	html	plot	stream
character	view	month	sql	model	prolog	dataframe	cs	matplotlib	java
string	swiftui	day	table	tensor	resource	panda	javascript	ggplot2	javastream
word	model	time	mysql	tensorflow	amazonwebservices	value	button	chart	java8
match	swift	pyspark	record	layer	instance	row	jquery	bar	kafka
number	io	year	result	pytorch	predicate	data	image	graph	topic
expression	image	hour	column	shape	queue	python	element	color	consumer
file	error	datetime	value	kotlin	aws	group	page	docker	lodash
space	button	format	postgresql	studio	rule	frame	div	label	map
python	field	week	oracle	error	terraformprovideraws	list	text	data	age

Table 6: Top-10 Keywords of topics in LDA

5.2 LDA Results

We have implemented the LDA topic modelling using a number of topics as 10,20,30,40. We have received the best performance with the number of topics as 25. The Table 6 illustrates the 10 topics of this model. In this section we have discussed the results of the best performed LDA model.

We have followed the same steps as K-means clustering and plotted the word cloud of one of the topics of the LDA model as seen in Figure 10. The Figure 10 clearly represents the topic as Swift-UI because of the high frequency words such as screen, preview, swiftui, navigation, color, swift, etc.

Figure 10: Word-cloud representation of topic 1 in KMean

Table 7 shows the list of recommenders along with the original tags for the sample question of topic *swift-ui*. The model shows good performance by recommending 8 out of 10 users belonging to the tag *swift-ui.*

Recommended Answerer	Tags
97337	swift swiftui
7948372	ios swift swiftui
12299030	ios swift swiftui
341994	ios swift xcode
5623035	swift xcode swiftui
7761488	ios swift swiftui
1104384	flutter
3664021	Go
5777554	ios swift swiftui
1780632	ios swift uitableview

Table 7: Top-10 recommended users for a question of topic 2

5.3 Evaluation

Evaluating the accuracy of expert finding algorithms is a very difficult task because the number of users who have given answers to at least one question is huge. In the used dataset of total 81,678 questions there are 24,184 unique answerers who have given the accepted answers. Although over the SO platform one question has more than one answerer, but in this project, we are working on only accepted answerers to give accurate results. Hence from the list of 24,184 labels predicting the correct answer becomes almost impossible for any model. Hence, to evaluate the accuracy of the model we are not doing the matching of recommended answerers and original answerer. The accuracy of the algorithm is evaluated by matching the original tags of the question and top tags of the cluster to which recommended answerers belong. We called this metric as Tag similarity.

To give better performance, it is crucial to form the well-defined topic clusters, because the further algorithm of the expert recommendation model is based on the clusters formed by the models. Table 5 and table 6 shows some of the extracted topics of the K-means and LDA model respectively. The purpose of these two tables is to intuitively demonstrate that the topics extracted using the LDA model are superior compared to the ones extracted using the K-Means Clustering model.

5.3.1 Tag Similarity Metric

To calculate the similarity of the tags we have developed a formula using the concept of Jaccard Similarity. Jaccard Similarity is used to measure the similarity and dissimilarity of the sample. On the other hand, in our problem we want to measure how close the original tags are with the predicted cluster tags. To calculate the similarity, we have used the below formula:

$$\textbf{\textit{Tag similarity}} = \frac{(\textbf{\textit{original tags}} \cap \textbf{\textit{Top 50 tags of predicted cluster}})}{\textbf{\textit{number of original tags}}}$$

The above formula gives us information of how many of the original tags are present in the cluster of the predicted tags.

The Figure 11 illustrates the tag similarity performance of the K-Means cluster model and LDA model. The performance of both the models is plotted for different values of k such as 10,20,22,25,30,40. LDA model gave better performance than the K-means model.

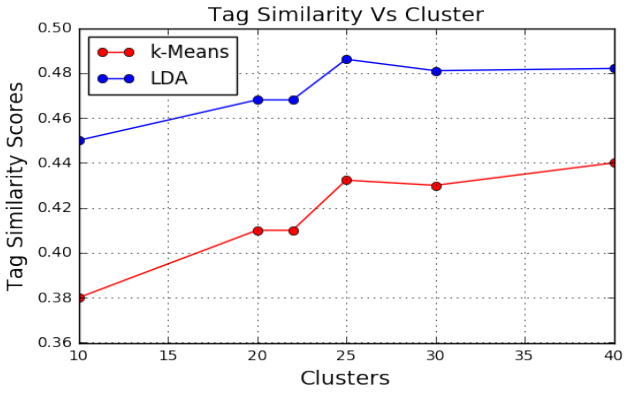

Figure 11: tag Similarity of LDA and K MEAN

We can also validate the best performance of the clusters selected by visualizing graphs of figure 11. The optimal number of clusters is selected as 25 by the Elbow method. Not only the K-means cluster model but also the LDA model performed well when the number of clusters selected was 25.

The LDA model gave the highest Tag Similarity score of 0.49 whereas the highest score of the K-Means cluster model was 0.43.

5.3.2 User Similarity Metric

Although predicting the original accepted answerer as a recommended answerer was not feasible for the implemented expert recommendation model, we have tried to analyze how many times the original accepted answerer was in the list of top 50 recommended answerer and found some interesting results.

To calculate the user similarity metric, we have introduced a column in the test dataset as *UserSim*. Whenever the original answerer was present in the list of recommended answerers, we made the value of column as 1 and otherwise the value remained as 0. Finally, to get the User similarity metric value we have taken the mean of the column *UserSim*.

The trend of predicting the actual users in the list of recommended users seems unusual from the Figure 12. We have received the highest value of user similarity as .20 for the LDA model when the number of clusters was 30. On the other hand, for K-means clustering the user similarity value was around 0.18 for optimal cluster value 25. The K-Mean algorithm showed an increase in the value of user similarity as the number of clusters increases after 30.

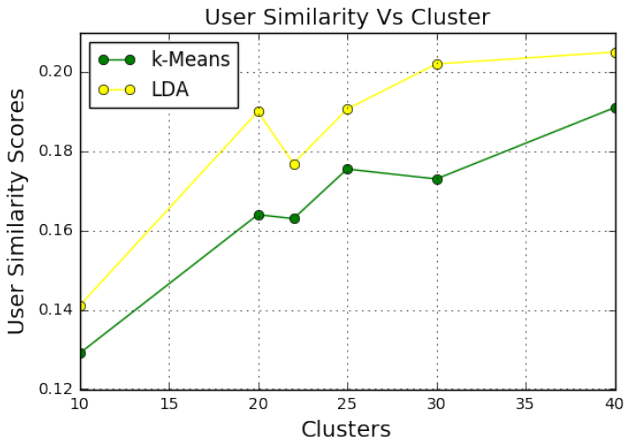

Figure 12: User similarity Comparison

For the User similarity metric as well the LDA model performed better than the K-means clustering model.

The metrics to evaluate the performance of the model are introduced as well as the dataset is also gathered manually using the data explorer tool of the SO. Hence, direct comparison with the work of previous researchers is not possible for this project. However, the comparison study of LDA and K-means clustering for SO expert recommendation systems has not been done previously. Our study proved that the LDA topic modelling technique gives better results for SO expert recommendation systems.

6 CONCLUSION AND FUTURE WORK

Due to the increasing popularity of the SO platform, there is a need to have automation in the system when a new question is posted, so that question can be addressed quickly by the correct user. Our expert recommendation model can be adopted by the SO community so that newly posted questions will be routed to the potential answerers who possess the skills required to answer that question.

We focused on finding answers to the two main questions while finding the experts to the new question, 1) Does the user possess the required skills to answer this question? 2) Is the user interested in giving answers to this question?

While implementing the expert recommendation system for any CQA platform most of the previous studies have worked on the first question to give recommendation. On the other hand, resolving both questions to develop the recommendation system makes this project unique.

First question is resolved by creating clusters of similar SO questions using the topic modelling technique or clustering technique and then by matching the skills of the new question with all the clusters to get the users as potential answerers from the closest matching cluster.

Finding the answer of the second question is important because even though the user has the right skills to answer the question but if that user is currently not active in giving answers then recommending that user as potential answerer is meaningless. Hence, we retrieve those users who have given answerers within a certain time frame when a new question is posted and then based on the scores of answers given by them, we perform ranking. Hence, our recommendation model gives more meaningful recommendations.

Our results indicate that the LDA model outperforms the K-means clustering model in retrieving a well-defined set of topics from the textual data. Hence, these results suggest that the Statistical topic modelling techniques can replace the traditional clustering techniques such as the K-Means clustering.

In many CQA websites such as SO, users can post multiple answers to the question. Although all the posted answers to the questions are not accurate, many answers are correct in different scenarios or environments. This expert recommendation model considers those users as experts whose answers were accepted by the asker. If we extend this model to consider all the users as experts who have given the quality answers even though those answers are not accepted by the asker, then there would be huge improvement in the performance of this model.

There are active users in the SO community who possess versatile skill sets. Suppose user u1 is one such user who has answered the multiple questions of python as well as Java. The user u1 has gained score of 10 by answering the questions of python and has received score of 5 by answering the Java question. However, the total score of this user is 15 and hence if user u1 is part of the java cluster then the rank of this user is higher than the user u2 who has gained score of 14 by only answering the java questions. In future scope, we need to handle this limitation to perform ranking by calculating the score based on individual skills.

In the collected dataset there are a higher number of Python and JavaScript questions this creates the bias in the dataset. Due to bias in the data multiple clusters of Python and JavaScript get created and which impacts the accuracy of the model. If the bias in the dataset is reduced, then the results would be more encouraging.

The proposed expert recommendation system model implantation has some limitations and hence it has not been evaluated using the standard matric evaluation methods. It would be interesting to see how it performs by using standard evaluation metric and then it would be possible to compare the results with previous implementation of SO expert recommendation system.

In our future work, along with resolving above limitations in the system, we would also like to find out the influential users over the SO community by creating the social network of the users.

REFERENCES

[1] "Stack Overflow - Where Developers Learn, Share, & Build Careers." https://stackoverflow.com/ (accessed Mar. 16, 2021).

[2] B. Shao and J. Yan, "Recommending answerers for stack overflow with LDA model," ACM Int. Conf. Proceeding Ser., vol. Part F1311, pp. 80–86, 2017, doi: 10.1145/3127404.3127426.

[3] Z. Yang, Q. Liu, B. Sun, and X. Zhao, "Expert recommendation in community question answering: a review and future direction," Int. J. Crowd Sci., vol. 3, no. 3, pp. 348–372, 2019, doi: 10.1108/ijcs-03-2019-0011.

[4] A. Verma, N. Sardana, and S. Lal, "Developer Recommendation for Stack Exchange Software Engineering Q&A Website based on K-Means clustering and Developer Social Network Metric," Procedia Comput. Sci., vol. 167, no. 2019, pp. 1665–1674, 2020, doi: 10.1016/j.procs.2020.03.377.

[5] M. Choetkiertikul, D. Avery, H. K. Dam, T. Tran, and A. Ghose, "Who Will Answer My Question on Stack Overflow?," no. July, pp. 155–164, 2015, doi: 10.1109/aswec.2015.28.

[6] P. Sumanth and K. Rajeshwari, "Discovering top experts for trending domains on stack overflow," Procedia Comput. Sci., vol. 143, pp. 333–340, 2018, doi: 10.1016/j.procs.2018.10.404.

[7] N. Zhao, J. Cheng, N. Chen, F. Xiong, and P. Cheng, "A Novel Expert Finding System for Community Question Answering," Complexity, vol. 2020, 2020, doi: 10.1155/2020/5346085.

[8] D. Van Dijk, M. Tsagkias, and M. De Rijke, "Early detection of topical expertise in community question answering," SIGIR 2015 - Proc. 38th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 995–998, 2015, doi: 10.1145/2766462.2767840.

[9] B. Li and I. King, "Routing questions to appropriate answerers in Community Question Answering services," Int. Conf. Inf. Knowl. Manag. Proc., pp. 1585–1588, 2010, doi: 10.1145/1871437.1871678.

[10] "Query Stack Overflow - Stack Exchange Data Explorer." https://data.stackexchange.com/stackoverflow/query/new (accessed Feb. 09, 2021).

[11] "dataframe - melt column by substring of the columns name in pandas (python) - Stack Overflow." https://stackoverflow.com/questions/59550804/melt-column-by-substring-of-the-columns-name-in-pandas-python (accessed Feb. 09, 2021).

[12] "Traditional Methods for Text Data | by Dipanjan (DJ) Sarkar | Towards Data Science." https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41 (accessed Mar. 16, 2021).

[13] "K-Means Clustering Algorithm - Javatpoint." https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning (accessed Mar. 16, 2021).

[14] "Topic modeling using Latent Dirichlet Allocation(LDA) and Gibbs Sampling explained! | by Ankur Tomar | Analytics Vidhya | Medium." https://medium.com/analytics-vidhya/topic-modeling-using-lda-and-gibbs-sampling-explained-49d49b3d1045 (accessed Mar. 16, 2021).

[15] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios, "Finding expert users in Community Question Answering," WWW'12 - Proc. 21st Annu. Conf. World Wide Web Companion, no. February 2015, pp. 791–798, 2012, doi: 10.1145/2187980.2188202.