Recommending Batch of Experts to Questions on Stackoverflow

By

Aiman Muzaffar

Supervisor

Dr. Hammad Afzal

A thesis submitted in the Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan for the partial fulfillment of the requirement for degree of MS in Software Engineering.

August 2022

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by **Aiman Muzaffar, Registration No. 00000317804**, of Military College of Signals has been vetted by undersigned, found complete in all respect as per NUST Statuses/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/MPhill degree. It is further certified that necessary amendments as pointed out by GEC Members of the student have been also incorporated in the said thesis.

Signature:
Name of Supervisor: **Assoc Prof Dr. Hammad Afzal**
Date:

Signature (HOD):
Date:

Signature (Dean):
Date:

Acknowledgments

First and foremost , I want to thank Allah Almighty for the strength and his blessing in completing this thesis. I would like to convey my gratitude and special thanks to my thesis supervisor Dr. Hammad Afzal, PhD, for his continuous support, monitoring and guidance. Without his assistance and dedicated involvement in every step throughout the duration, this work wouldn't have been completed. His incomparable help for constructive comments as well suggestions for all assessment tasks and thesis are major contributions to the success of this study. I would like to thank him very much for his understanding over this past duration. Also, I would like to thank the members of my committee; Assoc Prof. Naima Iltaf, and Asst. Professor Dr Bilal Rauf for their support on this topic.

Most importantly, none of this could have happened without the unceasing support and attention of my father (M. Muzaffar Sultan) and my mother (Iffat-Un-Nisa). I would also like to thank my siblings for their constant encouragement and my friend Saba Siddique for helping me with the submission process.

Dedication

"In the name of Allah, the most Beneficent, the most Merciful"
I dedicate this thesis to my parents, siblings, friends and teachers who supported me each step of the way.

Abstract

In order to assist the process of questions answering on CQA (Community Question Answering) websites, this paper proposes an improved methodology of batch recommendation of answerers (experts) to questions called BESF (BERT Expert Recommendation using Multi-Objective Sailfish Algorithm with Genetic Algorithm). First, experts and questions modeling is done using BERT Topic modeling technique, which creates clusters on the base of topics. Using TF-IDF values calculated by BERT, Question-Expert similarity, Question-Topic similarity and New-Old questions similarity are calculated, which helps in classification of new questions. Using the calculated similarities in each cluster, experts are ranked on the base of four basic parameters, i.e. reputation, past performance, recent activity and activeness. Keeping in view the bounded number of experts and avoiding duplicate answers to repeated or similar questions, this methodology optimizes three parameters i.e. increased question coverage, increased answerability and decreased expert resources utilization. This becomes a multiobjective optimization problem and MOSFO-GA (Multi-Objective Sailfish Optimization with Genetic Algorithm) is used to address this problem. The proposed approach is evaluated on StackOverflow dataset which shows that using BERT for topic modeling and clustering, gives better clustering results as well as increases the performance as a whole, in comparison with using MOSFO-GA for clustering. This approach can be helpful in time conservation of users and providing better answers to questions by recommending batch of experts to answer the questions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Problem Statement

The Community Question Answer (CQA) websites gives a vast variety of opportunities for users to provide, search and share information. Although the idea of getting a straightforward, direct answer to a question sounds very appealing, the quality of the question itself can have a profound effect on the chances of finding useful answers. High quality questions improve CQA knowledge and are therefore important for CQA forums to better understand what raises the most interesting questions in the forum community. The web has changed the way people used to offer knowledge and information sharing. It is advisable to drop the keywords in the search engine to display a need, and the search engine quickly calculates a large number of highly relevant or important web pages from which the user he can choose. However, search results may not provide results a straightforward solution to a user's problem and it may take some time to update them all, without having to be sure to find the answer you want. Websites that answer Community Questions provide a new opportunity to find what you are looking for information in a fast and efficient way. But still there is a lot of room for improvement of these CQA websites to provide better responses to questioners. This study provides an improved way of recommending batch of experts to a question on any CQA website for the timely and good quality answers.

1.0.1 Motivation and Problem Statement

A recommendation system for recommending batch of expert to questions can be very helpful to users in giving them timely response of their queries and hence improving their performance in whatever work they do. Using the batch recommendation technique helps decreasing the answer providing time, as many experts are being recommended to answer a particular query at same time. The expert finding parameters used can also be proven helpful to be used for finding experts on other question and answer websites too. Combing the two techniques, BERT topic modeling and MOSFO-GA, gives high results for answerability, coverage and less consumption of expert resources.

1.0.2 Objectives

The main objectives of this thesis are listed below:

- An improved way of assisting questions find their respective best answerers is proposed, which integrates BERT Topic Modeling and MOSFO-GA. The most efficient use of limited expert resources can be made while providing best suitable answer to a question.
- An enhanced way of ranking an expert on a Question and Answer platform has been introduced, which considers user's activeness on a website, past performance of answering questions, recency in answering a question and reputation on website.
- A novel way of ranking the experts on StackOverflow is proposed, discerning their reputation, activeness, past performance and recent activities.

1.0.3 Thesis Contribution

An improved way to help queries find its best respondents is suggested, which includes BERT Topic Modeling and MOSFO-GA. The most effective use of limited professional resources can be made while providing the most appropriate answer to the question. An advanced expert evaluation and ranking methodology for using in the Questions and Answers forum has been introduced, which takes into account the user's previous performance in answering questions, activeness in providing answers on the website, the timeliness factor in answering the question and dignity or reputation on the website. A new way of rating professionals in StackOverflow is proposed, taking into account their reputation, performance, previous performance and recent activities.

1.0.4 Thesis Organization

1. Chapter 2 gives an overview of the problem which is targeted in this thesis work. It also describes a review of the literature.
2. Chapter 3 provides a brief description of work related to this topic in the past.

3. Chapter 4 explains the BERT technique and its implementation for topic modeling.
4. Chapter 5 explains the process used for ranking the experts.
5. Chapter 6 provides explanation of Sailfish Optimizer and how it is being used for optimization of three parameters, i.e. resource consumption, coverage and answer-ability.
6. Chapter 7 discusses the process of formulation of problem. It also discusses the results.
7. Chapter 8 gives the conclusion of thesis work.
8. Chapter 9 lists the references used.

Chapter 2

Literature Survey and Related Work

Community Question Answering (CQA) websites provide a good environment to users where they can publish answers and ask questions on various subjects [6]. CQAs like Stack Overflow, Quora and Yahoo Answer!, have become an important knowledge repository and have allured much attention [7]. For example, in 2019, Stack Overflow added over and 2.6 million new questions, 2.8 million answers and more than 1.7 million new users to join the community [8]. Given the superabundant information stored and an influx of visits, provide relevant content for encouraging engagement and satisfying user searches within the community is crucial for a CQA website. For doing so, such platforms must provide recommendation services to users or experts of a particular topic so that a question gets the best possible answer. Although such CQA portals significantly benefit the users, there still exist various cons in current CQA systems. [9]

Most of the previous works have focused on identification of users who are most likely to provide a good quality answer or simply to provide an answer to a question, but the promptness and rapidity of the response is also an important factor for satisfaction of a user. P. Hansen et al. propose Neural network and Point process based algorithms in [10] for prediction of 3 tasks regarding response of a user to a question that whether or not the user answer, the net total votes that the answer will get and the time before that answer. These algorithms train over 20 features set they define for every pair of question and user which quantify both structural and topical aspects of the Q&A platforms, including social centrality measures and discussion post similarities. P. Hansen et al. used Sparse factor analysis (SPAFRA), Poisson regression (PR) and Matrix factorization (MF) for this purpose in [10]. The presence of a poor quality answer in a Q&A forum, indicates that there is an unqualified or unprofessional. Therefore, it is important to find reputable users or experts. Most of the current expert-ranking methods consider only the basic features, like answers quantity provided by a user, ignoring the consistency and quality of the user's answer. M. S. Faisal propose Exp-PC in [11] which adapts G-index for ranking experts. Rep-FS uses several features like vote ratio and voter's reputation in order to measure expertise level. Weighted Exp-PC combines Exp-PC and Rep-FS scores hence quality of an answer is determined by expert and their consistency in providing good quality answers. A. Diyanati et al. propose in [12], Analyzing the scores of answers and questions keeping in view the fact that an expert will ask question with higher scores and comment mining by scoring negative and positive words present in a comment. For facilitation of Q&A social websites, M. Li. et al. propose batch recommendation of answerers by optimization of expert resources utilization in [13]. Modeling of questions is done with the bi-term topic model (BTM), clustering of answered questions is done for topic distribution. Multi objective sailfish-genetic algorithm (MOSFO-GA) is constructed by integration of SFO and GA and using Jensen – Shannon divergence, Roulette-wheel method and finding Non dominated Pareto optimal solutions. Then, experts in each domain of knowledge are ranked on basis of their recentness, activeness and professionalism. H Wang et al. presents in [14] QAP, which is a promoter for answering the questions for CQA websites. The QAP makes it easy to use the archived filtered answers which are regarded as important knowledge and the experts which are recommended are taken as a source of knowledge the target questions. The QAP used HIT algorithm, agglomerative hierarchical clustering and H-SVM. Bottom-up multi-path evaluation (BUME) is used to verify the consistency of hierarchy and AHC (agglomerative hierarchical clustering) was used with the modification of a single-link for grouping the questions having issues similar to the targeted ones.

It is important for services like CQA to get good quality answers to the questions for maximizing the process of benefiting. However, people are generally considered experts only to their specific areas. J. Wang concerns in [15] with the issue of recommendation of experts for a new question post, reducing the waiting time of a questioner and improving the question quality in order to improve the level of satisfaction of whole CQA community. J. Wang proposes the use of CNN (Convolutional Neural Network) in [15] for solving this issue and used TF-IDF for term frequencies of the text in questions and answers, NLP for the basic pre-processing LDA (Latent Dirichlet Allocation), STM(Segment Topic Model), SSR, 1-max pooling operation and LR Classifier There is a massive amount of user-generated data for questions and Answers which is a valuable source of knowledge. But an issue is how to find the experts efficiently and effectively. C. Huang proposes a framework in [16] to find such users who are experts in their network. With the help of recent work and studies on distributed representation of words, we can now summarize chunks of text from the perspective of semantics and understanding the domains of knowledge by creating clusters of word vectors which are already trained. C. Huang used clustering method in [16] which is based on graph in order to extract knowledge of domain and perceive the shared factors of latent using factorization of matrices. The word vectors are clustered on the base of similarity of semantics using CLR graph based clustering. The TF-IDF is used for extracting term frequencies by considering the questions as documents. CLR does not require any processing of clustering indicators after the process and for expert ranking, a combination of matrix factorization and similarity of semantics for historical answers is used.

In most question answering systems, the past activities of a user are usually a few and thus, the model may not be very good for practice. Z. Zhao considered in [17], the issue of finding experts with the estimation of missing value. The social network of users are then employed to infer user model, hence improving the performance of the system to find experts in a CQA environment. An algorithm of Graph regularization matrix completion to create user models. Two more iterative procedures were developed in [17] GRMC-AGM and GRMC-EGM for solving the problem of optimization. GRMC-EGM makes the use of EGM (Extended Gradient Method), and Accelerated proximal Gradient Search (AG Method) is used in GRMC-AGM for its optimization.

The online communities consisting of millions of users have a large repository of knowledge and documents. But the user generated content is not always of good quality and developing a management system to facilitate knowledge sharing and seeking in online communities is a challenging task. G. A. Wang et al. suggests in [18] an algorithm named Expert-Rank which finds an expert after evaluation on the base of one's authority in the knowledge community and relevance to the documents. Using TF-IDF, WRR algorithm and PageRanking, three strategies for ranking of experts are explored i.e. multiplication scaling, cascade ranking and linear combination.

There may not be quality contents in online posts and discussions and may show user's biased opinion about a topic resulting in contradiction with a relevant answer. These discussions of low quality show that there exist users which are unprofessional. Therefore, it is essential for online forums to rank the users. M. Faisal et al. says in [19] that previously used expert-ranking techniques only consider content relevancy features and user's social network authority as parameters for evaluation of expertise. M. Faisal et al. proposed two ranking techniques in [19] using PageRank, ExpRank-COM, AQCS and Kendall's Tau. One technique based on reputation of user and their coexisting users in various discussion threads. The second one based on quality of answers by users. There is a large amount of questions which remain unanswered causing a setback for growing of online communities. D. P. Mandal et al. use QLL(Query Likelihood Language) model and Jelinek-Mercer's smoothing in [20] for finding the experts. The question similarity is determined from the question title and expertise level of a user is calculated on comparing it with archives.

D. R. Liu propose in [21] an expert finding method using authority of a category, user reputation and subject relevance. HITS, Page-Rank, TF-IDF and cosine similarity techniques are used. The subject relevance of a user denotes the pertinence of domain knowledge of a user and a particular question. The reputation of a user is obtained from the previous record for that user and link analysis gives the user authority.

T.P. Sahu proposed a mixture model in [22] which is parameter free for identification of topical authoritative users. The BMM framework is used in statistical framework which is created using activity information of users on a CQA site. The expectation maximization (EM), Newton-Raphson and Fuzzy C-means are used. The function for density of probability is devised and BMM which corresponds the most to a particular user is selected. The approach is tested on AskUbuntu and StackOverflow as well.

Keeping in view the growth of personal expertise of a user over time can improve the quality of result in finding an expert. For some applications it is advantageous to find the user which might become a potential expert in some time in future. This may also improve an overall engagement, participation and performance of users in community and improve their skills as well. For this a learning framework is proposed by M. Neshati in [23] using the evidences of expertise in current time for ranking of experts on StackOverflow. Various features like topic transition, user-behaviour, emerging topics and topic similarity are analyzed, The probability of a user to be an expert in future is calculated using Bayes theorem. Other techniques like SVM algorithm, JM-smoothing, LDA, Markov assumption and point-wise learning are also used. Results of standard methods for identification of experts are often biased towards more active users than the ones with more knowledge. J. Yang in [24] introduces a metric for better classification of experts by considering their work quality. The two classes identified are named as owls and

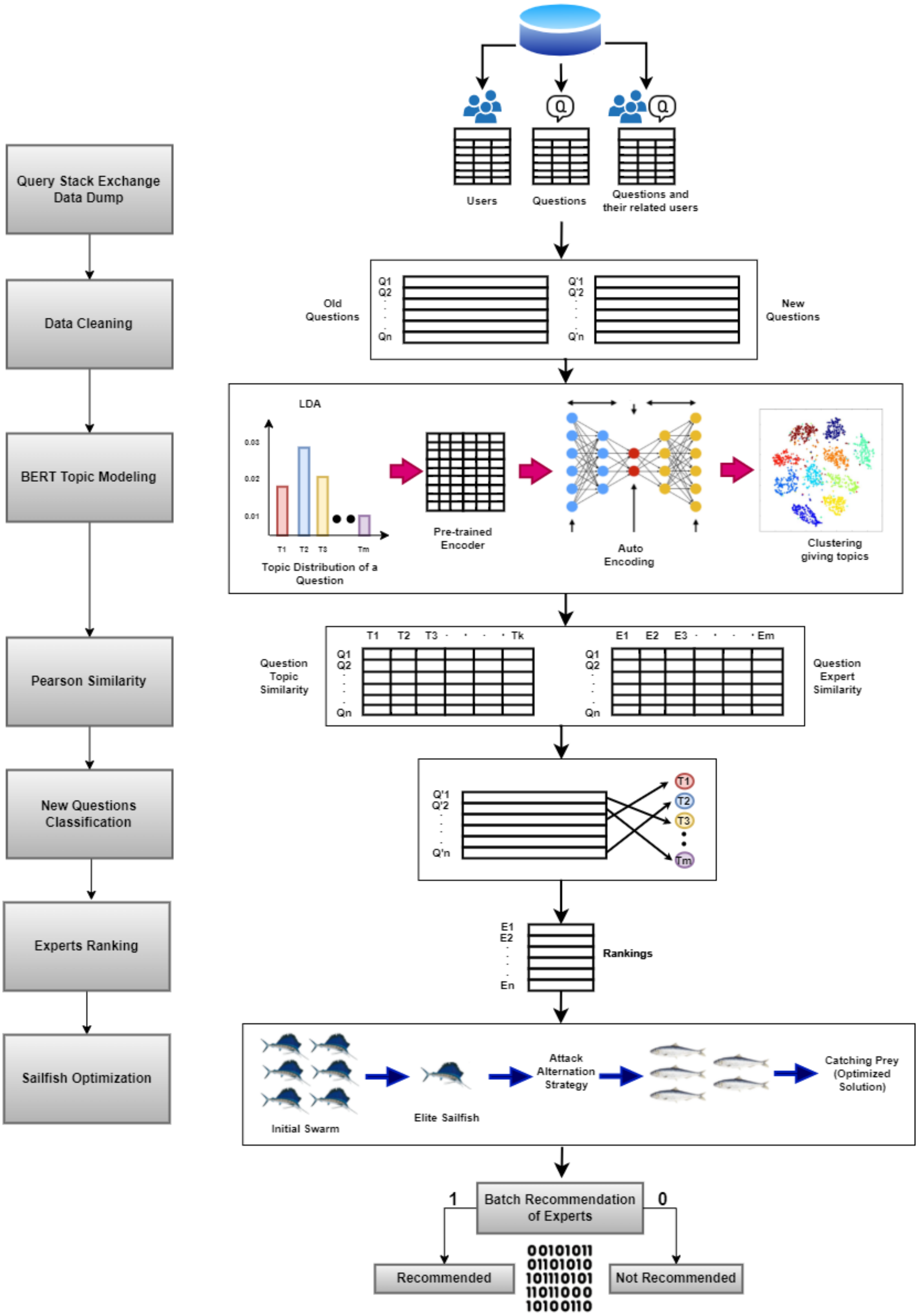


FIGURE 2.1: Framework of recommendation methodology

sparrows and behavioral characteristics are obtained using MEC (Mean Expertise Contribution), Spearman correlation and Z-score.

In CQA sites, the questioner most of the times has to wait for a significant amount of time for users to answer a question and it is not confirming that the quality of answer will be up to mark. S. Wang proposes TPLM (Topic Professional Level Model) in [25] for finding suitable experts for questions. The model uses PageRank, LDA and InDegree Algorithm for combining professional level model and topic model from perspectives of link structure and textual content.

D. Kundu proposes a method in [26] for detection of active experts for a new query for improving the effectiveness of routing process for any question. Query Likelihood method and Jelinek-Mercer's smoothing is used for identification of experts.

2.0.1 Background

Community Response Websites (CQA) provide a convenient place for users where they can publish answers and ask questions on a variety of topics. CQAs such as Stack Abundance, Quora and Yahoo Answer, are still an important archive and have attracted a lot of attention. For example, in 2019, Stack Abundance added 2.6 million new queries, 2.8 million responses and more than 1.7 million new users to join the community. Considering the vast amount of information stored and the entry of visitors, provide relevant content to promote interaction and satisfactory user search in the community is essential to the CQA website. In doing so, such forums should offer commendation services to users or experts on a particular topic so that the question can get the best answer. Although such CQA sites benefit users, there are still various disadvantages to current CQA programs.

2.0.2 Methodology

The used approach has four major parts. First the data is queried and refined. Subsequently clustering, modeling and topic creation is done. Afterwards, similarities between new and old questions, questions and topics and questions and experts are calculated. Then, experts belonging to each category are ranked on the base of their past performance, recency of activities, reputation and activeness. Eventually, the recommendation is optimized by variating the three parameters, i.e. answer-ability, minimum resource utilization and question coverage.

2.0.3 BERT

The BERTopic library is used in this study which models the topics using state of the art Transformer. The transformer encodes the data using word embedding. BERTopic is unsupervised, bidirectional and a pre-trained model that has originally been trained on the whole Brown Corpus and English Wikipedia due to

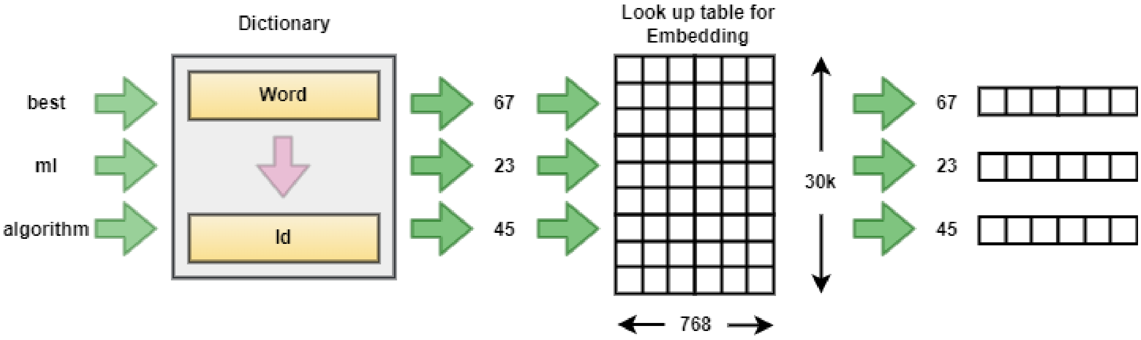


Figure 2.2: Look Up Table for finding word embedding in BERT

which it has already created lookup table for word embedding and a word to id Dictionary making it efficient in finding embedding for each word in dataset being used. Fine tuning is done on downstream NLP tasks e.g. question and answer pairs which makes BERT most suitable to be used for StackOverflow website. Sentence-transformers package is used for this study to extract word-embedding. These word embedding are used to divide questions from datasets into multiple clusters and generate topic frequencies. BERTTopic combines BERT embedding and CTFIDF which is a class based TFIDF for creating dense clusters and generating frequencies. For Topics T$\{t1, t2, \ldots, t_C\}$ where C is total number of topics/-clusters and total number of words $N_t\{n_{t1}, n_{t2}, \ldots, n_{tC}\}$ in every topic T. TFIDF is calculated as following:

$$TF = f_{w(x,y),t_c} = \frac{count(w_{x,y}, t_c))}{n_{t_c}} \tag{2.1}$$

$$IDF = \log \frac{C}{\sum_{r=1} f_{w(x,y),t_r}} \tag{2.2}$$

$$IDF = \log \frac{C}{\sum_{r=1} \frac{count(w_{x,y},t_r)}{n_{t_r}}} \tag{2.3}$$

$$TFIDF = TF * TDF \tag{2.4}$$

$$TFIDF = \log \frac{C}{\sum_{r=1} f_{w(x,y),t_r}} * \log \frac{C}{\sum_{r=1} \frac{count(w_{x,y},t_r)}{n_{t_r}}} \tag{2.5}$$

2.0.4 Pearson correlation

Pearson's correlation $Sim(Q^a, Q^b)$ shows the linear relationship strength between two documents belonging to $Q^a\{q_1, q_2, q_3, \ldots .q_N\}$ and $Q^b\{q_1, q_2, q_3, \ldots .q_N\}$. The similarity among clusters of question is determined by Pearson Correlation [27] as shown in Eq. 2.6-2.7.

$$Sim(Q^a, Q^b) = \frac{\sum_{j=1}(Q^a_j - mean(Q^a_j))}{\sum_{j=1}(Q^b_j - mean(Q^b_j))} * \frac{\sqrt{(\sum_{j=1}(Q^a_j - mean(Q^a_j)))^2}}{\sqrt{(\sum_{j=1}(Q^b_j - mean(Q^b_j)))^2}} \tag{2.6}$$

$$Sim(Q^a, Q^b) = \frac{\sum_{j=1} Q^a_j Q^b_j - \frac{\sum_{j=1} Q^a_j Q^b_j}{N}}{\sqrt{\sum_{j=1} Q^a_j - \frac{(\sum_{j=1} Q^a_j)^2}{N}}} * \frac{1}{\sqrt{\sum_{j=1} Q^b_j - \frac{(\sum_{j=1} Q^b_j)^2}{N}}} \tag{2.7}$$

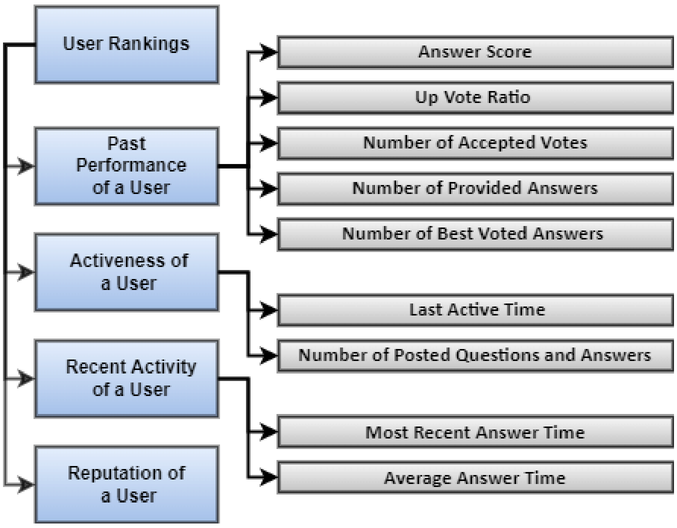


FIGURE 2.3: Expert Ranking Parameters

2.0.5 Ranking of Experts

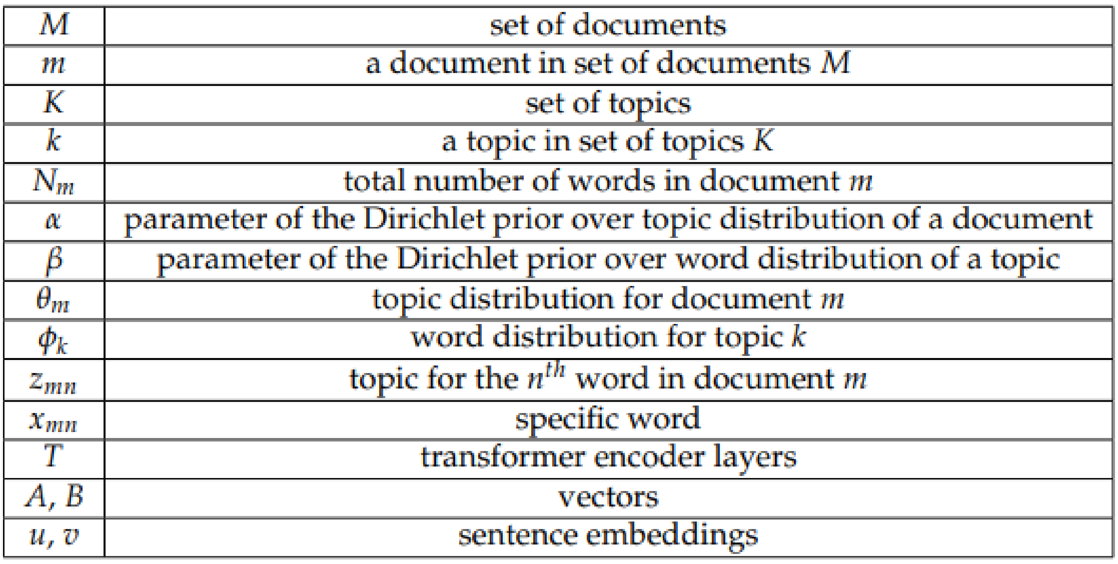
Reputation points are used on Stack Overflow (SO) for recognition of users. The reputation points have often proven to be a great base for users for building a profile for their career and show their expertise in different domains. Usually reputation of users is used as a baseline for the estimation of their expertise and experience. However, there are several other ways for a user to increase their reputation points for which expertise is not necessary, such as by asking good-quality questions. Therefore, high-reputation point is not a very good parameter for indicating a user experience and expertise on Stack Overflow [28]. In order to achieve better results, a new methodology for ranking of experts is proposed.

Chapter 3

BERT for Topic Modeling

3.1 Topic Modeling

The technique of topic modeling refers to some algorithms that finds representations for hidden underlying latent semantics in a data-set [29],[1]. Topic modeling analyzes text documents or simple text for discovering hidden or underlying topics or themes which are connected to one another [30]. In topic modeling, a topic means a word with highest probability among all the words in a cluster. These methods were developed initially for text-mining but currently are being used for images, genetic data for structure discovery, computer vision [31] and bio-informatics [32].



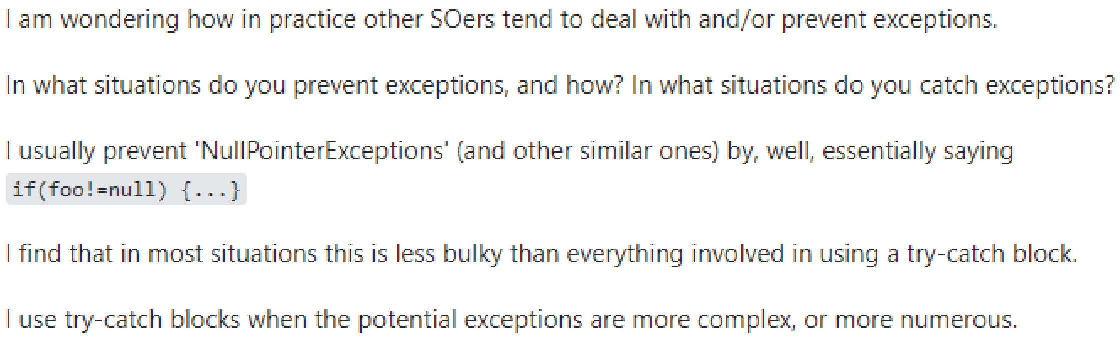
M	set of documents
m	a document in set of documents M
K	set of topics
k	a topic in set of topics K
N_m	total number of words in document m
α	parameter of the Dirichlet prior over topic distribution of a document
β	parameter of the Dirichlet prior over word distribution of a topic
θ_m	topic distribution for document m
ϕ_k	word distribution for topic k
z_{mn}	topic for the n^{th} word in document m
x_{mn}	specific word
T	transformer encoder layers
A, B	vectors
u, v	sentence embeddings

Table 3.1: Used notations and their meaning

Approaches for topic modeling are considered to be a form of machine learning unsupervised techniques as the mixture parameters and topics are unknown and are extracted from the data. We can say it is not trained on already labeled or tagged data. LDA is the most commonly used topic modeling probabilistic technique [33] and another technique is pLSA which is also very foundational technique [34]. Both of the models have been modified and extendedly used for new models and are used for topic modeling very frequently. Both pLSA and LDA believe that a topic consists of word collection and every document of text collection consists of topic mixture. To explain topic modeling concept, take an example of Figure Fig. 3.1 and Fig. 3.2 shows texts from two different questions from StackOverflow. Table 3.2 shows the topics which were extracted from the questions using technique of LDA. Theses topics explain an abstract or high level explanation or summary for these questions.

3.1.1 Latent Dirichlet Allocation (LDA)

Blei et al. [33] introduced LDA as a probabilistic genetic model for text data-set. It consists of three layers of Bayesian hierarchical model where each document or questions x in a given collection of questions I_{old} gets modeled as a mixture (finite) on a set of topics y in the set of questions. And each topic y (which is a collection of words) is modeled as a mixture (infinite) of topic probabilities set. A highest probability word is considered to be a topic or a label (words representation in a topic) which can be used for identification of a topic. e.g. the question in Fig. 3.1 and Fig. 3.2 can be considered to be placed in category "exceptions", "Java" or "error-handling".



I am wondering how in practice other SOers tend to deal with and/or prevent exceptions.

In what situations do you prevent exceptions, and how? In what situations do you catch exceptions?

I usually prevent 'NullPointerExceptions' (and other similar ones) by, well, essentially saying `if(foo!=null) {...}`

I find that in most situations this is less bulky than everything involved in using a try-catch block.

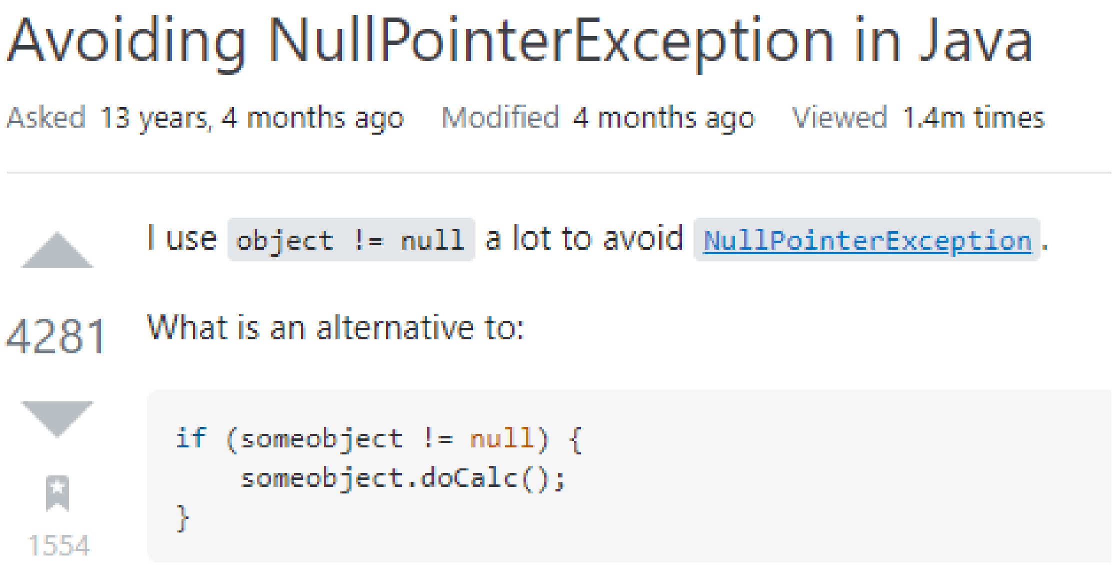
I use try-catch blocks when the potential exceptions are more complex, or more numerous.

Figure 3.1: StackOverflow example question 1

Figure 3.3 shows a diagram for LDA model giving a visual representation and explanation for process of LDA. Random variables are represented by nodes, probabilistic dependencies are represented by lines and repetitions are represented by rectangles. The highlighted variable is considered to be a variable under observation and others are considered as latent variables. Table 3.1 represent the notations

Table 3.2: List of generated topics

1	nbsp studio background web css visual documentation ui asp implementation
2	xslt xml xsd xsl openxml xmls xmldocument xhtml lxml xmlelement
3	datetime timezone utc timestamp date datepart timezones startdate daylightsavingtime dtscandate
4	exception exceptions registry inetsrv dylib boolean safehandle illegal biztalk cryptography
5	entity entities fk parentid pk ef framework poco id db
6	python py packages module matplotlib modules ipython pylons django pylint
7	jquery element dom javascript selector div sortable html js jqueryui
8	grid gridview datagrid jqgrid datagridview subgrid datatable grids datarow databinding
9	wcf service services webservice proxyzipeeeservice xmlserializer threadid servicemodel soap wsdl
10	regex regular regexp rege stringbuilder backslash curly replace reg string
11	url domain urls htaccess subdomain uri site seo website domains
12	iphone xcode ipad ios apple sensor icloud xfe itunes nsnumber
13	memory gb allocation ram mb allocations bytes cpu bandwidth storage
14	handler handlers eventhandler clientmodify filesystemwatcher serverevents listener viewer submitformcountdown onprogresshandler
15	ajax jquery js callback bookmarklet submission blogengine async jwebunit webapp
16	django admin py queryset nginx djangoproject djangolean pydev virtualenv hostname
17	dll assembly dlls assemblies studio referer clr ibobjects dl closurecompiler
18	android emulator appid apps androidmanifest activities textview blackberry samsung mobile
19	jvm java jdk netbe bytecode jython jhat simplelogger gwt logger
20	uitableview uiview uiscrollview tableview uiviewcontroller subviews subview uiimageview uiimageviews nsscrollview
21	git repo repository github repositories rebase gist changelog revision gitlab
22	validation validator validate xmlvalidationschemafactory validators mvc dataannotations valid validations jasperservice
23	php userproject phpcs aptana datarecovery allowread mapdb configurationdata closure lighttpd
24	password passwords username login authentication passwordchar encryption secretkey passwordbox wmsvc
25	ruby rspec irb aes trollop rvm gemfile eax gem rails
26	compiler gcc compilers preprocessor compilation cgi fcgi conversion mfcgi llvm
27	session permission permissions users login username getuser guestuser registereduser admin
28	silverlight xaml adsense businessapplication ioa oob childwindow wpf mainwindow applications
29	json jsonresult jsonp jsonb getjson tojson extjs jsobjectdata timeout jsonobject
30	svn svg subversion tortoisesvn folders folder visualsvn svnsync sv tortoisehg



Avoiding NullPointerException in Java

Asked 13 years, 4 months ago Modified 4 months ago Viewed 1.4m times

4281

1554

I use `object != null` a lot to avoid `NullPointerException`.

What is an alternative to:

```
if (someobject != null) {
    someobject.doCalc();
}
```

FIGURE 3.2: StackOverflow example question 2

being used in LDA diagram. For the questions corpus with I questions, the genera-

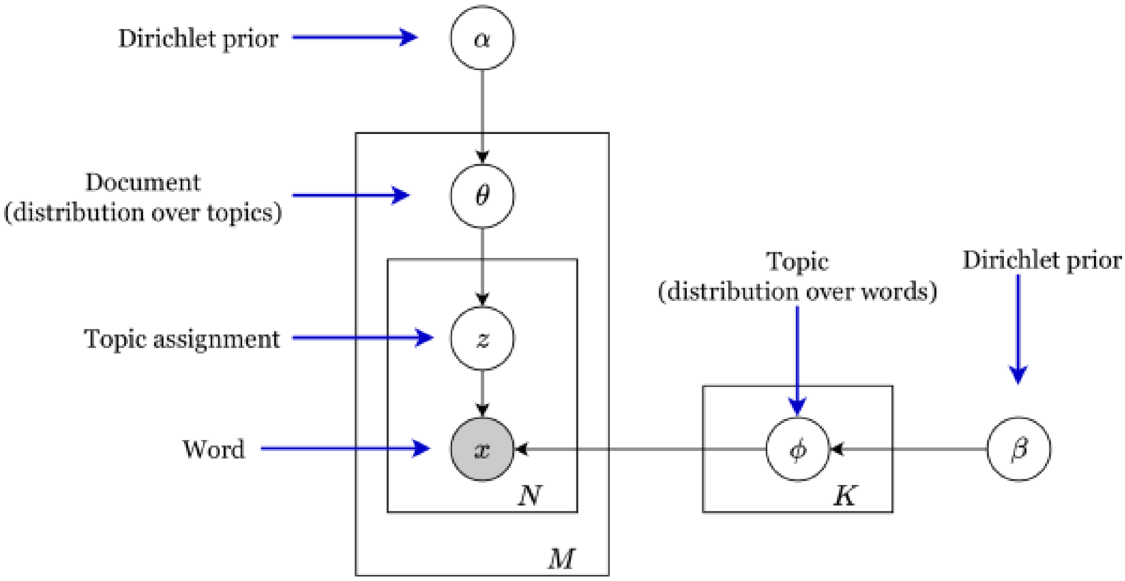


FIGURE 3.3: Plate diagram for LDA [1]

tive step-by-step process of LDA for y topics is shown below [1]: The degenerative LDA process initiates with β which is a Dirchilet parameter on the distributions of words of theme y in set of topics Y. Each topic y can be considered to be a multi-nomial distribution ϕ_y over all the words. α is a Dirchilet prior parameter over the distribution of topic of questions x in the set of questions X. Each question x can be defined as a distribution of multi-nomiality θ_m over the themes. For every word n in question x having maximum words N_m, the assignment of topic

```
1. For each topic $k \in \{1, ..., K\}$
       $\phi_k \sim \text{Dirichlet}(\beta)$                [draw distribution over words]
2. For each document $m \in \{1, ..., M\}$
       $\theta_m \sim \text{Dirichlet}(\alpha)$             [draw distribution over topics]
       For each word $n \in \{1, ..., N_m\}$
         $z_{mn} \sim \text{Multinomial}(1, \theta_m)$      [draw topic assignment]
         $x_{mn} \sim \phi_{z_{mn}}$                        [draw word]
```

for a word n is $z_{m,n}$. In the process, the model practice understanding of the procedure of question creation from vocabulary of words. The practice of document understanding can be then used for the process of reverse engineering for inferring the topics. In LDA, the only observed variables are the words.

3.2 Word Embedding

Word embedding are a vectorial representation of words in form of numbers which either represent the semantic or syntactic meaning of a word or combination of words or context of a word. Word embedding represent a words as a real valued vectors of fixed length. These association of words are learned by making use of various techniques like auto-encoders, LSTM bi-directional, neural networks etc. Once the learning process is done, these pretrained embedding can be used on various dets of data. For transfer learning, these pre-trained embedding of words can be utilized [1].

These embedding of words can be used in mathematical functions like Euclidean distance or Cosine similarity for finding similar words.

The measure of cosine similarity between two vectors (non-zero) 'A' and 'B'. The $cos(\theta)$ can be derived as following [1]:

where B_i and A_i are B and A components respectively [1]. Between two points,

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

$$\text{Cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.1}$$

the length of segment of line is called Euclidean distance and is represented as follows. Various number of model exists for the construction of embedding for words [1], e.g. BERT presented by Devlin et. al [3], ELMO presented by Peters

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} \tag{3.2}$$

et al. [35], Clove presented by Penningtom et al. [36] and Word2Vec presented by Miklov et al. [37] etc.

3.2.1 TFIDF

The TFIDF (Term Frequency - Inverse Document Frequency) shows a measure of the statistical importance or relevance of a word to a question or document in a collection of document or questions. In the calculations of TF-IDF, the frequency of number occurring in a document is directly proportional to the importance or significance of a word. The significance of a word is indirectly proportional to the frequency of a word occurring in the whole set of data or the corpus. According to the calculations of TF-IDF, the words which appear very frequently and are common in entire set of data like "is" and "the" seems to have less significance than the words like "experiment" etc if it appears frequently in a question of document and not mostly in other documents. The score for TF-IDF of a word x in question m for the set of questions M can be calculated as follows [38]. where the frequency

$$\mathrm{TF} - \mathrm{IDF}(x, m, M) = \mathrm{TF}(x, m) \cdot \mathrm{IDF}(x, M) \tag{3.3}$$

for term TF is as follows:and the inverse of frequency for a document is follows:

$$\mathrm{TF}(x, m) = \log(1 + \mathrm{freq}(x, m)) \tag{3.4}$$

TFIDF can be used for process of searching homogeneous documents having alike

$$\mathrm{IDF}(x, M) = \log(D/\mathrm{count}(m \in M : x \in m)) \tag{3.5}$$

significant words. This score can prove to be very helpful in word representation as count of words on the base of statistics can be easily provided as an input to other classification algorithms.

3.3 Language Model which are based on Transformers

A theoretical explanation of some topics is provided in this section for better understanding and explanation of BERT [3]. A short explanation of transformer architecture and models for language are explained in this section.

3.3.1 Language Models

Language Models based on probability are the distribution of probability over word sequences. Given a word sequence having j length and probability P(X) which is equal to $P(x_1, ..., x_j)$ on to the entire sequence. Chain rule can be used for its further decomposition. [39].

Models like N-gram often approximate the language models. In case of uni-gram it is considered that no word depends on another word prior or after it.

The MLE i.e. maximum likelihood estimation for a document or a question shows the probability for a word generation.

Questions are further ranked on the base of p(x—m) probability. The higher this probability, more greater is the relevance of two questions [40].

3.3.2 Transformers

On the base of concept of attention, transformer models has high speed as the use the concept of parellizations too. Vaswani et al. presented the architecture of transformer[41].

Transformers make use of basic decoding-encoding concept for machine translation traditional neural networking system [42]. A block of multi-head attention is also used in transformers. It is an effective and efficient way to handle large sentences in neural sequence to sequence translation machine models. It is done by focusing on selected parts of a sentence being input for translation [43]. I the mechanism of attention, a portion of sentence being input is considered under focus or attention. Which gives a benefit to encoder which now can easily encode data in a fixed length of a vector [44]. Figure 3.4 shows the architecture of a transforming encoder and Figure 3.5 shows details for the architecture of transformer decoder. These images are extracted from [41]. An encoder consists of T layers stack and the layers are identical. And the decoder also comprises of T layers stack. Every encoder has sub-layers (two), a feed forward fully connected neural network and self attention multi head layer. The self-attention multi head layer encodes a word from incoming input along with considering other words too which are coming from input. The input of neural network of feed forward type is the output coming from self attention multi head layer. The output coming from every sub-layer of encoder gets followed by normalization layer [42]. For example if the output is y vector, then the output coming from every sub-layer encoder will be *LayerNormalization(y + Sublayer(y))*. Every sub-layer also has connections around it which are utilized by a step of layer-normalization.

The decoder of a transformer also consists of the exact same number of layers T

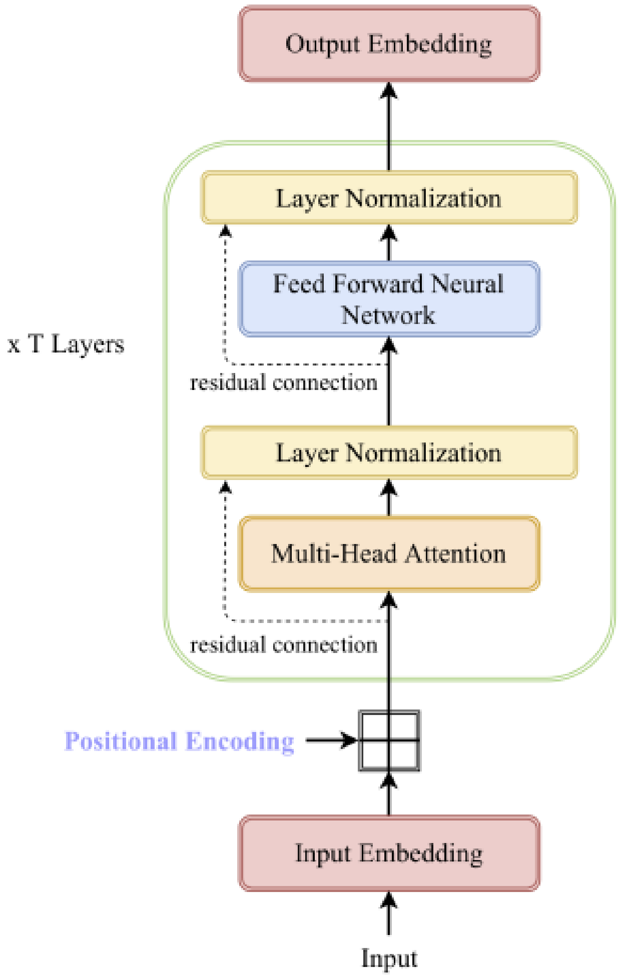


FIGURE 3.4: Architecture of encoder of a transformer

which are also identical as the encoder. Each layer of decoder has an third layer for multi headed attention in addition to the two sub-layers an encoder has. The third layer focuses on the different relevant portions of a sentence [42]. In case of the decoder, the output of decoder then gets passed softmax function and linear transformation for getting distribution of probability. It then gives the predicted probabilities.

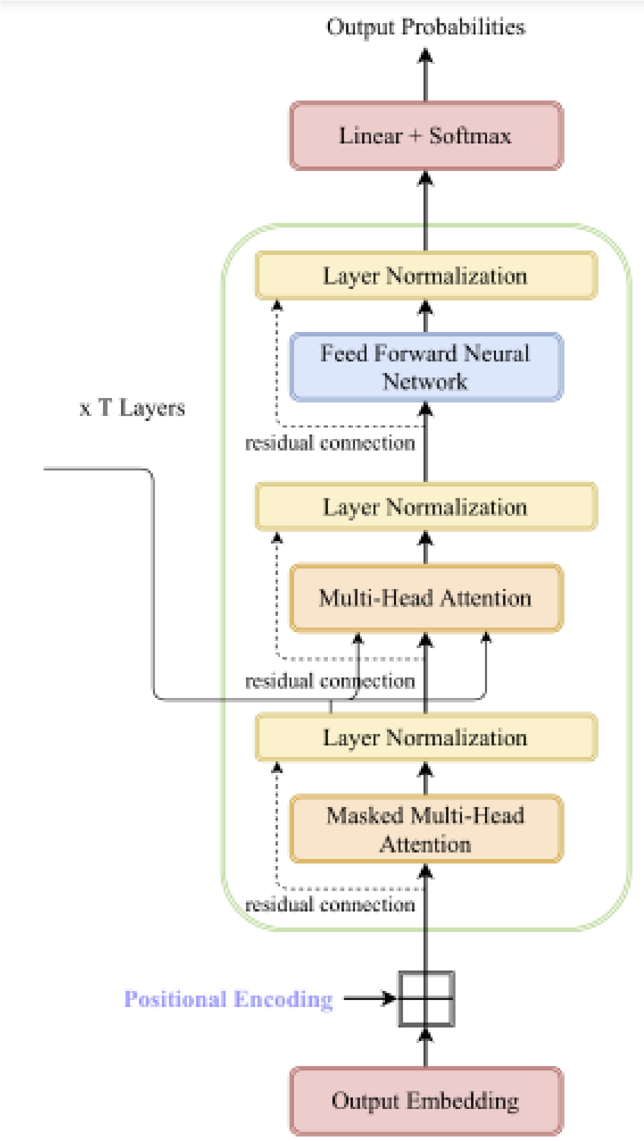


Figure 3.5: Architecture of decoder of a transformer

3.4 BERT (Bidirectional Encoder Representations for Transformers

A language model BERT [3] is based on an architecture model of a transformer. Conventional models for language are usually one directional but BERT is a two directional model. Since the architecture of BERT is composed of encoder layers which are bi-directional which means that the encoder can read an a sentence of a sequence of words from both of the directions at single instance of time. It provides embedding of word on the base of full context. Attention concept is also used by the transformers which helps it process lengthy sentences very efficiently and effectively. BERT is a model which is already trained i.e. pre-trained on

Wikipedia data set of English language and on BookCorpus (a data set which consists of 11038 books which are unpublished and which belong to 16 different genres). This model which is already trained can be used for many different purposes. Un-labeled data has been used for bi-directional training of BERT and hence it can be tuned finely for various tasks by simply just inserting an extra layer for output at the end [3]. BERT makes use of some tokens specially created

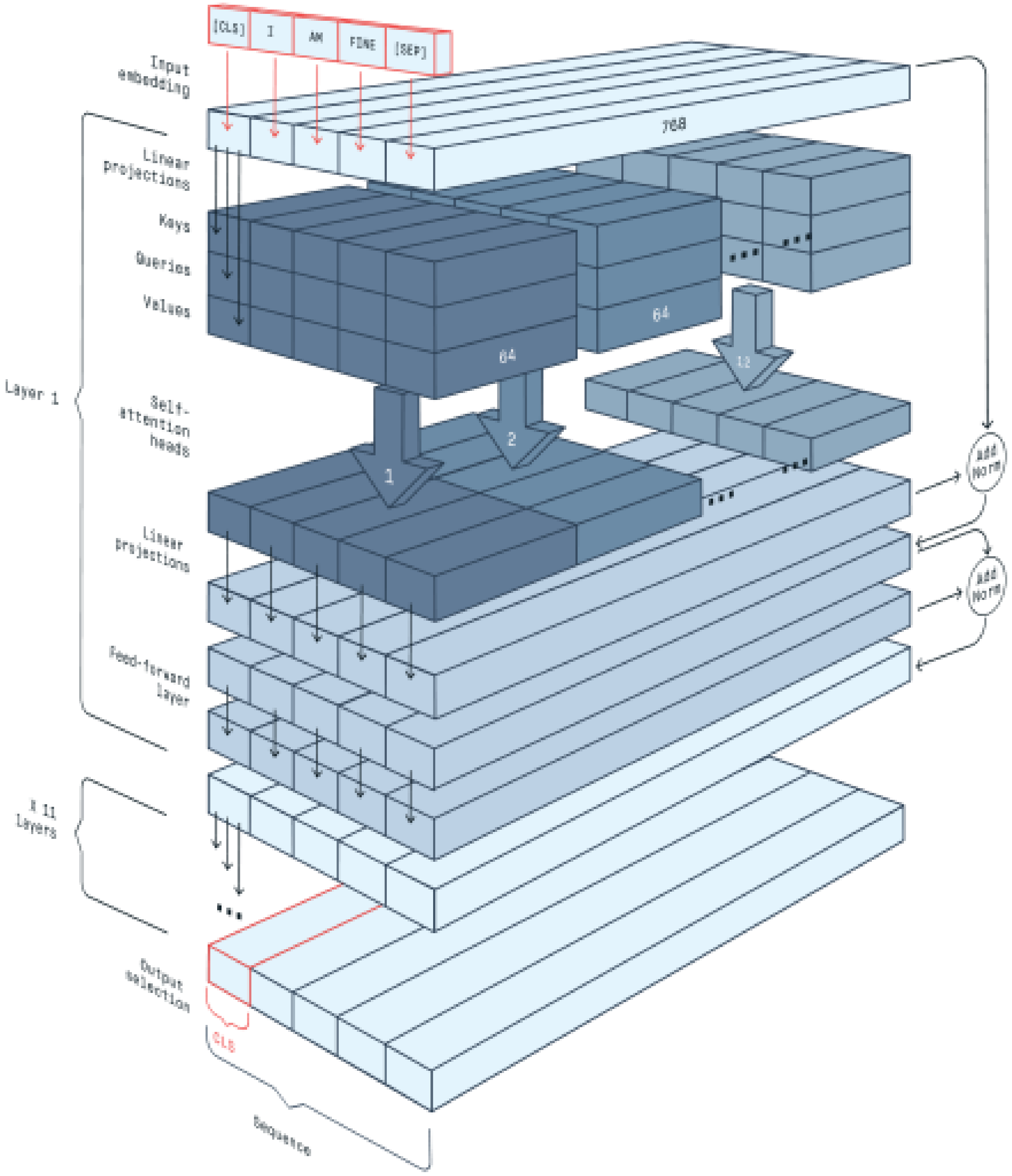


FIGURE 3.6: 3-D diagram for BERT [2]

for handling of variety of works and tasks. CLS is a token which is specifically created for classification which indicates the starting value of every sentence or a sequence. The embedding which are extracted from this token are further used as and aggregated sequential representation for the tasks like classification. SEP is a special token for separating a pair of two sentences. Figure 3.7 shows a visual impression of construction of input embedding for use of special tokens and aggregation of position embedding, segment embedding and token embedding. The embedding (segment) consists of multiple labels for various segments which are passed as an input for distinguishing segments.

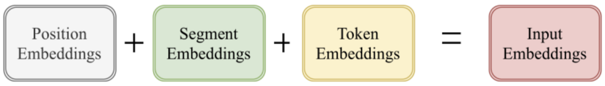


FIGURE 3.7: Input embedding for BERT [3]

3.4.1 BERT pre-trained on MLM

BERT is already trained on two tasks on NLP i.e. MLM (Masked Language Modeling) and NSP (Next Sentence Prediction). In the Masked modeling of language MLM, the model has been already trained for the task of predicting and guessing the missing word from a given sequence of words or a sentence. In the training of BERT, some of the words were the token of MASK and were treated as a missing word. This process was randomly done on 15% fifteen percent of the words for the prevention of model so that it can greatly focus on masked tokens or on the same position. In the training, 80% was the number of times that a word got replaced by the token of MASK, 10% was the number of times that a word got replaced by a number which is random and another 10% was the number of times when the words were remained as it is [45]. With the method of masked modeling for a language, the embedding usually can capture of detect the understanding of a connection or relationship among the words [3].

3.4.2 BERT pre-trained on NSP

BERT has also been trained on various tasks of NSP so that embedding can easily capture the essence of connection and relationships among different sentences. It is a classification task which can be considered binary. The data used is generated from a data set corpus by dividing then into pair of sentences. For 50% of the pair of sentences, the 2nd sentence was usually the next sentence and was labeled IsNext. For the remaining 50% of the pairs of sentences is some other randomly numbered sentence from the data set corpus which was labeled as NotNext.

3.5 Sentence BERT

The architecture of sentence BERT for finding the similarity between two sentences is demonstrated in Figure 3.8 [46] which uses cosine similarity index for finding the similarity of two sentences. It makes use of objective function of regression. For objective function of regression, the sentence embedding of v and u are compared by computation of cosine similarity index between both of them. The objective function used in this case is the mean square of the loss. These embedding of fixed sizes can also be calculated using Euclidean or a Manhattan distance.

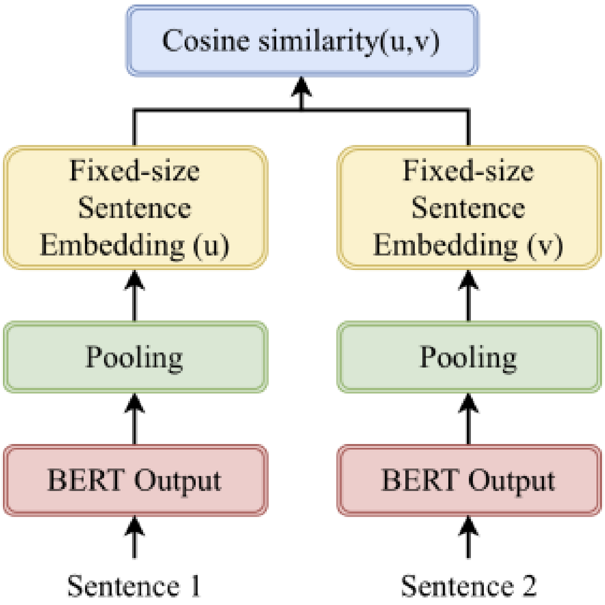


FIGURE 3.8: Sentence BERT architecture [3]

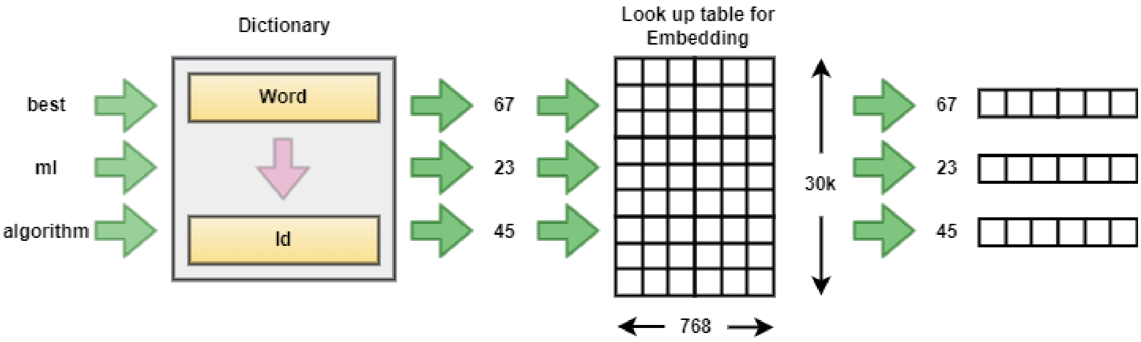


FIGURE 3.9: Look Up Table

The BERTopic library is used in this study which models the topics using state of the art Transformer. The transformer encodes the data using word embedding. BERTopic is unsupervised, bidirectional and a pre-trained model that has originally been trained on the whole Brown Corpus and English Wikipedia due to which it has already created lookup table for word embedding and a word to id Dictionary making it efficient in finding embedding for each word in data set being used. Fine tuning is done on downstream NLP tasks e.g. question and answer pairs which makes BERT most suitable to be used for StackOverflow website. Sentence-transformers package is used for this study to extract word-embedding. These word embedding are used to divide questions from data sets into multiple clusters and generate topic frequencies. BERTTopic combines BERT embedding and CTFIDF which is a class based TFIDF for creating dense clusters and generating frequencies. For Topics T$\{t1, t2, \ldots, t_C\}$ where C is total number of topics/clusters and total number of words $N_t\{n_{t1}, n_{t2}, \ldots, n_{tC}\}$ in every topic T. TFIDF is calculated as following:

$$TF = f_{w(x,y),t_c} = \frac{count(w_{x,y}, t_c))}{n_{t_c}} \tag{3.6}$$

$$IDF = \log \frac{C}{\sum_{r=1} f_{w(x,y),t_r}} \tag{3.7}$$

$$IDF = \log \frac{C}{\sum_{r=1} \frac{count(w_{x,y},t_r)}{n_{t_r}}} \tag{3.8}$$

$$TFIDF = TF * TDF \tag{3.9}$$

$$TFIDF = \log \frac{C}{\sum_{r=1} f_{w(x,y),t_r}} * \log \frac{C}{\sum_{r=1} \frac{count(w_{x,y},t_r)}{n_{t_r}}} \tag{3.10}$$

Chapter 4

Ranking of Experts

Ranking of expert shows a hierarchy depending on expertise level of a user. Reputation points are used on Stack Overflow (SO) for recognition of users. The reputation points have often proven to be a great base for users for building a profile for their career and show their expertise in different domains. Usually reputation of users is used as a baseline for the estimation of their expertise and experience. However, there are several other ways for a user to increase their reputation points for which expertise is not necessary, such as by asking good-quality questions. Therefore, high-reputation point is not a very good parameter for indicating a user experience and expertise on Stack Overflow [28]. In order to achieve better results, a new methodology for ranking of experts is proposed.
For every topic category d, experts e are ranked on the base of multiple factors values. These factors are majorly divided into four categories: reputation, activeness, recency and past performance. Various parameters like number of provided answers by an expert on a particular topic, recent activity time, number of best answers given by an expert etc. are integrated to calculate sub factor values. Eq. 4.1 shows the calculation formula used for ranking. The higher the value of ranking $O_{Rank,e}$, the higher is the expertise level of an expert.

$$O_{Rank,e} = O_{PastPerformance,e} * O_{Activeness,e} * O_{Recency,e} * O_{Reputation,e} \tag{4.1}$$

4.1 Past Performance

Past Performance factor of an expert is calculated by combining five sub-factors: answer score $O_{AnswerScore,e}$, number of provided answers $O_{NumberofProvidedAnswers,e}$, number of accepted votes $O_{AcceptedAnswers,e}$, number of best answers, up vote ratio

$O_{UpVotesRatio,e}$ of answers.

$$\begin{aligned}O_{PastPerformance,e} =& O_{AnswerScore,e} + O_{NumberofProvidedAnswers,e} + |O_{AcceptedAnswers,e}| \\ &+ O_{NumberofVotedBestAnswers,e} + O_{UpVotesRatio,e}\end{aligned} \tag{4.2}$$

Answer score $(AnswerScore, e)$ is calculated by difference of up votes and down votes values of an expert e in a particular category d.

$$O_{AnswerScore,e} = (V_{up})_{e,d} - (V_{down})_{e,d} \tag{4.3}$$

The number of provided answers is calculated by counting total number of answer posts by an expert e on topic d.

$$O_{NumberofProvidedAnswers,e} = count((P_a)_{e,d}) \tag{4.4}$$

The number of accepted votes or the number of best answers is represented by $(O_{AcceptedAnswers,e})$ by taking the number of votes of type accepted by an expert e on topic d.

$$O_{AcceptedAnswers,e} = (V_{accepted})_{e,d} \tag{4.5}$$

The number of best voted answers is calculated by using two values: the count of answer posts given by expert e on topic d, which are accepted by questioner and total number of answer posts given by an expert on topic d.

$$O_{NumberofVotedBestAnswers,e} = \frac{count((P_a)_{V_{accepted},e,d})}{count(P_{a,e,d})} \tag{4.6}$$

The up vote ratio $(O_{UpVotesRatio,e})$ is calculated by ratio of two factors: total up votes of an answer given by expert e and total votes given to that answer. $R_e =$ Set of all answers by expert e

$$O_{UpVotesRatio,e} = \sum_{i \in R_e} \frac{(V_{up})_i}{V_i} \tag{4.7}$$

4.2 Activeness

Activeness factor $(O_{Activeness,e})$ of an expert shows how active an expert is on a website. This factor is calculated by integrating two sub-factors: number of posted questions and answers $(O_{NumofPostedQA,e})$ by an expert on topic d and the last active time ($O_{LastActiveTime,e}$) of an expert e.

$$O_{Activeness,e} = O_{LastActiveTime,e} + O_{NumofPostedQA,e} + O_{MostRecentAnswerTime,e} \tag{4.8}$$

The number of posted questions and answers is calculated by using total number of answer posts given by expert e and total number of question posts asked by expert e.

$$O_{NumofPostedQA,e} = count((P_q)_e) + count((P_a)_e) \tag{4.9}$$

The last active time is calculated by taking difference of current date and Last Access Date (LAD) of an expert. The hours are extracted from the difference of date/time for calculation purpose.

$$O_{LastActiveTime,e} = hours(Time_{Current} - Time_{LAD}) \tag{4.10}$$

4.3 Recency

The recency ($O_{Recency,e}$) of an expert e shows worthy recent activities done by the user. The recency is calculated using the factor that how much time on an average, user took to answer a question and time of the most recent answer given by the user.

$$O_{Recency,e} = O_{MostRecentAnswerTime,e} + O_{AverageAnswerTime,e} \tag{4.11}$$

The average answer time ($O_{AverageAnswerTime,e}$) means average amount of time taken by a user e to answer a question.

$$O_{AverageAnswerTime,e} = \frac{\sum_{i \in S_{P_q,e}} (Time_i - (Time_i)_{P_a,e})}{N_{S_{P_q},e}} \tag{4.12}$$

The most recent answer time ($O_{MostRecentAnswerTime,e}$) is calculated by checking the most recent time of all the answers given by an expert.

$$O_{MostRecentAnswerTime,e} = max(Time_{P_a,e}) \tag{4.13}$$

4.4 Reputation

The reputation ($O_{Reputation,e}$) of an expert shows an overall expertise level of a user. It is given on the base of up-voted question, up-voted answer, accepted answer and approval of suggested edit. Usually CQAs provide a mechanism for voting making it feasible for user to pick best quality answers.[9]

$$O_{Reputation,e} = R_{e,d} \tag{4.14}$$

Table 4.1: Used notations and their meaning

Notation	Meaning
P	Post
P_q	Question Post
P_a	Answer Post
$V_{accepted}$	Accepted by Originator
V_{up}	Up-vote
V_{down}	Down-vote
$V_{up,e,d}$	Number of Up-votes given to an expert e in category d
$V_{down,e,d}$	Number of Down-votes given to an expert e in category d
$Time_{LAD}$	Last Access Date/Time
$Time_{Current}$	Current Date/Time
x	Question
g	Expert
y	Category
N_y	Total number of categories
I_{new}	Set of all new questions yet to be answered.
I_{old}	Set of new questions which are selected for experts to provide answer.
S_{xy}	Relevance of question x to category y.
O_{gy}	Ranking of expert g in category y.
S_g	Question for expert g
$\|I_{new}\|$	Total number of new questions.
$N_{experts}$	Total number of experts.
N_x^{max}	Maximum number of questions that can be chosen for answering by an expert x

Chapter 5

Sailfish Optimizer

SFO is a novel, naturally-inspired metaheuristic algorithm likened to a sailfish hunting group as shown in Figure 5.1. Demonstrates competitive performance compared to popular metaheuristic methods. Development can be applied in a variety of fields from engineering to economics or holiday planning to online. Metaheuristic algorithms can provide an effective strategy for solving development problems by using a mathematical model of social evolution. These algorithms use methods that find solutions close to the maximum value and acceptable cost. Since randomized methods play an important role in creating their structure, metaheuristic algorithms are known as subtle methods for solving complex functional problems [47].

In the last few decades, metaheuristic algorithms have been used in many applications. The main reason for the success of metaheuristic algorithms is that they use information that is often shared between multiple agents. And a few features can help these algorithms create higher quality results such as self-organization, evolution, and learning. All metaheuristics algorithms are ineffective and a few real-world problem-solving strategies can work very well. Number of agents used for people-based searches. This approach is an effective way to improve the testing phase. Indeed, the method of deceiving the population affects the performance of the algorithm. Different algorithms are used in different areas by many researchers [48]. Many algorithms and different applications could not provide a specific algorithm for solving all operating problems.

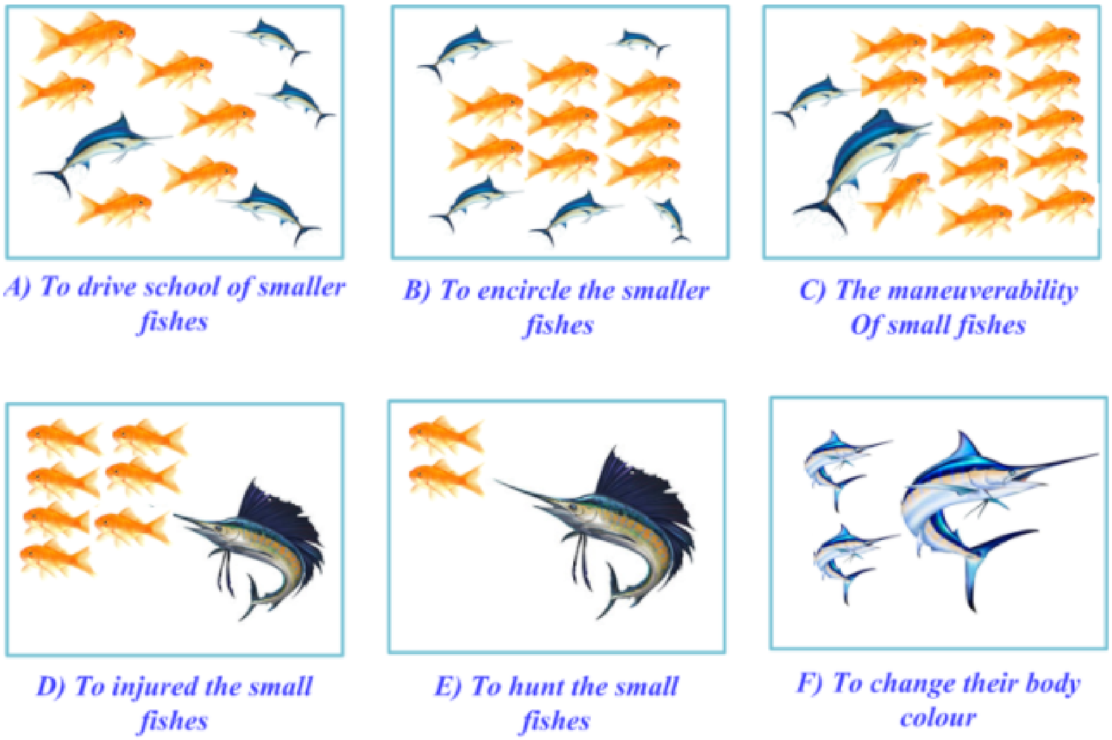
The SFO algorithm mimics the hunting of a group of sailfish alternating their attacks on school sardine levels. To the knowledge of current authors, there is no previous research on this topic in development literature. Current research has several significant differences with other recently published methods. First, the SFO was hired by two hunting groups and hunting animals to mimic the hunting team's strategy. Second, the proposed algorithm uses attack exchange to break down the combined defense of deer collection. Third, the movement of pets can be reviewed in the search area, and the hunter is allowed to catch a deer that is more suitable than before [49]. The main promotions of the SFO algorithm will be explained in this section. Then, the proposed algorithm and the mathematical model are discussed in detail. Details for the SFO is explained below [Shadravan et al 2019].

5.1 Sailfish Optimizer Inspiration

The sail is usually kept upright while swimming and is lifted only when the sailfish is attacking its prey. The raised sail has been shown to reduce head rotation on the sides, which may make the debt less noticeable to fish hunting. This tactic allows a sailfish to place its debts near fishing schools or even in them without being detected by a deer before striking it. Sailfish often attack one at a time, and their small teeth hurt predators in terms of scale and tissue removal. In general, about two hunting fish are injured during a sailfish attack, but only 24% of the attacks lead to capture [50]. As a result, injured fish increased in number over time at the invasive fish school. Considering that damaged fish are easy to catch, sailfish benefit from an attack on its conspecifics but up to a certain group size. A statistical model showed that sailfish in groups of up to 70 people should receive benefits in this way. The original method was called proto type [51] co-operation because it did not require any aggregation of aggression and could be a precursor to more complex forms of group hunting. The movement of the sailfish bill during a fish attack is usually left or right. Identification of each sailfish based on the shape of its back wings identified individual preferences for right or left flaps. The strength of this side effect was positively correlated with the success of the scan [5]. These side preferences are believed to be a type of behavior that enhances performance.

5.2 Initialization of SF Algorithm

SFO is a human-based metaheuristic algorithm. In this algorithm, sailfish is thought to be a human solution and the flexibility of the sailfish site in the search area. Therefore, the population in the solution area is generated randomly. Sailfish can search for one, two, three or more hyper dimensional locations with their variable vectors [9]. Sardine school is another important contributor to the SFO algorithm. It is thought that a group of sardines are also swimming in the search area. It is noteworthy that sailfish and sardine are compatible in finding solutions. In this algorithm, sailfish is a key element scattered throughout the search area



A) To drive school of smaller fishes

B) To encircle the smaller fishes

C) The maneuverability Of small fishes

D) To injured the small fishes

E) To hunt the small fishes

F) To change their body colour

FIGURE 5.1: Sailfish Algorithm [4]

and sardines can work together to find the best location in the area. In fact, sardines can be eaten by sailfish when it searches a search site and sailfish updates its location in the event that a better solution is found to date [52].

In the SFO algorithm, sailfish is thought to be a candidate solution and that the sailfish areas in the search area are represented flexibility of the problem. Location of i$^{\text{th}}$ sailfish in k$^{\text{th}}$ search frequency is defined by $SailF_{i,k}$, and its corresponding qualifications can be evaluated with $SailF_{i,k}$.

Sardines are one of the most important participants in the game SailFO algorithm. It is thought that sardine school enters search engine. The status of ith sardine is indicated by Si, and its the corresponding fit is measured in $f(S_i)$. In the SFO algorithm, i the most prominent sailfish is chosen as the elite sailfish, which is affected the movement and acceleration of sardines during an attack. The flowchart is shown in Figure 5.2

5.3 Eliteness of a Sailfish

Occasionally good solutions may be lost when reviewing positions of search agents and these positions may be weaker than older positions unless elitist selection is used. Elitism involves copying unchanging solutions for the next generation. In the SFO algorithm too, the best location for the sailfish is kept in each repetition and is considered elite. The elite sailfish is the strongest fish found to date and

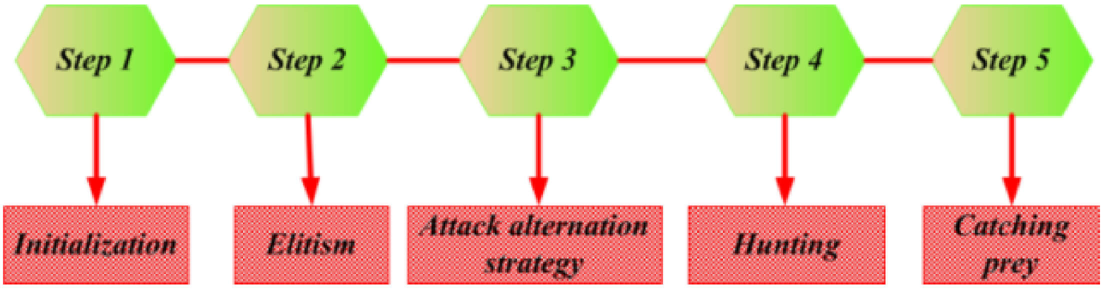


FIGURE 5.2: Sailfish Algorithm Flow Chart [5]

should be able to disrupt the movement and speed of sardines during an attack. In addition, as mentioned earlier, sardines will be damaged by the movement of sailfish's rostrum during group hunting [11]. Therefore, the location of the sardine injured in each repetition is also saved and this sardine will be selected as the best target for joint hunting by the sailfish. The position of elite sailfish and sardine victims with high repetitiveness is called consecutively.

5.4 Alternative attacks

The Sailfish often attack a hunting school when none of their people are attacking. In other words, sailfish can promote success in hunting by temporarily integrated aggression. Sailfish chase and herd their prey. The sailfish herd system adjusts its location according to the location of other hunters near the hunting school without direct contact between them. The SFO algorithm shows the sailfish attack strategy while hunting in groups. In doing so, the SFO algorithm helps any sailfish re-evaluate its location near a hunting school in two ways [12]. In the first case, the sailfish has a different attack to eat the school in relation to the elite sailfish and the injured sardine as seilfish 2, in the second method, the sailfish takes up empty space around the school hunting and mimics the circulation of the deer as a sailfish 1. In both ways. , sailfish will damage many sardines in the early stages of hunting and lead to a high level of successful catch in the latest stages of joint hunting [13].

Additionally, the position of the sardine is injured in each recurrence selected as the best place for joint hunting by the sailfish. This machine aims to prevent previously discarded solutions re-elected. wounded sardines and Elite sailfish are shown by $A^i_{injured_s}$ and $A^i_{elite_{SailF}}$, respectively, in repetition. In hunting, the strategy of sailfish's attack-alternation is often used for improvement of the success full hunting. The new position of sailfish $A^i_{new_{SailF}}$ is updated on the base of following:

$$A^i_{new_{SailF}} = A^i_{elite_{SailF}} -_i *(rand(0,1)x(\frac{A^i_{elite_{SailF}} - A^i_{injured_s}}{2}) - A^i_{current_{SailF}} \quad (5.1)$$

where $A^i_{current_{SailF}}$ is the sailfish's current position and ranNumb(0, 1) is a randomly chosen number which ranges between 0 and 1. The variable λ_i represents the coefficient in the ith iteration, and its value is found and derived as follows:

$$\lambda_i = 2 * ranNumb(0,1) * SarDin - SarDin \quad (5.2)$$

where SarDin is the density of a sardine , and it represents the number of sardines for every individual iteration. The variable 'SarDin' is found and derived as follows:

$$SarDin = 1 - (\frac{N_{SailF}}{N_{SailF} + N_{SarDin}}) \quad (5.3)$$

where N_{SarDin} and N_{SailF} shows the numbers of sardines and sailfishes, respectively.

5.5 Hunting

At the beginning of the group hunt, complete sardine slaughter is rarely seen. In 95% of cases, the sardine scales will be removed when the sailfish debts hit the sardines bodies. This causes a lot of sardine in schools to show damage to their bodies. The researchers found a positive correlation between the success rate of photography and the number of injuries in a hunting school [53]. At the beginning of the hunt, sailfish have extra strength to catch prey and sardines do not get tired and damaged. Sardines therefore maintain a high speed of escape and have great ability to navigate. Gradually, the sailfish's attack power will diminish over time. At the beginning of the hunting process, sailfish seems to be very energetic, and sardines seems not be very injured or tired. Sardines escape easily and rapidly. But, with continuous and uninterrupted hunting, the attack power of sailfishes will get decreased gradually. While, sardines will become exhausted and get tired, and their location awareness of the sailfishes will also get lower. As a result, the sardines gets chased and are hunted down. Based on this process of algorithm, the sardine's new position $A^i_{new_S}$ gets renovated and updated on the base of following:

$$A^i_{new_S} = rand(0,1)(A^i_{elite_{SailF}} - A^i_{old_S} + Power_{Attack}) \quad (5.4)$$

where $A^i_{old_S}$ is the former and old location of the sardine and rand(0,1)is a random number which lies in the range between 0 and 1.ATP represents the attack power of sailfish.

The $Power_{Attack}$ variable can be found and calculated as follows:

$$Power_{Attack} = B(1 - (2 * Itr * \epsilon)), \tag{5.5}$$

where ϵ and β and are coefficients that are utilized for reduction of the linear power for attack between B and 0 and 'Itr' is the iteration number. As the power of attack of the sailfish get lowered gradually when the time for hunting get passed, this decrease encourages and assist the search convergence. When the value for $Power_{Attack}$ is high, if taking for example, > 0.5, the location of each and every sardines gets updated. In contrast, only sardines that belong to α having β variables update their location. The number of sardines that update their location can be found as follows:

$$\alpha = N_{SailF} * Power_{Attack} \tag{5.6}$$

where N_{SarDin} is the sardines number for every iteration. The number of varying values of the sardines that things that updates their locations is found as follows:

$$\beta = d_i * Power_{Attack} \tag{5.7}$$

where d_i is the total number of variables in the i$^{\text{th}}$ iteration.

5.6 Hunt and catch of a prey

As a result of the intense and frequent attacks, hunting power stores will also be reduced and may have a decrease in the ability to see directional information about the sailfish, which affects the school escape route. Eventually, the sardines will be hit by a sailfish bill, separated from the sea and will catch up quickly [5]. In the final stage of the hunt, the wounded garden cut off from the tree will be caught immediately. In the proposed algorithm, it is thought that catching deer occurs when the sardine becomes better than its corresponding sailfish. In this case, the position of sailfish replaces the latest sardine hunters to increase the chances of hunting new prey.

When hunting a sardine, the fitness of the sardines should be better than the fitness of sailfishes. Under these circumstances and conditions, the location of sailfish A^i_{SarDin} is updated with the latest location of the sardine which got hunted A^i_SailF to enhance and promote the chasing and hunting down of sardines. The equation corresponding to it is as follows:

$$A^{i}_{SailF} = A^{i}_{SarDin} iff(SarDin_{i}) < f(SailF_{i}) \tag{5.8}$$

Chapter 6

Problem Formulation and Result Discussion

6.1 Problem Formulation

The constraints and objectives for optimization are: maximizing the answerability, minimizing the expert resource usage and maximizing the coverage. Making the use of limited resource consumption, the questions that are selected should cover the main essence of unanswered questions, avoiding the duplicate or repeated answers. For this [13] suggests using MOSFO-GA (Multi Objective Sailfish Optimization- Genetic Algorithm).

There are three constraints and objectives of multi-objective optimization. The questions which are selected must have a good probability of getting answered and should have good coverage. Since experts are an important resource, they must be used on a moderate degree. The main decision variable is whether or not an expert is recommended to question or not. An expert is recommended to a question for its answer if the decision variable is not zero. In case of “0” that particular expert is not recommended to answer the question. In case its zero for every expert, then no expert gets to be recommended for that questions to be answered. These constraints and objectives are formulated by [13] as following.

$$maximizingCoverage(I_{new}, I_{old}) = \frac{1}{|I_{new}|} * \sum_{s_a \in I_{new}} \max_{s_b \in I_{old}} (sim(s_a, s_b)) \tag{6.1}$$

$$maximizingAnswerability = \frac{1}{N_c * |I_{new}|} * \sum_{x=1}^{j} \sum_{y=1}^{N_y} \max(S_{x,y} * O_{g,y} * z_{x,s_g}) | g = 1, 2, ..., k) \tag{6.2}$$

$$minimizingExpert = \frac{1}{\sum_{x=1}^{N_{experts}} * N_x^{max}} * \sum_{x=1,g=1}^{x=j,g=k} z_{x,s_g} \tag{6.3}$$

subject to

$$\sum_{g=1}^{k} z_{x,s_g} \leq N_x^{max}, x = 1, 2, ..., j \tag{6.4}$$

$$z_{x,s_g} = 0, 1, x = 1, 2, ..., j; g = 1, 2, ..., k \tag{6.5}$$

	0	1	2	3	4	·	·	·	i
0	1	0	0	1	1	.	.	.	0
1	1	1	1	0	1	.	.	.	1
2	1	0	0	0	0	.	.	.	1
3	1	1	0	0	0	.	.	.	1
4	1	0	0	1	0	.	.	.	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
j	1	0	0	0	0	.	.	.	0

FIGURE 6.1: Expert recommendation matrix where '1' represents that an expert on y-axis is recommended to a question on x-axis

6.1.1 Optimization model using multi-objective binary sailfish algorithm

A binary optimal solution is used in optimization model. The sardines and sailfish gets represented by a matrix. The number or experts takes one dimension of matrix and number of new questions which are to be answered takes the other dimension. The value of every entry in the matrix is either one "1" or zero "0", "1" showing that the expert is recommended to answer that particular questions and "0" showing that the expert is not recommended to answer that query. [13] This decision mechanism is illustrated in the diagram. Simple SFO can only deal with continuous values and one objective. But MOSFO-GA [13] proposes use of multi objective optimizer. For expanding the space of search and avoiding to a local solution which is also an optimal solution, mutation and crossover functions in GA [54] is adapted in order to update the fish position.

6.1.2 Updating Position

Each element in matrix of fish has either 0 or 1 value, hence the updated matrix is also a binary matrix. The updating mechanism used is following [13]:

$$w_{j,k}(m+1) = \begin{cases} 1, & \text{if } sigmoid(w_{j,k}(m+1)) \geq Rand(n) \\ 0, & \text{otherwise} \end{cases} \tag{6.6}$$

where n$\in$ $\{1, 2, 3, ...,\infty\}$, $Rand(n)$ gives output between 0 and 1, $w_{j,k}(m+1)$ is the updated, derived and continuous value. The sigmoid function $sigmoid(.)$ is defined as follows [Binary grey wolf optimization approaches for feature selection. Neurocomputing, 172, 371–381.]

$$sigmoid(w) = \frac{1}{e^{-10(w-0.5)}+1} \tag{6.7}$$

6.1.3 Pareto-optimal non-dominated solutions

For performing multi-objective optimization an archive controller is used in order to store Pareto-optimal no-dominated solutions [55]. The archive controller updates the archive. Each derived solution is inserted into the archive when archive is not full. In case any new solution dominates some old ones in the archive, they will be removed from archive. Grid mechanism is used for re-arranging the objective space segmentation and removing the most crowded one of all the segments, in case the archive is full. Whereas, segment extension is done for covering the new solution when a solution is entered outside the premises of hypercube [13]. The injured sardine and elite sailfish are selected for guiding the update of positions. Since the search space used is multi-objective, hence due to Pareto-optimality, solutions are impossible to be compared directly. A mechanism for finding the injured sardines and elite sailfish proposed by [13] is used. Similarly, sardines which are injured are also selected. The Pareto-optimal solution Po is defined using roulette-wheel methodology [55] as following:

$$P_o = \frac{\lambda}{M_i}(\lambda > 1) \tag{6.8}$$

6.2 Evaluation and Results

The evaluation of the proposed approach is done in this section. The StackOverflow website data is queried from Stack Exchange Data Explorer. StackOverflow website has over 14 million registered users, more than 31 million answers and 21 million questions. For experimentation, 50,000 questions were used, out of which

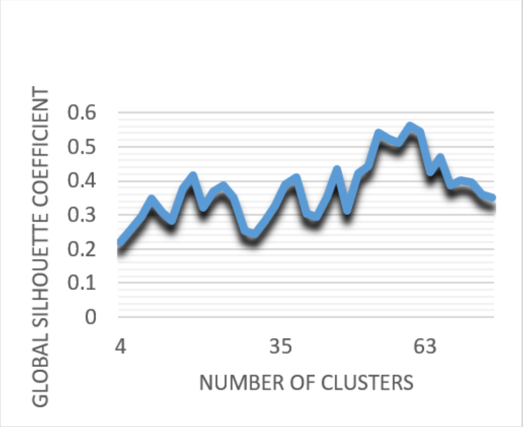


Figure 6.2: Global Silhoutte Coefficient

33300 questions were new and 16700 were already answered. Python language is used to implement all the code. The overall recommendation process is evaluated in two steps. The clustering methodology used is compared with other clustering algorithms like k-means++, SFO, SFO-GA using Silhouette coefficient. In step two, the credibility of recommendation process is evaluated. The comparison is done with the question priority and expert priority recommendation mechanisms in order to validate the single and batch recommendation mechanisms. Here optimization of batch recommendation process using BERT algorithm in Section 2.0.3 and combination of different expert finding parameters, is proposed. The MOSFO-GA [13] proposed by M. Li et. al. is compared with BSF for validating improvement of recommendation results.

6.3 Evaluation Criteria

6.3.1 Clustering

For evaluation of clustering performance for this dataset, the Global Silhouette coefficient is calculated for the algorithms to be compared. It is defined by Cagnina et al. [56] as follows

$$GSC = \frac{1}{N_c}\sum_{j=1}^{N_c} c * h_i = \frac{1}{N_c}\sum_{j=1}^{N_c}[\frac{\sum_{k=1}^{N_j} h(s_{j,k})}{N_j}] \tag{6.9}$$

1. Reputation

   The reputation of an expert shows an overall expertise level of a user. It is given on the base of up-voted question, up-voted answer, accepted answer and approval of suggested edit.

$$O_{Reputation,e} = R_{e,d} \tag{6.10}$$

2. Activeness Activeness factor of an expert shows how active an expert is on a website. This factor is calculated by integrating two sub-factors: number of posted questions and answers by an expert on topic d and the last active time of an expert e.

$$\begin{aligned} O_{Activeness,e} =& O_{LastActiveTime,e} \\ &+ O_{NumofPostedQA,e} \\ &+ O_{MostRecentAnswerTime,e} \end{aligned} \tag{6.11}$$

3. Past Performance Past Performance factor of an expert is calculated by combining five sub-factors: answer score, number of provided answers, number of accepted votes, number of best answers, up vote ratio of answers.

$$\begin{aligned} O_{PastPerformance,e} =& O_{AnswerScore,e} \\ &+ O_{NumberofProvidedAnswers,e} \\ &+ |O_{AcceptedAnswers,e}| \\ &+ O_{NumberofVotedBestAnswers,e} \\ &+ O_{UpVotesRatio,e} \end{aligned} \tag{6.12}$$

6.4 Parameter Setting

Clustering Performance The Kmeans++ clustering is used firstly and questions clustering is done using it. Along with the clustering results, it gives the optimal number of clusters to be used for other clustering algorithms as well as shown in Fig. 6.2. Secondly, the GS for SFO-GA is calculated using the kmeans++ as an initial solution and finally the global silhouette for BERT is calculated for the dataset being used. The clustering results are shown in Table 6.2 and it can be seen that GS coefficient value for BSF is better than kmeans++ and SFO-GA algorithm.

The clustering is done using BERT topic modeling. Various topics are created using the process of topic modeling. The inter-topical distance graph created is shown in Fig. 6.3 which puts the question with topics like datetime, timezone, utc, timestamp and date in same cluster Topic 1. Similarly Fig. 6.4 shows a topic cluster which can be represented by words like children, child, parent, family, childs

Table 6.1: Parameters Setting

Type	Parameter	Definition	Value
Random	mu_Rand	Mutation determination by random values from 0 to 1	_
	cr_Rand	Crossover determination by random values from 0 to 1	_
Select	A	Coefficient to decrease the power attack value linearly (from A to 0)	4
	pp	Rate between the number of sardines and sailfish	0.2
	E	Coefficient to decrease the power attack value linearly (from A to 0)	0.01
	cr_Rate	Probability of crossing a superior parent	0.5
	mu_Rate	Probability of mutation of a superior parent	0.5
	sf_size	Sailfish populations number	100

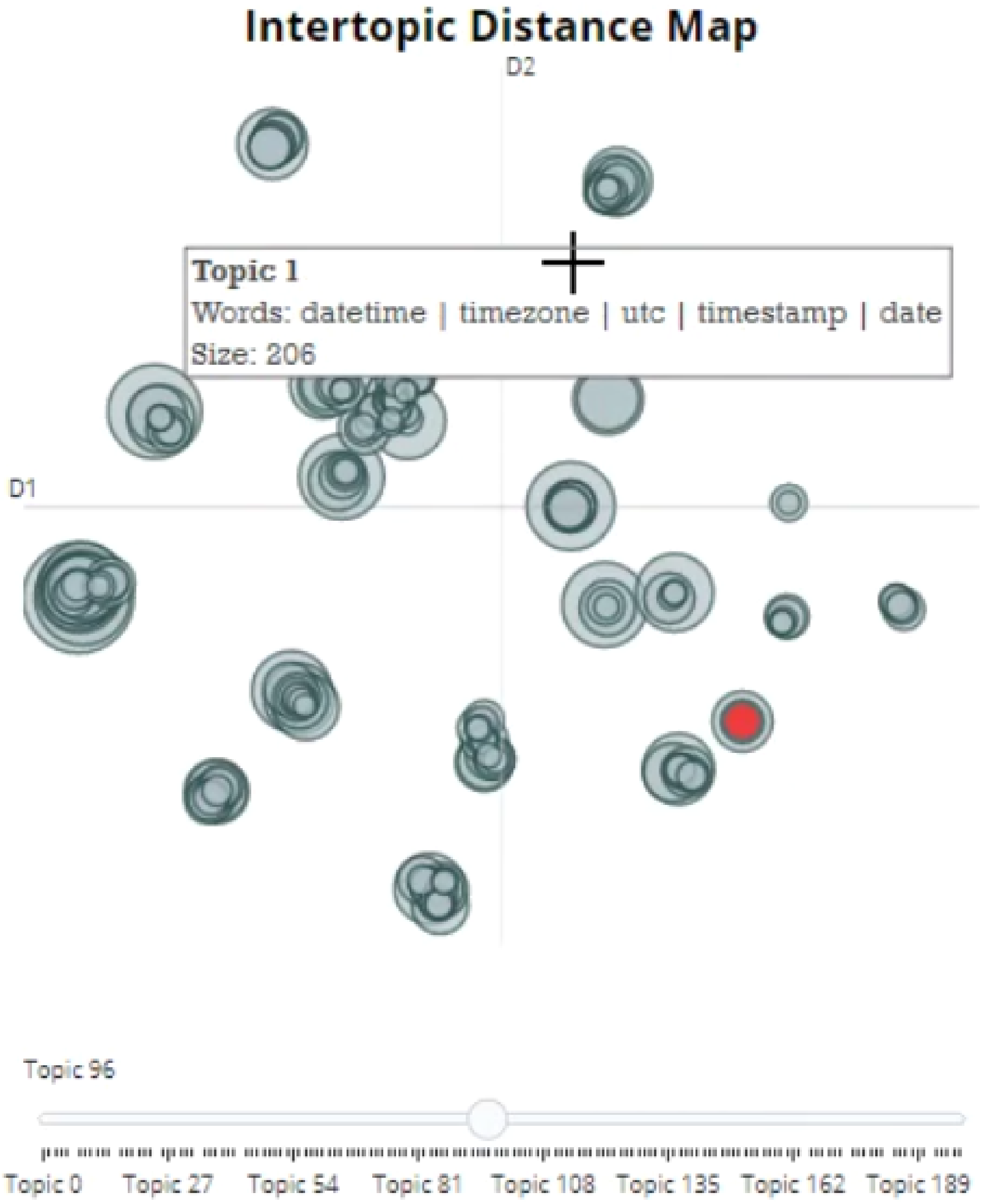
Table 6.2: Clustering result comparison

Method	Silhouette Coefficient Value
K-means++	0.56191
SFO-GA	0.64392
BESF	0.67058

etc.

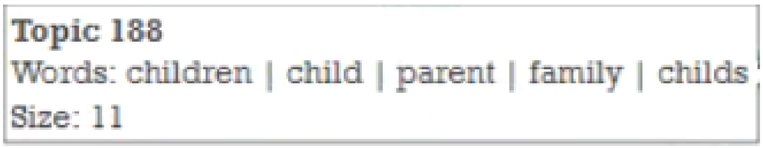
The recommendation methodology used by M. Li et al. [13] is compared with the currently used recommendation methodology BESF. The performance of both recommendation methods is given in Fig. 6.5-6.9. The x-axis shows the maximum number of questions that each expert can answer. Fig. 6.5 shows the coverage results. The coverage is a comparison of set of new questions selected by the BESF and MOSFO-GA and the original questions set. When the maximum number of questions that can be answered by the expert is increased, the coverage results also increase for both. But the coverage values obtained by MOSFO-GA is always less than those by BESF. Although both are not linear, but since BESF covers more number of topics obtained from new questions, hence is a better fit to fulfill requirements of the users posting a question.

When the value for maximum number of question that can be answered N_{qa} is limited to 5 then all recommendation methods considered here gives small values of coverage. MOSFO-GA gives results less than 0.12 but BESF gives almost 0.23. Similarly if $N_{qa} = 20$ then the coverage results are much better for both i.e. 0.35 and 0.53 respectively. This shows that on increase of N_{qa}, the coverage results also



Topic 1
Words: datetime | timezone | utc | timestamp | date
Size: 206

FIGURE 6.3: Inter Topical Distance Map



Topic 188
Words: children | child | parent | family | childs
Size: 11

FIGURE 6.4: Cluster Topic 188

get increased but not exactly linear as show in Fig. 6.5. Still the results for BESF are better than MOSFO-GA.

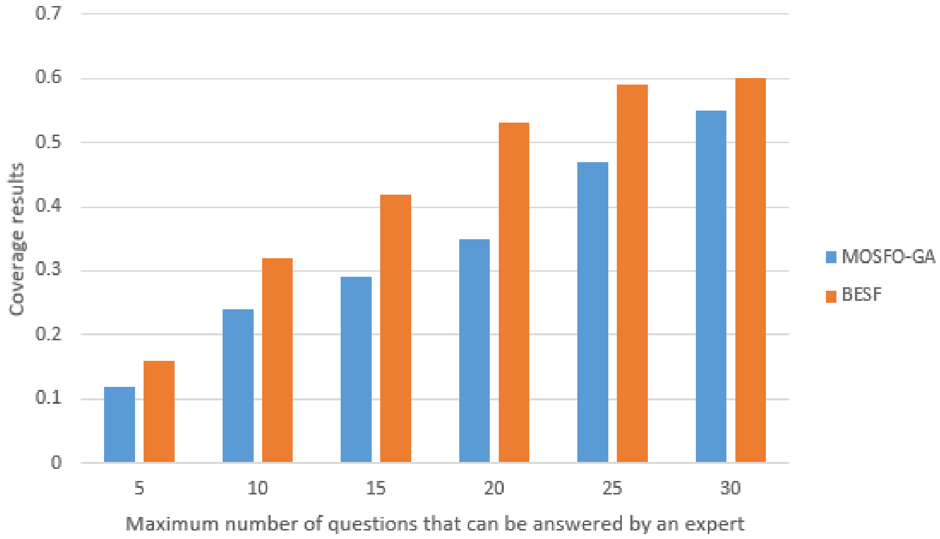


FIGURE 6.5: Coverage of recommendation techniques

Taking these recommendation methodologies into account, the coverage for BESF is approximately 0.43667 on average which is greater than all others as explained in Table 6.3. Considering the SI Algorithms, the coverage for BESF is approximately 0.42455 on average which is greater than all others as explained in Table 6.4. When $N_{qa} = 30$, the coverage for GWO its value is 0.403, WOA gives result value of 0.475, BMSFO gives 0.5025, MOSFO-GA has 0.59725 coverage value and BESF outlying all others give 0.62 as provided in Fig. 6.6.

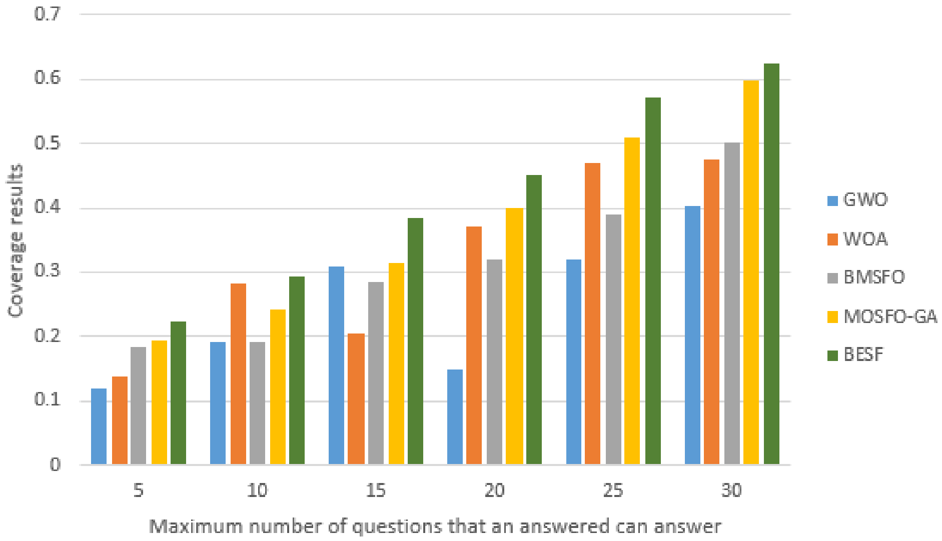


FIGURE 6.6: Coverage of Swarm Intelligence Algorithms

The results in 6.7 shows that the expert resource consumption in BESF is lower than MOSFO-GA recommendation technique. Limiting the maximum number of new questions that an expert answers to 30, less than 0.29 expert resources are required for BESF. When the number of answers that each expert is allowed to answer is less that is 5, then more number of experts are needed to answer the question. But still, the expert resource consumption at this stage is 0.36 for BESF, 0.44 for MOSFO-GA. On an average, BESF gives less values for resource consumption i.e. 0.3633 as shown in Table 6.4

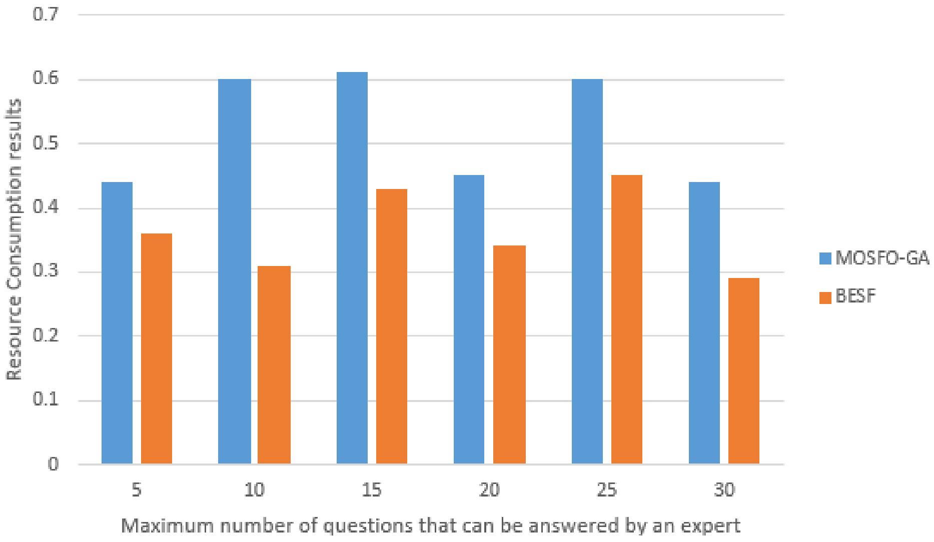


FIGURE 6.7: Expert Resource Consumption of recommendation techniques

On setting the value of N_{qa} to 10 considering the SI algorithms, the value of resource consumption for BESF and MOSFO-GA are approximately 0.605 and 0.6485 respectively. Whereas for GWO, WOA and BMSFO its values are 0.5705, 0.719 and 0.6901 respectively as shown in Fig. 6.8. On increasing the N_{qa} to 15, the value for resoruce consumption by BESF gets extremely low i.e. 0.55 which is very less than that of MOSFO-GA i.e. 0.691. The results for resource consumption shows that N_{qa} is not linear to resource consumption and it gives varying values for both MOSFO-GA and BESF on varying the values of N_{qa}. On average, BESF gives 0.58667 value for resource consumption as shown in Table 6.4 which is less than all others.

As shown in 6.9, the questions selected by BESF has higher answerability than MOSFO-GA methodology and others. Answerability is such a factor that it gets increased by increasing the maximum number of questions that can be answered by an expert. The answerability is always higher in case of BESF than that of other recommendation methodologies. When the limit is 30, the answerability

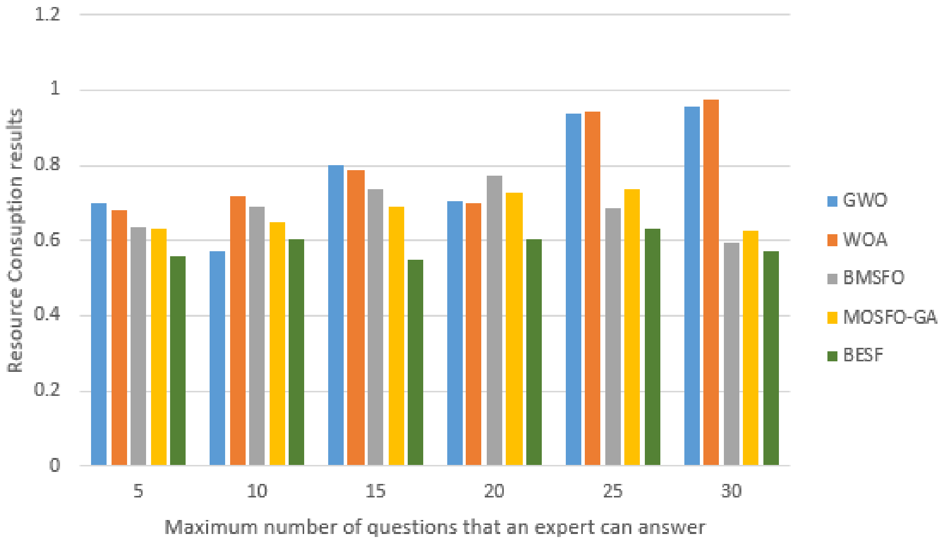


FIGURE 6.8: Expert Resource Consumption of Swarm Intelligence Algorithms

TABLE 6.3: Performance analysis of recommendation techniques

	Coverage	Resource Consumption	Answerability
MOSFO-GA	0.33667	0.52333	0.50333
BESF	**0.43667**	**0.36333**	**0.57167**

factor of BESF is 0.82 and that of MOSFO-GA is 0.75. The higher is the limit, the more space there exists for optimization and the more questions an expert can answer. On an average, BESF gives answerability value of 0.57167 which is greater than all others as shown in Table 6.4.

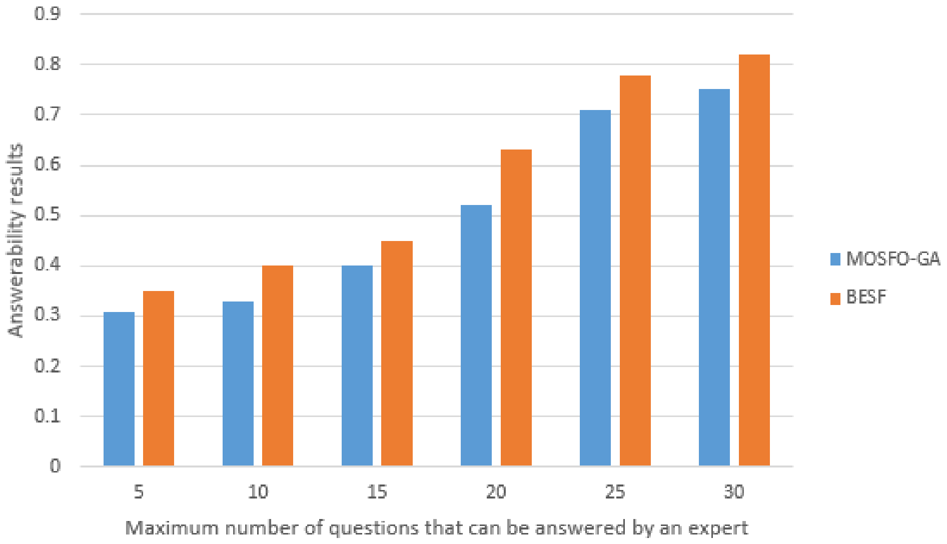


FIGURE 6.9: Answerability of recommendation techniques

Considering the Swarm Intelligence Algorithms, the answerability for BESF is approximately 0.58367 on average which is greater than all other SI techniques as shown in Table 6.4. When $N_{qa} = 25$, the answerability value for GWO is 0.555, WOA gives result value of 0.69, BMSFO gives 0.489, MOSFO-GA has 0.615 answerability value and BESF outlying all others give 0.71 as provided in Fig. 6.10.

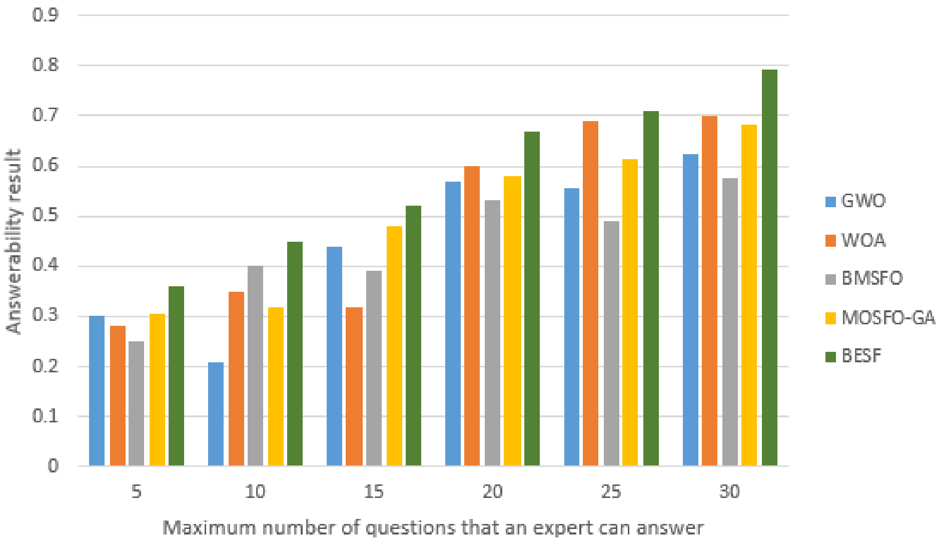


Figure 6.10: Answerability of Swarm Intelligence Algorithms

Although all the recommendation methodologies and SI algorithms not always behave linearly, but still BESF always shows better average value of answerability than that of other mothodologies and algorithms. Hence for all the three parameters i.e. coverage, resource consumption and answerability, BESF shows best results.

Table 6.4: Performance analysis of SI techniques

	Coverage	Resource Consumption	Answerability
GWO	0.24902	0.77775	0.44983
WOA	0.32308	0.80100	0.49017
BMSFO	0.31243	0.68535	0.43933
MOSFO-GA	0.37671	0.67558	0.49733
BESF	**0.42455**	**0.58667**	**0.58367**

Chapter 7

Conclusion

This paper proposes an improved outlook to the batch recommendation of experts in order to answer new questions on StackOverflow, website keeping in view the optimization of using expert resources while providing good quality and high coverage of answers. First, the questions already having answers are modeled and clustered using BERT Topic modeling. Then using the TFIDF values, the similarity between a topic and each new question is calculated. Afterwards, expert ranking is done using activeness, past performance, recent activities and reputation, all of which are sub categorized for better calculation of ranking values. MOSFO-GA is used as an optimization model which gives a matrix of recommendation results for new questions and experts. Furthermore, the approach used is evaluated with data from StackOverflow website and comparison with previous approaches show that the proposed approach is superior in performance.

Future research can be done using additional features for expert ranking. Data from other CQA websites can also be used for further validation of proposed approach.

Bibliography

[1] J. K. Mann, "Semantic topic modeling and trend analysis," 2021.

[2] "3-d diagram for bert transformer," 2019.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] N. M. Khan, U. A. Khan, and M. H. Zafar, "Maximum power point tracking of pv system under uniform irradiance and partial shading conditions using machine learning algorithm trained by sailfish optimizer," in *2021 4th International Conference on Energy Conservation and Efficiency (ICECE)*, pp. 1–6, IEEE, 2021.

[5] C. Feichtinger, S. Donath, H. Köstler, J. Götz, and U. Rüde, "Walberla: Hpc software design for computational engineering simulations," *Journal of Computational Science*, vol. 2, no. 2, pp. 105–112, 2011.

[6] J. Zhang, X. Kong, R. J. Luo, Y. Chang, and P. S. Yu, "Ncr: A scalable network-based approach to co-ranking in question-and-answer sites," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 709–718, 2014.

[7] Z. Ma, A. Sun, Q. Yuan, and G. Cong, "A tri-role topic model for domain-specific question answering," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[8] "State of the stack 2019: A year in review," 2019.

[9] L. Amancio, C. F. Dorneles, and D. H. Dalip, "Recency and quality-based ranking question in cqas: A stack overflow case study," *Information Processing & Management*, vol. 58, no. 4, p. 102552, 2021.

[10] P. Hansen, R. J. Bustamante, T.-Y. Yang, E. Tenorio, C. Brinton, M. Chiang, and A. Lan, "Predicting the timing and quality of responses in online discussion forums," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1931–1940, IEEE, 2019.

[11] M. S. Faisal, A. Daud, A. U. Akram, R. A. Abbasi, N. R. Aljohani, and I. Mehmood, "Expert ranking techniques for online rated forums," *Computers in Human Behavior*, vol. 100, pp. 168–176, 2019.

[12] A. Diyanati, B. S. Sheykhahmadloo, S. M. Fakhrahmad, M. H. Sadredini, and M. H. Diyanati, "A proposed approach to determining expertise level of stackoverflow programmers based on mining of user comments," *Journal of Computer Languages*, vol. 61, p. 101000, 2020.

[13] M. Li, Y. Li, Y. Chen, and Y. Xu, "Batch recommendation of experts to questions in community-based question-answering with a sailfish optimizer," *Expert Systems with Applications*, vol. 169, p. 114484, 2021.

[14] H.-C. Wang, C.-T. Yang, and Y.-H. Yen, "Answer selection and expert finding in community question answering services: A question answering promoter," *Program*, 2017.

[15] J. Wang, J. Sun, H. Lin, H. Dong, and S. Zhang, "Convolutional neural networks for expert recommendation in community question answering," *Science China Information Sciences*, vol. 60, no. 11, pp. 1–9, 2017.

[16] C. Huang, L. Yao, X. Wang, B. Benatallah, and Q. Z. Sheng, "Expert as a service: Software expert recommendation via knowledge domain embeddings in stack overflow," in *2017 IEEE International Conference on Web Services (ICWS)*, pp. 317–324, IEEE, 2017.

[17] Z. Zhao, L. Zhang, X. He, and W. Ng, "Expert finding for question answering via graph regularized matrix completion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 993–1004, 2014.

[18] G. A. Wang, J. Jiao, A. S. Abrahams, W. Fan, and Z. Zhang, "Expertrank: A topic-aware expert finding algorithm for online knowledge communities," *Decision support systems*, vol. 54, no. 3, pp. 1442–1451, 2013.

[19] M. Faisal, A. Daud, and A. Akram, "Expert ranking using reputation and answer quality of co-existing users.," *International Arab Journal of Information Technology (IAJIT)*, vol. 14, no. 1, 2017.

[20] D. P. Mandal, D. Kundu, and S. Maiti, "Finding experts in community question answering services: a theme based query likelihood language approach," in *2015 International conference on advances in computer engineering and applications*, pp. 423–427, IEEE, 2015.

[21] D.-R. Liu, Y.-H. Chen, W.-C. Kao, and H.-W. Wang, "Integrating expert profile, reputation and link analysis for expert finding in question-answering websites," *Information processing & management*, vol. 49, no. 1, pp. 312–329, 2013.

[22] T. P. Sahu, N. K. Nagwani, and S. Verma, "Multivariate beta mixture model for automatic identification of topical authoritative users in community question answering sites," *IEEE Access*, vol. 4, pp. 5343–5355, 2016.

[23] M. Neshati, Z. Fallahnejad, and H. Beigy, "On dynamicity of expert finding in community question answering," *Information Processing & Management*, vol. 53, no. 5, pp. 1026–1042, 2017.

[24] J. Yang, K. Tao, A. Bozzon, and G.-J. Houben, "Sparrows and owls: Characterisation of expert behaviour in stackoverflow," in *International conference on user modeling, adaptation, and personalization*, pp. 266–277, Springer, 2014.

[25] S. Wang, D. Jiang, L. Su, Z. Fan, and X. Liu, "Expert finding in cqa based on topic professional level model," in *International Conference on Data Mining and Big Data*, pp. 459–465, Springer, 2018.

[26] D. Kundu, R. K. Pal, and D. P. Mandal, "Finding active experts for question routing in community question answering services," in *International Conference on Pattern Recognition and Machine Intelligence*, pp. 320–327, Springer, 2019.

[27] M. C. ABOUNAIMA, F. Z. EL MAZOURI, L. LAMRINI, N. NFISSI, N. EL MAKHFI, and M. OUZARF, "The pearson correlation coefficient applied to compare multi-criteria methods: case the ranking problematic," in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pp. 1–6, IEEE, 2020.

[28] S. Wang, D. M. German, T.-H. Chen, Y. Tian, and A. E. Hassan, "Is reputation on stack overflow always a good indicator for users' expertise? no!," in *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 614–618, IEEE, 2021.

[29] D. M. Blei and J. Lafferty, "Topic models. text mining: theory and applications," 2009.

[30] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[31] Y. Feng and M. Lapata, "Topic models for image annotation and text illustration," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 831–839, 2010.

[32] L. Liu, L. Tang, W. Dong, S. Yao, and W. Zhou, "An overview of topic modeling and its current applications in bioinformatics," *SpringerPlus*, vol. 5, no. 1, pp. 1–22, 2016.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[34] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, 1999.

[35] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Research*, vol. 304, p. 114135, 2021.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[38] N. B. Vaghasiya, *Extractive Summarization and Simplification of Scholarly Literature.* PhD thesis, The University of Texas at Arlington, 2020.

[39] L. T. M. Fischer, *Anonymization of Text Data with Attention-Based Networks.* PhD thesis, 2020.

[40] J. L. Boyd-Graber, Y. Hu, D. Mimno, *et al.*, *Applications of topic models*, vol. 11. now Publishers Incorporated, 2017.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[42] J. Alammar, "The illustrated transformer," *The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time*, vol. 27, 2018.

[43] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[45] "Demystifying bert," 2019.

[46] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[47] S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 20–34, 2019.

[48] M. Januszewski and M. Kostur, "Sailfish: A flexible multi-gpu implementation of the lattice boltzmann method," *Computer Physics Communications*, vol. 185, no. 9, pp. 2350–2368, 2014.

[49] A. Shah and M. Padole, "Apache hadoop: A guide for cluster configuration & testing," *International Journal of Computer Sciences and Engineering*, vol. 7, pp. 792–796, 2019.

[50] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.

[51] D. Bairathi and D. Gopalani, "Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: multiple leader salp swarm algorithm," *Evolutionary Intelligence*, vol. 14, no. 3, pp. 1233–1249, 2021.

[52] L. Ni, O. Yang-AiJia, L. Ken-Li, *et al.*, "Improved particle swarm optimization for constrained optimization functions," *Journal of Computer Applications*, vol. 32, no. 12, p. 3319, 2012.

[53] S. A. Xu, S. H. Cui, Y. Zhou, Z. B. Tang, and W. J. Zhu, "The application of modified covariance ekf algorithm to target-tracking modeling," in *Applied Mechanics and Materials*, vol. 427, pp. 953–956, Trans Tech Publ, 2013.

[54] R. Kuo, Y. Syu, Z.-Y. Chen, and F.-C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Information Sciences*, vol. 195, pp. 124–140, 2012.

[55] C. A. C. Coello and G. B. Lamont, *Applications of multi-objective evolutionary algorithms*, vol. 1. World Scientific, 2004.

[56] L. C. Cagnina, M. L. Errecalde, D. A. Ingaramo, and P. Rosso, "A discrete particle swarm optimizer for clustering short-text corpora," *Proc. Bioinspired Optimization Methods and their Applications, BIOMA-2008, Ljubljana, Slovenia*, 2008.