International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

Ranked Stack Overflow: Mathematics & Statistical Analytics

Aboli Wankhade[1], Deekshita Prakash Savanur[2], Gouri Benni[3], Uzair Riyaz Pachhapure[4], Ming – Wang Wang[5]

[1]*Graduate Student, San Jose State University, San Jose, California*
[2]*Graduate Student, San Jose State University, San Jose, California*
[3]*Graduate Student, San Jose State University, San Jose, California*
[4]*Graduate Student, San Jose State University, San Jose, California*
[4]*Professor, Department of Applied Data Science, San Jose State University, San Jose, California*

---***---

Abstract - *In the contemporary era of data-centric professions, the efficacy of Internet-based solution retrieval is paramount. This research endeavors to optimize the process of sourcing accurate and efficient solutions by innovating a ranking metric system within an enhanced Stack Overflow framework. This initiative is spurred by the escalating demand for reliable, hierarchically structured responses in mathematical and statistical domains. Central to this research is the development of a sophisticated application to provide ranked responses. This is achieved through evaluating various parameters: the balance of upvotes and downvotes (focusing on accepted answers), user views, the comprehensiveness of solutions, and user credibility scores. The research aims to deliver optimal solutions to user queries through multi-source data amalgamation, maintaining an average solution quality benchmark of at least 4.0 on a 5.0 scale. A substantial dataset of questions and answers within mathematics and statistics has been collated from Stack Exchange and other platforms. This corpus undergoes rigorous preprocessing and cleansing, using advanced Natural Language Processing (NLP) methodologies. Key mathematical expressions and formulas are meticulously analyzed. Utilizing this dataset, the study employs a model, based on a bespoke ranking algorithm, to discern semantic correlations and contextual nuances. The efficacy of this innovative project is gauged through multiple lenses: the accuracy of the ranking system, the effectiveness of solution retrieval and presentation, and the overall user experience of the application. The culmination of these elements contributes significantly to advancing the field of knowledge retrieval and management in data-intensive disciplines.*

Keywords: Ranking Algorithm, Stack Overflow, Machine Learning Models, Data Preprocessing, Knowledge Retrieval, Natural Language Processing.

1. INTRODUCTION

In the era of the Digital Age, the vast expanse of data structures, underpinned by Mathematics and Statistical Analytics, plays a pivotal role. However, as the digital domain expands, it presents complex challenges for professionals, educators, and learners alike.

Despite the abundance of online resources, a notable fragmentation and lack of cohesion persist. This paper introduces "Ranked Stack Overflow: Mathematics & Statistical Analytics," an innovative platform designed to address these challenges. The project aims to consolidate scattered knowledge into a unified, reliable source, harnessing collaborative, and community-driven approaches. It seeks to establish a digital hub where expertise in Mathematics and Statistical Analytics is shared, validated, and efficiently accessed.

The necessity of such a platform is underscored by the current state of digital resources in these fields. The plethora of available information often leads to confusion, with contradictory answers and a lack of reliable guidance. Our initiative, therefore, focuses on the curation and validation of content, offering users a streamlined experience with prioritized and peer-reviewed information. This is achieved through advanced algorithms that rank solutions based on accuracy, relevance, and community validation. This paper outlines the project's motivation, goals, approaches, methods, and expected contributions. We discuss the challenges inherent in navigating the vast, often disorganized online resources in Mathematics and Statistical Analytics, and how our platform aims to address these through sophisticated data collection, preprocessing, model training, and a novel ranking algorithm.

Additionally, we delve into our methods for evaluating the platform's performance and its potential applications in education, research, and professional practice. By bridging the gap between scattered resources and the need for coherent, comprehensive insights, "Ranked Stack Overflow: Mathematics & Statistical Analytics" endeavors to set a new benchmark for digital knowledge exchange in these critical fields. It aspires to be more than just a question-and-answer platform; it aims to cultivate a community-centric hub for learning, collaboration, and the advancement of knowledge in Mathematics and Statistical Analytics.

2. RELATED WORK

In a study by Setiawan et al. (2021), they developed a method to rate potential answers for fact-based queries in an Indonesian restricted domain question answering system. Their architecture, which included Question Analysis for query creation, Candidate Document Selection, Candidate Document Analysis, Answer Extraction with weighted scores, and Response Generation, merged information retrieval and natural language processing. Utilizing Answer Ranking's weighted scores, their method attained a remarkable accuracy rate of 54%. Advanced NLP techniques, such as morphological to semantic processing, were credited with this improvement. Additionally, removing unnecessary words from user queries could improve efficiency by 15% to 39%.

Using an actual question-answering system, Li et al. (2016) addressed the answer ranking issue by highlighting the significance of ranking answer-bearing sentences correctly. In order to capture both structural relevance and semantic similarity, their method involved creating a composite representation for questions and responses using models from bidirectional long short-term memory (biLSTM) and convolutional neural network (CNN). They included a hypernym method to supplement the training data, improving the robustness of the model. The deep learning-based strategy outperformed earlier approaches that relied on syntactic characteristics and some deep learning models in the evaluation on a TREC benchmark dataset.

The research by Omondiagbe et al. (2022) delves into the assessment of model performance and quality for the prediction of accepted answers on Stack Overflow, a widely used platform for developer questions. The study employs a dataset composed of Stack Overflow posts spanning the years 2014-2016. Four models are employed, including Random Forest and Recurrent Neural Network (RNN) models, both with and without hand-crafted and neural-generated features. The results reveal that the RNN model equipped with hand-crafted and neural-generated features outperforms all other models, achieving an impressive balanced accuracy of 82.73%. The research employs various techniques such as feature engineering, sampling methods like SMOTE and ADASYN, and hyperparameter tuning. Evaluation metrics such as balanced accuracy, F1-score, and Matthews Correlation Coefficient (MCC) are utilized to gauge model performance. Additionally, the study incorporates a developer questionnaire to validate model predictions from a human perspective. Notably, the RNN model with hand-crafted and neural-generated features emerges as the most accurate in predicting accepted answers, offering valuable insights for software developers aiming to integrate Q&A prediction with Stack Overflow.

In Subramani et al. (2023) research paper, the researcher focuses on the prediction of the tags or labels that should be attached in the questions based on what questions are about. For instance, if any user is asking a question about any bugs to be fixed in a specific programming language, their system should automatically suggest the tag name “bug” for the specific programming language. To implement this they have used deep learning methods called Long Short Term Memory, Gated Recurrent Unit (GRU) and Multi-Layer Perceptron that helps the system to learn the text in the questions and suggest right tags for them. This has become useful to help users to organize questions and make it easier to find answers to that question. The evaluation of the model is done using test accuracy, hamming loss, subset accuracy, jaccard score, precision, recall & fl score. After comparing all the models based on the evaluation metrics, GRU algorithm found to be better performing which has highest subset accuracy and lowest hamming loss as compared to other models.GRU algorithm found to be better performing which has highest subset accuracy and lowest hamming loss as compared to other models.

As part of this research, Özyurt and Grethe (2019) unveiled Bio-AnswerFinder, a biomedical question-and-answer system made to handle the difficulties brought on by the growing body of scientific material. They used supervised deep learning approaches for keyword selection and relevance ranking within a greedy iterative retrieval framework to tackle the challenge of document retrieval. A baseline employing syntactic parsing, an ensemble strategy, and an approximation of the k-nearest neighbor method based on word embeddings were all used in comparison to this creative approach and uses deep learning models like Bidirectional Encoder Representations from Transformers (BERT) and Glove word/phrase embeddings, significantly improved Mean Reciprocal Rank (MRR@10) performance.

The paper by Yazdaninia et al. (2021) titled "Characterization and Prediction of Questions without Accepted Answers on Stack Overflow," addresses the growing concern surrounding unanswered questions on Stack Overflow, a prominent platform for programming-related queries. The study conducts a comprehensive examination of factors that influence the likelihood of a question receiving an accepted answer. It introduces innovative features related to tags, content quality, and user engagement, aiming to predict this critical outcome. Utilizing predictive modeling, particularly the XGBoost algorithm, the paper demonstrates the significant impact of these features on predicting whether a question will obtain an accepted answer. The top-performing model achieved an impressive AUC score of 0.70, signifying strong predictive capabilities. Additionally, the research highlights the relative importance of various features, offering insights into the key determinants of question resolution. This paper offers valuable guidance for Stack Overflow users striving to enhance the quality of their questions and researchers interested in understanding the dynamics of online programming communities.

3. METHODOLOGY

Based on Figure 1, the initial phase focuses on the automated generation of tags which is a crucial component for organizing and retrieving information from a vast internet archive. This process begins with a comprehensive data pre-processing step. The raw data retrieved from the internet archive undergoes HTML tag removal, tokenization, conversion of text to lowercase, and the removal of non-alphabetic characters to ensure a clean and standardized dataset for model input. The output of this phase is a set of relevant tags generated for each document, which is later used to assist the retrieval process in the second phase.

In the second phase, the system shifts to the retrieval and ranking of answers based on the tags generated earlier. This phase introduces a Tag-Based Filtering Layer that uses the generated tags as a filter criterion to narrow down the search space for potential answers.

To refine the retrieval process further, a custom Ranking Algorithm Layer is introduced. This proprietary algorithm integrates the scores from both transformer models to prioritize the most relevant answers. This algorithmic layer is designed to weigh the semantic relevance provided by the transformer models, ensuring that the highest quality answers are selected.

Finally, the Output Layer presents the ranked answers to the end-user, completing the architecture's workflow. The output is a list of answers sorted by relevance, directly addressing the user's query with the assistance of the generated tags.

The integration of these two phases establishes a comprehensive system for automated tag generation and tag-based answer retrieval and ranking, which aims to improve the efficiency and accuracy of information retrieval from large-scale data archives.

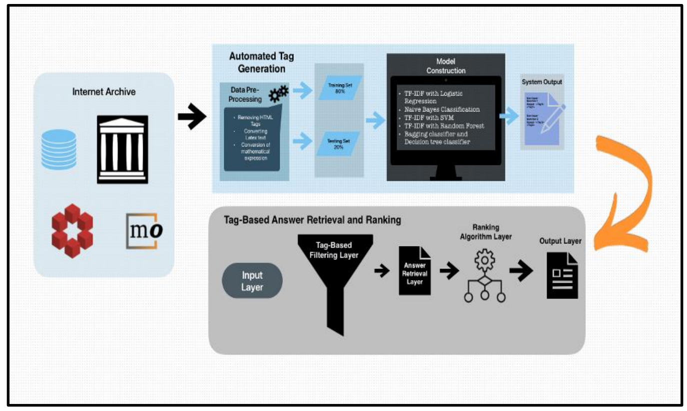

Fig -1: Model Architecture

3.1 Data Collection

Within the Data Engineering framework of our study, the Data Collection process is a cornerstone that sets the stage for our in-depth analysis. This section elucidates the meticulous steps undertaken to gather and amalgamate the datasets pivotal to our research.

Our journey commenced with the Internet Archive, an esteemed nonprofit entity established in 1996, dedicated to the preservation and democratization of a vast array of digital content. This repository, known for its monumental role in archiving a diverse spectrum of media including web pages, books, music, and films, served as our initial data source.

Focusing our efforts on the domain of mathematics, we delved into the Mathematical Stack Exchange, a vibrant community rich with user-generated content. From this platform, we meticulously compiled the Users dataset, a comprehensive collection encompassing details from user profiles. This dataset, with its 1,215,674 rows and 9 columns, provides a granular view of the community's members, including unique identifiers for each user, account creation dates from the year 2012 onward, display names, last access dates, among other pertinent details. This dataset not only offers a snapshot of the community's demographic but also serves as a key element in understanding user engagement and contribution patterns.

Expanding our data collection to encompass the content generated by these users, we aggregated the Posts dataset. This data set amalgamates contributions from several sections of the Stack Exchange network, including MathOverflow and Math.Educators, in addition to the Mathematical Stack Exchange. Encompassing an impressive 3,708,430 rows and 17 columns, this dataset is a treasure trove of insights, with each post uniquely identified and detailed with attributes such as post creation date, score, view count, and more. This comprehensive dataset enables us to delve into the nuances of content creation, user interaction, and information dissemination within these mathematical communities.

Through these datasets, our Data Collection process lays a robust foundation for our subsequent analysis, capturing the dynamic interplay of users and content within the mathematical discourse on the Stack Exchange network. This meticulous approach to data collection not only enriches our research with a multi-dimensional perspective but also underscores the significance of digital archiving platforms like the Internet Archive in facilitating historical and academic inquiries.

3.2 Data Pre-processing

3.2.1 Data Cleaning

As summarized in Figure 2

a) Removing HTML Tags: Essential for text data, especially from web sources, to ensure the content is free of web formatting elements, making it cleaner for text processing and analysis.

b) Handling LaTeX Expressions: The dataset includes mathematical expressions in LaTeX format, which need to be converted into a more readable form for a wider audience. This involves transforming LaTeX symbols into their corresponding mathematical symbols, enhancing the dataset's comprehensibility.

c) Addressing Inconsistent Data: Standardizing date formats is critical for consistency and avoiding errors in time-based analyses. This involves converting various date formats to a uniform standard.

d) Eliminating Noisy Data: Removing extraneous symbols like the '$' symbol, which do not contribute meaningfully to the data, helps in cleaning the dataset for more accurate analyses.

e) Dealing with Incomplete & Missing Data: The approach to handling missing values, especially in columns like "Last Edit Date," depends on the analysis context. In this case, missing values are left as-is due to their relevance to user interactions rather than core data insights.

```
<p>Possibly something like this. Correct me if I'm
wrong.</p>

<p>$j$ = semi-major<br>
$n$ = semi-minor<br>
$e$ = eccentricity</p>

<p>$n = \sqrt{(j\sqrt{1 - e^{2}}) \times (j(1 -
e^{2}))}$</p>
```

Fig -2: Sample code of Data Cleaning

3.2.2 Data Transformation:

a) Feature extraction: It is pivotal in transforming raw data into a format that's more amenable for analysis or machine learning. It involves identifying and isolating significant attributes that contribute most to the dataset's underlying patterns.

b) Separating Questions and Answers: Initially, questions and answers were combined in a single column, making it difficult to analyze them separately. Feature extraction was employed to create distinct columns for questions and answers, facilitating clearer analysis and understanding.

c) Converting Data Type to Suitable Format: Ensuring data types are appropriate for their content is fundamental for accurate and efficient analysis. This includes converting date columns to datetime formats and numerical values to their correct data types, enabling appropriate mathematical and chronological operations.

3.3 Proposed Models

3.3.1 Phase 1: Automated Tag Generation

a) TF-IDF with Logistic Regression: The integration of TF-IDF (Term Frequency - Inverse Document Frequency) with Logistic Regression forms a robust model for automated tag generation. TF-IDF quantifies the importance of a word within a document relative to a corpus, enabling the differentiation of words based on their significance. Logistic Regression, a binary classification algorithm, is then applied to these quantified textual features to predict relevant tags for each document. This model capitalizes on the statistical properties of text to facilitate a nuanced understanding of content, ensuring that the generated tags are both relevant and contextually appropriate.

b) Naive Bayes Classification: Employing the Naive Bayes Classification model for tag generation leverages the probabilistic foundation of Bayes' Theorem, coupled with the assumption of feature independence. This model excels in high-dimensional text classification tasks, making it ideal for analyzing and categorizing textual data into distinct tags. The simplicity and efficiency of the Naive Bayes classifier make it particularly effective in environments where computational resources are limited, without compromising on the accuracy of the tag assignment.

c) TF-IDF with SVM: The combination of TF-IDF and Support Vector Machine (SVM) provides a powerful mechanism for text classification and tag generation. SVM operates by identifying the optimal hyperplane that segregates different classes (tags) in the feature space created by TF-IDF. This method is particularly adept at handling complex classification landscapes where the boundary between different tags is not immediately apparent. The high-dimensional feature space afforded by TF-IDF allows SVM to perform efficiently, even in cases where the number of features surpasses the number of training samples.

d) TF-IDF with Random Forest: Random Forest, an ensemble of decision trees, is utilized in conjunction with TF-IDF for tag generation to enhance prediction accuracy and control over-fitting. Each decision tree in the Random Forest operates on a random subset of features, with the final classification determined by a majority vote across all trees. This approach not only increases the model's robustness to noise but also improves its generalizability to unseen data. The use of TF-IDF ensures that the textual data is suitably transformed into numerical features that can be effectively processed by the Random Forest model.

e) Bagging Classifier: The Bagging Classifier, a meta-algorithm that combines multiple models to improve their collective accuracy, is applied to tag generation by training each base model on a random subset of the dataset. This process, known as bootstrap aggregating, reduces variance and helps to mitigate overfitting, resulting in a more reliable and stable prediction of tags. The ensemble nature of the Bagging Classifier allows it to capture a broader range of patterns in the data, enhancing its ability to accurately classify and generate tags for diverse textual content.

f) Decision Tree Classifier: The Decision Tree Classifier is employed for its intuitive approach to classification, where decisions are made based on the values of input features, leading to a clear and interpretable model structure. In the context of tag generation, the Decision Tree Classifier evaluates the features derived from the textual data to assign appropriate tags. Its hierarchical structure, which mimics human decision-making processes, allows for efficient classification even in the presence of complex and nuanced data relationships.

3.3.2 Phase 2: Tag-based Answer Retrieval and Ranking

a) Sentence Transformers: all-MiniLM-L6-v2 and all-mpnet-base-v2 : For tag-based answer retrieval and ranking, Sentence Transformers like all-MiniLM-L6-v2 and all-mpnet-base-v2 are utilized to encode textual content into dense vector representations. These models, optimized for understanding the semantic content of text, enable the comparison of questions and answers based on their underlying meanings rather than superficial text matches. This approach facilitates the retrieval of answers that are semantically relevant to the questions posed, enhancing the quality and relevance of the information provided to users.

b) Custom Ranking Algorithm: Building on the capabilities of Sentence Transformers, a custom ranking algorithm is developed to refine the answer retrieval process further. This algorithm integrates the semantic similarity scores obtained from models like all-MiniLM-L6-v2 and all-mpnet-base-v2 with additional criteria, potentially including contextual relevance, user feedback, and historical data. The objective is to synthesize these diverse inputs into a coherent ranking that accurately reflects the relevance and utility of each answer to the given question. This bespoke algorithm represents the culmination of the project's analytical efforts, offering a tailored solution to the challenge of information retrieval in large-scale textual datasets.

4. MODEL EVALUATION METHODS

4.1 Precision

Within our evaluation framework, precision stands as a critical metric to gauge the exactness of the model in identifying positive instances. In essence, precision reflects the model's competency in discerning true positives from the sum of all instances classified as positive. It is the quotient of true positive predictions over the collective of true positives and false positives. This measure is particularly pertinent in scenarios where the cost of a false positive is significant, as it provides insights into the model's ability to minimize such occurrences.

4.2 Recall

Recall, also referred to as sensitivity, is employed as a complementary metric to precision. It assesses the model's proficiency in capturing all pertinent positive instances from the available data. Specifically, it calculates the ratio of true positives detected by the model against the total actual positives. This metric is of paramount importance in situations where failing to identify a positive instance carries substantial consequences, as it reflects the model's capability to identify all potential positive outcomes.

4.3 Hamming Loss

Hamming Loss is utilized predominantly in multi-label classification problems. It quantifies the fraction of labels that are incorrectly predicted, thus offering an average measure of the model's error rate across all classes. Hamming Loss is the ratio of the sum of the incorrect label predictions to the total number of labels, which makes it an essential metric for models that output multiple labels simultaneously, providing an aggregate indication of performance across label sets.

4.4 Accuracy

Accuracy is one of the most intuitive performance measures and it provides a straightforward metric for overall model performance. It is computed as the proportion of true results (both true positives and true negatives) in the total number of cases examined. In the domain of classification, accuracy serves as a baseline metric, offering a quick snapshot of the model's effectiveness in correctly predicting outcomes across all categories.

4.5 F1 Score

The F1 Score is a robust metric that balances the precision and recall of the model. It is the harmonic mean of precision and recall, giving a singular score that factors in both the false positives and false negatives. This balanced measure is particularly valuable when there is a need to achieve an equilibrium between precision and recall, ensuring that neither metric is disproportionately weighted in the evaluation of the model's performance.

4.6 Cosine Similarity

Cosine Similarity is a metric that arises from the field of information retrieval and is pertinent to the evaluation of models where the comparison of instances in vector space is necessary. It measures the cosine of the angle between two non-zero vectors of an inner product space, providing an indication of their orientation with respect to one another. In machine learning, this metric can be particularly insightful when assessing the similarity between feature vectors, with a value of 1 indicating vector parallelism and -1 denoting vectors that are diametrically opposed.

5. DISCUSSION AND RESULTS

Our study deployed a suite of machine learning models to categorize and rank answers within a Q&A platform centered on mathematics and statistics. Rigorous evaluation was imperative to discern the most effective model. The evaluation, summarized in Figure 3, employed a comprehensive set of metrics—accuracy, precision, recall, F1 score, and hamming loss—to provide a multifaceted view of model performance.

ML Models	Accuracy	Precision	Recall	F1 score	Hamming loss
TF-IDF with Logistic Regression	0.59	0.96	0.60	0.74	0.04
TF-IDF with Naive Bayes Classifier	0.64	0.96	0.64	0.77	0.06
TF-IDF with SVM (Support Vector Machine)	0.55	0.81	0.59	0.68	0.02
TF-IDF with Random Forest	0.62	0.94	0.64	0.76	0.03
Bagging with Decision Trees	0.64	0.84	0.71	0.77	0.04

Fig -3: Model Performance Metric Values

The TF-IDF with Naive Bayes Classifier emerged as the superior model, demonstrating the highest accuracy (0.64) and an optimal F1 score (0.77), suggesting a robust balance between precision and recall—a critical aspect for a reliable recommendation system. However, a marginally higher hamming loss (0.06) in comparison to the SVM model indicates a slightly elevated rate of misclassification.

Conversely, the TF-IDF with Logistic Regression model, while less accurate (0.59), showcased remarkable precision (0.96). This indicates that while it predicts correct answers less frequently, its predictions are highly reliable when they are correct.

The SVM model, notable for its minimized hamming loss (0.02), signals fewer misclassifications across the board, albeit at the cost of lower accuracy and F1 score.

The Bagging with Decision Trees model struck a commendable balance, exhibiting a high F1 score (0.77) and the second-best recall (0.71), making it a viable model for scenarios prioritizing the retrieval of all relevant answers over minimizing incorrect classifications.

6. CONCLUSION & FUTURE WORK

In an age where data reigns supreme, our research represents a transformative leap in refining the accuracy and efficiency of solution retrieval systems online, particularly within the specialized realms of mathematics and statistics on Stack Overflow. The cornerstone of our endeavor is the novel ranking metric system we've introduced, intricately crafted to serve up structured responses that resonate with the intricate demands of data-focused fields. By harnessing a rich dataset culled from a plethora of questions and answers across diverse forums, and applying state-of-the-art Natural Language Processing techniques, we've fortified the textual data with extracted mathematical notations, enhancing the system's semantic understanding. The ranking algorithm, a bespoke creation, has excelled in identifying contextually pertinent content, validated through stringent statistical scrutiny, bolstering the overall user experience. The project's success, as measured by the precision of the ranking algorithm and the quality of solutions, marks a significant contribution to the field of knowledge retrieval.

Looking to the horizon, our project is poised for expansion, driven by the promise of broader datasets that transcend mathematics and statistics, delving into multifaceted disciplines. The integration of advanced machine learning techniques, including deep learning and reinforcement learning, stands to refine the understanding of user intent and bolster response accuracy. Moreover, personalization is a frontier we seek to conquer, aiming to tailor responses to individual user patterns and historical interactions. The assimilation of real-time data will pivot our system to mirror the dynamic evolution of academic discourse, ensuring continued relevance in a fast-paced digital ecosystem.

Our envisioned trajectory includes fostering a community-powered knowledge exchange, leveraging collective insights to enhance collaborative problem-solving. As we forge ahead, our goal is to not only cement the efficacy of the current system but to redefine the possibilities of knowledge exchange, ensuring our platform remains an indispensable beacon in the landscape of data-centric professions.

REFERENCES

[1] Setiawan, E. I., Santoso, J., & Gunawan. (2021). Answer Ranking with Weighted Scores in Indonesian Hybrid Restricted Domain Question Answering System. Answer Ranking With Weighted Scores in Indonesian Hybrid Restricted Domain Question Answering System.

[2] Li, Z., Huang, J., Zhou, Z., Zhang, H., Shoufeng, C., & Huang, Z. (2016). LSTM-based Deep Learning Models for Answer Ranking. LSTM-based Deep Learning Models for Answer Ranking.

[3] Omondiagbe, O. P., Licorish, S. A., & MacDonell, S. G. (2022). Evaluating Simple and Complex Models' Performance When Predicting Accepted Answers on Stack Overflow. *Evaluating Simple and Complex Models' Performance When Predicting Accepted Answers on Stack Overflow*.

[4] Subramani, S., Rajesh, S., Wankhede, K., & Wukkadada, B. (2023). Predicting Tags of Stack Overflow Questions: A Deep Learning Approach. *Predicting Tags of Stack Overflow Questions: A Deep Learning Approach*.

[5] Özyurt, İ. B., & Grethe, J. S. (2019). Iterative Document Retrieval via Deep Learning Approaches for Biomedical Question Answering. Iterative Document Retrieval via Deep Learning Approaches for Biomedical Question Answering.

[6] Yazdaninia, M., Lo, D., & Sami, A. (2021). Characterization and Prediction of Questions without Accepted Answers on Stack Overflow. Characterization and Prediction of Questions Without Accepted Answers on Stack Overflow.