IgMin Research STEM ISSN: 2995-8067 A Multidisciplinary Open Access Journal

TECHNOLOGY SUBJECT
Machine Learning I Artificial Intelligence I Sensors TOPIC(S)

Article Information

Submitted: December 08, 2023
Approved: January 08, 2024
Published: January 09, 2024

How to cite this article: Faseeh M, Jamil H. Revolutionizing Duplicate Question Detection: A Deep Learning Approach for Stack Overflow. IgMin Res. Jan 09, 2024; 2(1): 001-005. IgMin ID: igmin135; DOI: 10.61927/igmin135; Available at: www.igminresearch.com/articles/pdf/igmin135.pdf

Keywords: Sarcsm; CNN; LSTM; Sentiment; Word2vec; Stack Overflow; RNN (Recurrent Neural Network)

Check for updates

Mini Review

Revolutionizing Duplicate Question Detection: A Deep Learning Approach for Stack Overflow

Muhammad Faseeh and Harun Jamil*

Department of Electronic Engineering, Jeju National University, Jeju-si, Jeju-do, Republic of Korea

***Correspondence:** Harun Jamil, Department of Electronic Engineering, Jeju National University, Jeju-si, Jeju-do, 63243, Republic of Korea, Email: harunjamil@hotmail.com

Abstract

This study provides a novel way to detect duplicate questions in the Stack Overflow community, posing a daunting problem in natural language processing. Our proposed method leverages the power of deep learning by seamlessly merging Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to capture both local nuances and long-term relationships inherent in textual input. Word embeddings, notably Google's Word2Vec and GloVe, raise the bar for text representation to new heights. Extensive studies on the Stack Overflow dataset demonstrate the usefulness of our approach, generating excellent results. The combination of CNN and LSTM models improves performance while streamlining preprocessing, establishing our technology as a viable piece in the arsenal for duplicate question detection. Aside from Stack Overflow, our technique has promise for various question-and-answer platforms, providing a robust solution for finding similar questions and paving the path for advances in natural language processing.

Introduction

In the current digital era, the internet's substantial user base is growing due to technical developments. User-generated, short, subject-focused material is included on social web platforms and question-and-answer websites, which serve as structured hubs for knowledge sharing and interaction. The significance of assessing text similarity is highlighted for tasks like information retrieval, document clustering, and automated essay grading, considering the millions of people who depend on these systems [1]. Research in efficiently identifying and handling brief text is still vital. Social media changes how people communicate with one another. This facilitates knowledge exchange for programmers with coding problems, making Question and Answer (Q&A) websites crucial in the digital age.

Online question forums are popular platforms for people looking for answers and support. Stack Overflow [2], a popular community-driven website for software programming questions, exemplifies the importance of authenticity in these forums. Stack Overflow, founded in 2008, has over 14 million registered users and handles over 10,000 daily questions with a typical response time of 11 minutes. It contains a large repository with 18 million questions, 28 million answers, 75 million comments, and 55 thousand tags [1]. Given the variety of features such as Question ID, Title, Body, and Tags, users may pose comparable queries [3]. Stack Overflow diligently solves this, maintaining a secure and authentic user environment.

Question similarity compares the similarity of two questions by examining their content, structure, and intent. It looks into text similarity, including similarities (paraphrasing, synonymy, implication) and differences (contradictions, antonyms) in semantic relationships. Even for simple queries like "How does Ruby code use the Unary operator?" and "Explain the Ruby Unary Operator," Information Retrieval techniques aid in identifying textual resemblances, a critical role in Question Duplication. Text similarity is a complex yet significant notion since researchers rely on numerous dimensions, including text features, to find shared content characteristics across trials.

Literature review

Detecting Semantic Similarity between Questions on Stack Overflow is critical for most NLP applications. Various strategies are presented for calculating semantic similarity between questions. The

main issue in the Stack Overflow community is identifying semantic commonalities between queries [4].

The difficulties in categorizing brief inquiries are elucidated, encompassing challenges like context-dependent text selection, obstacles in classification, and concerns with automatic coding involving non-standard words and limited term occurrences [5]. These issues illustrate a common problem: the number of duplicate questions on sites like Stack Overflow [3]. This emphasizes the need for a streamlined procedure for discovering and extracting relevant insights. Deep learning has gained popularity in recent years for a variety of Natural Language Processing (NLP) tasks such as paraphrase identification [6], sentiment analysis [7], and question detection [8] utilizing models such as CNN (Convolutional Neural Network) [9,10], RNN (Recurrent Neural Network), and LSTM (Long Short-Term Memory) [11]. These models are critical for problem-solving and performance enhancement, with CNN beneficial for sentence-based classification tasks [12]. For prediction tasks, RNN excels at representing text features and managing word-level inputs [13,14]. These techniques substantially contribute to developments in numerous NLP applications, demonstrating their versatility and effectiveness across multiple domains. The preference for deep learning methods over traditional techniques in text mining for duplicate question detection is highlighted, as the former has shown considerable advancements and superior results [15,16].

Many studies adopt multiple techniques to overcome this type of problem. In a study by Eyecioglu, et al. [17], unigram and bigram features were introduced to detect paraphrases, yielding an F1 score of 0.67. In contrast, using the same data, Mudgal, et al. [18] achieved F1 scores of 0.667 and 0.742 for paraphrase and semantic similarity detection. In a study by Roul and colleagues [19], they introduced SemTF/IDF, a novel similarity measure, to incorporate semantics into class prediction using text-to-text similarity metrics. Meanwhile, Shrivastava and their team [20] utilized support vector machine-based learning, achieving superior F1-scores of 0.717 for paraphrase detection and 0.741 for semantic similarity detection compared to existing systems. Hassanzadeh and associates [21] also applied sentence-level semantic similarity calculations in evidence-based medicine. Soğandoğlu, et al. [22] introduced a sentence-level similarity calculation method in the biomedical question-answering field. Notably, their approach was broad-domain rather than domain-specific. In contrast, Wu, et al. [23] proposed a hybrid model that combined feature engineering with deep learning models, showing superior performance compared to individual models.

Addressing the challenges of categorizing short inquiries requires innovative approaches to overcome their unique characteristics and the high demand for text classification [24,25]. The initial steps involve navigating the context-dependent selection of short question text and grappling with challenges arising from word co-occurrences and context variability. The abundance of short texts poses a bottleneck for conventional classification methods, rendering manual labeling impractical. Furthermore, the automatic coding of short text often falls short of achieving accuracy standards, presenting difficulties in handling these texts effectively. Challenges also arise from non-standard words, including misspellings, grammar issues, and jargon, leading to misconceptions during classification. Additionally, extracting genuine sentiment from short texts proves challenging due to limited-term occurrences, hindering the derivation of reliable conclusions. In addressing these obstacles, developing advanced algorithms, machine learning techniques and NLP solutions becomes crucial for enhancing the efficiency and accuracy of categorizing short inquiries.

Implementation

Combining Long Short-Term Memory (LSTM) with Convolutional Neural Networks (CNN) for duplicate question identification takes advantage of the synergistic benefits of these architectures. LSTM captures long-term dependencies in textual data well, while CNN extracts local patterns and characteristics well. In this fusion, input queries are analyzed independently by the LSTM and CNN components. The LSTM captures the context and relationships between words throughout the sequence, whereas the CNN discovers local patterns using convolutional procedures with varying kernel sizes. These two representations are then joined by concatenation or element-wise addition to form a full feature representation that includes local nuances and long-term interdependence. This fused representation is then employed for classification, making the model more resilient in detecting duplicate questions by considering both local and global textual information, resulting in increased performance in question similarity tasks. Figure 1 showing the complete proposed model.

Experimental results

Our unique deep neural network algorithm was tested on the Stack Overflow community to identify duplicate queries. Our research study aims to discover duplicate queries, and we've used the Stack Overflow dataset. This dataset was made public by Stack Overflow in 2019. There are names of Java Programming Language questions in this collection, some of which may be duplicates while others are not. In this context, "duplicate" questions have the same meaning as a master question. Each question in the dataset has a unique ID and duplicate question pairs are marked by 1, while master question pairs are denoted by 0. Figure 2 depicts the initial rows of the dataset. The dataset contains 416,860 questions, resulting in 208,430 question pairs.

The selection of different models for duplicate question detection

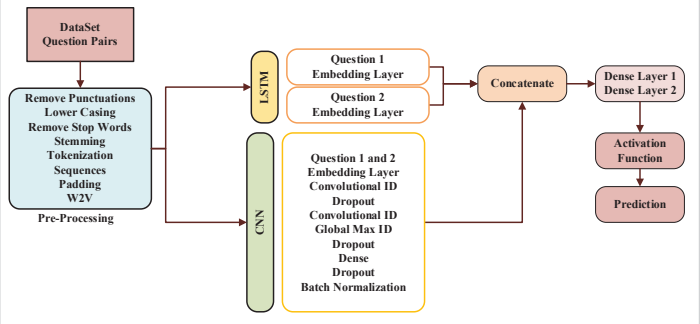

Figure 1: Proposed CNN and LSTM-based Approach.

ISSN 2995-8067 DOI: 10.61927/igmin135

id	qid1	qid2	question1	question2	is_duplicate
0	1	2	use config property file variable	use property property file property file java	0
1	3	4	way use list string parameter regular expression g	java regex capture group x	0
2	5	6	overlap two date range	unable get total decrease	0
3	7	8	stringlength 2 split string	convert dot separate string array list	1
4	9	10	java 8 stream join return multiple value	new line write file	0
5	11	12	pointcut match method annotate directly inherit	spring aop annotation annotation	0
6	13	14	extract value use json path array	limit character command line argument spa	0
7	15	16	call method class inside another class	call object within object	0
8	17	18	insert blob oracle database 11g use vertx	error fetch blob data use jdbc	0
9	19	20	maven nexus find dependent project	sonatype maven repo search project depen	1
10	21	22	write file java	write console output txt file	1
11	23	24	run java web application ii	host java web project ii	1
12	25	26	android javalangnullpointerexception settext sin	label array image icon null pointer error	1
13	27	28	rmi null pointer	null pointer exception use split	1

Figure 2: Initial rows of Stackoverflow dataset.

reflects a thoughtful consideration of various architectures and optimization strategies, each with unique characteristics. Let's delve into the reasons behind the choice of each model and subsequently discuss why Model No. 6, CNN + LSTM with Hybrid Sigmoid/ReLu activation functions and Nadam optimizer, performed the best.

LSTM with Sigmoid and Nadam: Using LSTM with Sigmoid activation and Nadam optimizer is a classic choice for binary classification tasks. Sigmoid activation is well-suited for output layers dealing with binary decisions, and Nadam is an adaptive optimizer that combines the benefits of Adam and Nesterov momentum.

LSTM with Sigmoid and Adam: Similar to above, using Adam as an optimizer provides adaptive learning rates, and Sigmoid activation is appropriate for binary classification. This model explores the impact of a different optimizer while maintaining the Sigmoid activation.

Glove + LSTM with Tanh and Adam: Including Glove embeddings coupled with LSTM using Tanh activation and Adam optimizer suggests exploring pre-trained word embeddings to capture semantic information. Tanh activation is chosen to handle the vanishing gradient problem often associated with LSTMs.

Glove + LSTM with Tanh and Nadam: Similar to the above (Glove + LSTM with Tanh and Adam), this variant introduces Nadam as an alternative optimizer, allowing for comparing the impact of different optimization algorithms.

CNN + LSTM with Hybrid ReLu and Adam: Combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers allows the model to capture local and global dependencies within the input data. CNNs excel at extracting spatial features, while LSTMs effectively capture temporal dependencies, making this architecture suitable for sequential data like text.

CNN + LSTM with Hybrid ReLu and Nadam: The superior performance of the CNN + LSTM model with a hybrid ReLU activation function and Nadam optimizer can be attributed to the synergistic effects of its architecture and optimization choices. The hybrid nature of the model, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers, allows for effective feature extraction by capturing both local and global dependencies in the sequential data. The ReLU activation function applied strategically to the CNN layers, introduces non-linearity, enhancing the model's ability to learn complex patterns. Additionally, using the Nadam optimizer, with its adaptive learning rates and momentum, contributes to efficient convergence and improved generalization. Carefully integrating these elements results in a model that excels in understanding the nuanced patterns indicative of duplicate questions, leading to its superior performance compared to other architectures.

Hyperparameters and their values for the proposed model

We optimized performance by implementing numerous critical hyperparameters in our deep learning model for duplicate question identification, as shown in Table 1. We effectively processed massive datasets with a maximum sequence length of 25 words and a large batch size of 2,048. We set the total amount of words to 150,000 to handle a diversified vocabulary, allowing the model to capture a wide range of textual information. The training was carried out across 25 epochs, with validation accuracy used to track progress. As our loss function, we used binary cross-entropy, which is a good choice for binary classification problems like spotting duplicate questions. Our optimizer of choice was Nadam, which is notable for combining Nesterov momentum and Adam optimization to achieve faster convergence. We used the Rectified Linear Unit (ReLU) activation function to inject nonlinearity into the model, which improved its ability to grasp complicated relationships in the data. We incorporated a 0.2 dropout rate to avoid overfitting and randomly deactivating neurons during training. Finally, we used a learning rate patience of 5 to fine-tune model convergence and get optimal outcomes by adjusting the learning rate during training. These hyperparameters were carefully chosen and modified to guarantee that our model effectively detected duplicate questions. We used these settings for all models, which we implemented after carefully executing each iteration.

Evaluation matrics

Accuracy is the ratio of correctly predicted true positive and true negative samples to the total number of samples. We use accuracy as a standard evaluation Matrix. We have generated the confusion matrix values as well to calculate the accuracy.

Results

We tested numerous methodologies and used deep learning techniques to train the model, as shown in Figure 3. We then evaluated the model's performance on a test dataset. The optimal settings for our models were determined through the meticulous selection of the optimizer, activation unit, and dropout configurations. For the LSTM model with the sigmoid function, the Nadam optimizer achieved an

Table 1: Hyperparamaters of the Applied Models.

Parameter	Value
Max Sequence Length	25
Batch size	2048
Maximum Number Words	1,50,000
Epochs	160 Max
Loss	Binary cross entropy
Optimizer	Adam/Nadam/ReLu
Activation unit	ReLU
Dropout / LR patience	0.2 / 5

ISSN 2995-8067 DOI: 10.61927/igmin135

accuracy of 0.69, while the Adam optimizer resulted in 0.64 accuracy. Utilising Glove + LSTM with the tanh activation function yielded accuracies of 0.4 for Adam and 0.39 for Nadam optimizers. In our proposed CNN-LSTM hybrid model, the ReLU activation function was employed, achieving an accuracy of 0.68 with the Adam optimizer and 0.74 with the Nadam optimizer. The best model train and validation loss is shown in Figure 4. These carefully chosen configurations demonstrate the nuanced impact of different settings on the model's performance across various architectures.

Discussion

The study focuses on enhancing user experiences by identifying duplicate questions. A hybrid model is employed to achieve this, incorporating CNN and LSTM networks with the power of Google Word2Vec for feature extraction. The study offers an advanced deep-learning model carefully constructed to detect duplicate queries in the Stack Overflow community. Our technique seamlessly mixes Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) architectures, specifically designed to address the complexities of this problem. We demonstrate the model's robustness in finding duplicate questions through rigorous experimentation, and we validate its usefulness through extensive validation. Our suggested model outperforms existing approaches, yielding significantly greater accuracy rates. Furthermore, a thorough comparative study was carried out, emphasizing the critical element of semantic similarity identification, producing significant insights into the model's capabilities and prospective applications outside Stack Overflow. We used accuracy to evaluate the model performance. The study focuses on enhancing user experiences by identifying duplicate questions. A hybrid model is employed to achieve this, incorporating CNN and LSTM networks with the power of Google Word2Vec for feature extraction. The effectiveness of this model surpasses previous techniques, showcasing its proficiency in addressing the challenges posed by duplicate questions.

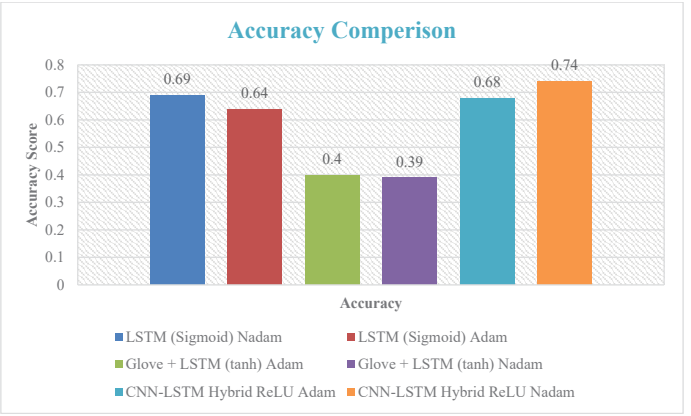

Figure 3: Accuracy Comparison of Applied Techniques.

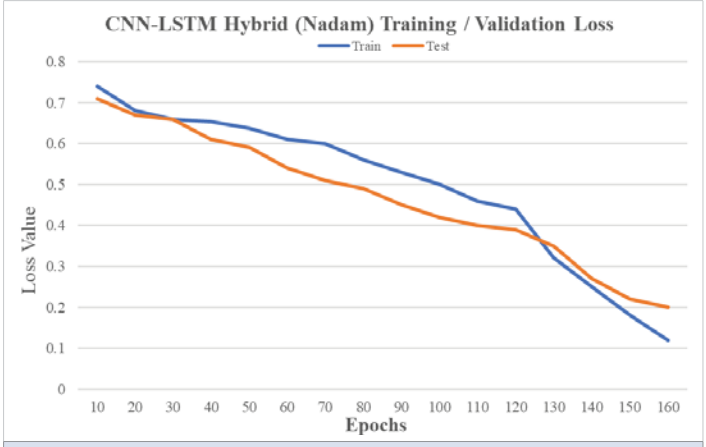

Figure 4: Train/Test Comparison of the proposed technique.

J Huang, et al. [26], Ferreira, et al. [27], Jyun-Yu Jiang, et al. [28], and Yushi Homma, et al. [29] show accuracy of 0.72, 0.69, 0.733, and 70.5, respectively. Our proposal shows a 0.74 accuracy. The hybrid nature of the proposed model, combining CNN and LSTM layers, allows for effective feature extraction by capturing both local and global dependencies in the sequential data. The ReLU activation function applied strategically to the CNN layers, introduces non-linearity, enhancing the model's ability to learn complex patterns. Additionally, using the Nadam optimizer, with its adaptive learning rates and momentum, contributes to efficient convergence and improved generalization. Carefully integrating these elements results in a model that excels in understanding the nuanced patterns indicative of duplicate questions, leading to its superior performance compared to other architectures.

Our proposal can be enhanced if we increase the training size of the dataset. Biases in the dataset used for training and evaluation can occur in case of fewer data. The quality of data is also an essential factor. More quality data with maximum instances can ensure good training results, avoiding the bias factor during the model training.

Conclusion

Our study presents a cutting-edge approach to tackle the pervasive issue of duplicate question detection on platforms like Stack Overflow. By seamlessly merging Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, we have harnessed the power of deep learning to capture both local nuances and long-term relationships within textual input. The integration of word embeddings such as Google's Word2Vec and GloVe has significantly elevated the quality of text representation. Our approach has demonstrated remarkable results through extensive experiments on the Stack Overflow dataset, showcasing its efficacy in addressing the challenges associated with duplicate question identification. Notably, our CNN-LSTM hybrid model, with its carefully selected configurations, outperforms other methodologies, achieving accuracy rates of 0.68 with the Adam optimizer and 0.74 with the Nadam optimizer. These findings highlight the potential of our technique not only for Stack Overflow but also for a wide range of question-and-answer platforms, promising enhanced user experiences and paving the way for advancements in natural language processing. Our research underscores the pivotal role of deep learning in text mining and its capacity to streamline the process of duplicate question detection, contributing to the ever-evolving landscape of knowledge sharing on the internet.

References

1. Ye X, Manoharan S. Marking essays automatically. In Proceedings of the 2020 4th International Conference on E-Education, E-Business and E-Technology. 2020; 56–60.
2. Stackexchange. Stack Overflow Dataset. https://www.kaggle.com/datasets/stackoverflow/ stackoverflow
3. Yazdaninia M, Lo D, Sami A. Characterization and prediction of questions without accepted answers on stack overflow. In 2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC). IEEE. 2021; 59–70.
4. Zhang H, Zeng P, Hu Y, Qian J, Song J, Gao L. Learning visual question answering on controlled semantic noisy labels. Pattern Recognition. 2023; 138:109339.
5. Roy PK, Saumya S, Singh JP, Banerjee S, Gutub A. Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art-review. CAAI Transactions on Intelligence Technology. 2023; 8(1):95-117.
6. Fan M, Lin W, Feng Y, Sun M, Li P. A globalization-semantic matching neural network for paraphrase identification. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018; 2067–2075.
7. Vani K, Gupta D. Text plagiarism classification using syntax-based linguistic features. Expert Systems with Applications. 2017; 88:448–464.
8. Wang L, Zhang L, Jiang J. Duplicate question detection with deep learning in a stack overflow. IEEE Access. 2020; 8:25964–25975.
9. Prabowo DA, Herwanto GB. Duplicate question detection in question-answer websites using a convolutional neural network. In 2019 5th International conference on science and technology (ICST). IEEE. 2019; 1:1–6.
10. Roy PK, Singh JP. Predicting closed questions on community question answering sites using convolutional neural network: Neural Computing and Applications. 2020; 32(14):10555-10572.
11. Chali Y, Islam R. Question-question similarity in online forums. In Proceedings of the 10th annual meeting of the forum for information retrieval evaluation. 2018; 21–28.
12. Kamath CN, Bukhari SS, Dengel A. Comparative study between traditional machine learning and deep learning approaches for text classification. In Proceedings of the ACM Symposium on Document Engineering. 2018; 1–11.
13. Kim Y, Jernite Y, Sontag D, Rush A. Characteraware neural language models. In Proceedings of the AAAI conference on artificial intelligence 2016; 30.
14. Jiang JY, Zhang M, Li C, Bendersky M, Golbandi N, Najork M. Semantic text matching for long-form documents. In The world wide web conference. 2019; 795–806.
15. Imtiaz Z, Umer M, Ahmad M, Ullah S, Choi GS, Mehmood A. Duplicate questions pair detection using siamese malstm. IEEE Access. 2020; 8:21932–21942.
16. Goldberg Y, Levy O. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722. 2014.
17. Eyecioglu A, Keller B. Twitter paraphrase identification with simple overlap features and svms. In Proceedings of the 9th International Workshop on Semantic Evaluation. 2015; 64–69.
18. Mudgal RK, Niyogi R, Milani A, Franzoni V. Analysis of tweets to find the basis of popularity based on events semantic similarity. International Journal of Web Information Systems. 2018; 14(4):438–452.
19. Roul RK, Sahoo JK, Arora K. Modified tf-idf term weighting strategies for text categorization. In 2017 14th IEEE India council international conference (INDICON). IEEE. 2017; 1–6.
20. Dey K, Shrivastava R, Kaushik S. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016; 2880–2890.
21. Hassanzadeh H, Groza T, Nguyen A, Hunter J. A supervised approach to quantifying sentence similarity: with application to evidence based medicine. PloS one. 2015; 10(6):e0129392.
22. Soğancıoğlu G, Öztürk H, Özgür A. Biosses: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017; 33(14):i49–i58.
23. Wu D, Huang J, Yang S. A joint model for sentence semantic similarity learning. In 2017 13th International Conference on Semantics, Knowledge and Grids (SKG). IEEE. 2017; 120–125.
24. Shaheer S, Hossain I, Sarna SN, Mehedi MHK, Rasel AA. Evaluating Question generation models using QA systems and Semantic Textual Similarity. In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC). IEEE. 2023; 0431-0435
25. Amur ZH, Hooi KY, Bhanbhro H, Dahri K, Soomro GM. Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives. Applied Sciences. 2023; 13(6):3911.
26. Huang J, Yao S, Lyu C, Ji D. Multi-granularity neural sentence model for measuring short text similarity. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10177 LNCS. 2017; 439– 455. doi: 10.1007/978-3-319-55753-3_28.
27. Ferreira R, Cavalcanti GDC, Freitas F, Lins RD, Simske SJ, Riss M. Combining sentence similarities measures to identify paraphrases. Comput. Speech Lang. 2018; 47:59–73. doi: 10.1016/j.csl.2017.07.002.
28. Jiang JY, Bendersky M, Zhang M, Golbandi N, Li C, Najork M. Semantic text matching for long-form documents. Web Conf. 2019 - Proc. World Wide Web Conf. WWW 2019. 2019; 795–806. doi: 10.1145/3308558.3313707.
29. Homma Y, Sy S, Yeh C. Detecting Duplicate Questions with Deep Learning. 30th Conf. Neural Inf. Process. Syst. (NIPS 2016), no. Nips. 2016; 1–8. https://pdfs.semanticscholar.org/6ffd/e80e503fe6125237476494e777f4fe6d62c4.pdf

How to cite this article: Faseeh M, Jamil H. Revolutionizing Duplicate Question Detection: A Deep Learning Approach for Stack Overflow. IgMin Res. Jan 09, 2024; 2(1): 001-005. IgMin ID: igmin135; DOI: 10.61927/igmin135; Available at: www.igminresearch.com/articles/pdf/igmin135.pdf