Deep Semantic Matching of Stack Overflow Questions Using Word2Vec and Neural Networks

Mr. K Vivek Vardhan
UG Student, Dept of CSE,
CMR Technical Campus
Hyderabad, Telangana,
India
237r1a05v0@cmrtc.ac.in

Etthidi Vikas
UG Student, Dept of CSE,
CMR Technical Campus
Hyderabad, Telangana, India
237r1a05u2@cmrtc.ac.in

M Navya Sri
UG Student, Dept of CSE,
CMR Technical Campus
Hyderabad, Telangana,
India
237r1a05w1@cmrtc.ac.in

G Swathi
Assistant Professor,
Dept of CSE,
CMR Technical Campus
Hyderabad,
Telangana, India
swathigummadidala67@gmail.com

G Swarnalatha
Assistant Professor,
Dept of CSE,
CMR Technical Campus
Hyderabad,
Telangana, India
gswarnalatha.cse@cmrtc.ac.in

How to Cite this Article:
Vardhan, K. V., Vikas, E., Sri, M. N. & Swarnalatha, G. (2026). Deep Semantic Matching of Stack Overflow Questions Using Word2Vec and Neural Networks. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.345

https://doi.org/10.55041/ijcope.v2i4.345

ABSTRACT— The growth of community-based query answering platforms such as Stack Overflow has led a way to raising duplicate questions, creating redundancy and reduced answer retrieval efficiency and content quality. Manually identifying duplicate question is time consuming and also needs experienced users. Traditional methods based on lexical similarity have mostly failed to cover the semantic gap between syntactically varying but semantically corresponding queries. The new system proposes a same semantic matching system that systematizes duplicate question detection utilizing Word2Vec and neural network architectures. This proposed system preprocesses input data and changes textual content into semantic representations applying Word2Vec. These preprocessed data is then processed via Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) models to capture contextual relationships between question pairs. Performance evaluation is done on metrics like accuracy and recall, with analysis across different models. The results provide an expandable, automated solution for optimizing knowledge management in large-scale technicalities.

INTRODUCTION

Community-based Question Answering (CQA) platforms have become essential resources for developers to share knowledge, solve problems, and collaborate globally. These platforms host millions of questions and answers related to software development. However, with the rapid growth of user-generated content, the issue of duplicate questions has become increasingly prevalent. Duplicate questions not only clutter the platform but also reduce the efficiency of information retrieval and waste valuable resources by repeating already available solutions.

Traditional approaches for duplicate question detection rely on keyword matching, syntactic similarity, or manual moderation by experienced users. Although these methods provide basic filtering, they often fail to capture the underlying semantic meaning of questions. As a result, semantically similar questions expressed using different wording may not be identified as duplicates. Additionally, manual detection is time-consuming and cannot scale effectively with the continuously growing volume of data.

Machine learning techniques such as Support Vector Machines (SVM), Logistic Regression, and Random Forest have been applied to address this problem. While these methods improve detection performance compared to rule-based systems, they still struggle with capturing deep contextual and semantic relationships within textual data. They are also sensitive to feature engineering and may not perform well on large and diverse datasets.

In this project, a deep semantic matching framework is developed using Word2Vec and neural network models such as CNN, RNN, and LSTM. The system performs data preprocessing, feature extraction, and classification to accurately identify duplicate questions. This approach improves the efficiency of question retrieval by capturing semantic similarities, reduces redundancy in question-answering platforms, and provides a scalable solution for managing large volumes of textual data.

I. PROBLEM DEFINITION

Community-based Question Answering (CQA) platforms contain a vast number of user-generated questions, making them valuable resources for knowledge sharing. However, the rapid growth of content leads to a significant problem of duplicate questions, where multiple users post similar queries with different wording. This redundancy increases data volume, makes information retrieval difficult, and reduces the overall efficiency of the platform.

Conventional duplicate question detection methods, including keyword-based matching and traditional machine learning algorithms, often fail to capture the true semantic meaning of text. These approaches suffer from limitations such as low accuracy, inability to detect semantically similar questions with different phrasing, and dependence on manual feature engineering. Additionally, they struggle to handle large-scale datasets and complex textual patterns, leading to ineffective detection performance.

Therefore, there is a need for an advanced and efficient approach that can accurately identify duplicate questions by understanding both syntactic and semantic relationships in text data. This project addresses these challenges by proposing a deep learning-based semantic matching framework using Word2Vec and neural network models such as CNN, RNN, and LSTM, which improves detection accuracy, reduces redundancy, and enhances the overall efficiency of question-answering platforms.

1.1 PROJECT FEATURES

The proposed duplicate question detection system includes advanced features that enhance the efficiency of community-based question answering platforms by accurately identifying semantically similar questions using a deep learning approach. It combines Word2Vec for capturing word-level semantic relationships and neural network models such as CNN, RNN, and LSTM for learning complex textual patterns, resulting in improved accuracy and better generalization. The system applies data preprocessing techniques such as text cleaning, tokenization, and vectorization to improve performance and eliminate irrelevant information. It is capable of handling large-scale textual

data and detecting both exact and semantically similar duplicate questions while reducing redundancy. Additionally, it provides detailed performance evaluation using metrics like accuracy, precision, recall, and F1-score, and offers a scalable, efficient, and cost-effective solution for improving information retrieval and knowledge management in question-answering systems.

Related Work

Several research studies have explored the use of machine learning and deep learning techniques to improve duplicate question detection in community-based question answering platforms. Traditional approaches include keyword-based matching, similarity measures such as cosine similarity, and classical machine learning algorithms like Support Vector Machines and Random Forest. While these methods provide a certain level of effectiveness, they often fail to capture the true semantic meaning of questions and may incorrectly classify semantically similar questions as different.

Recent research has focused on deep learning models for semantic text matching, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks. These approaches are capable of learning complex patterns and contextual relationships from large-scale textual data, leading to improved detection accuracy. Techniques such as Word2Vec have been widely used to generate word embeddings, helping models better understand semantic similarity between questions.

However, many existing solutions face challenges such as high computational complexity, longer training time, and difficulty in handling large and diverse datasets. Some models also struggle to generalize across different domains or fail to effectively capture both word-level and document-level semantics. This project builds upon existing research by proposing a deep learning-based framework that integrates Word2Vec with CNN, RNN, and LSTM models, providing improved accuracy, better semantic understanding, and a more efficient solution for duplicate question detection.

II. METHODOLOGY

1. Data Collection

A Stack Overflow dataset is used, containing pairs of questions labeled as duplicate and non-duplicate.

2. Data Preprocessing

Text data is cleaned by removing stop words, special characters, and noise, then tokenized and converted into a suitable format for processing.

3. Feature Extraction

Word embeddings are generated using Word2Vec to convert textual data into meaningful vector representations capturing semantic relationships.

4. Model Training

Deep learning models such as CNN, RNN, and LSTM are trained on the vectorized data to learn patterns and relationships between question pairs.

5. Classification & Evaluation

The model classifies question pairs as duplicate or non-duplicate and is evaluated using accuracy, precision, recall, F1-score, and confusion matrix.

III. PROPOSED SYSTEM

The proposed system uses a deep learning-based semantic matching approach for detecting duplicate questions by combining Word2Vec with neural network models such as CNN, RNN, and LSTM. It analyses question pairs from the dataset to identify whether they are duplicate or non-duplicate based on their semantic similarity. Word2Vec is used to generate meaningful vector representations of words, while neural networks are used to learn complex relationships and contextual patterns in the text. The system includes preprocessing and feature extraction techniques to handle large and unstructured textual data and improve performance. Compared to traditional methods, it reduces redundancy, improves detection accuracy for semantically similar questions, and provides a scalable and efficient solution for managing large-scale question-answering platforms.

IV. IMPLEMENTATION DETAILS

The implementation of the proposed duplicate question detection system is carried out using both frontend and backend technologies along with deep learning integration. The frontend is developed using HTML, CSS, and JavaScript to provide a simple and user-friendly interface for displaying results and system outputs. The backend is implemented using Python to handle data preprocessing, feature extraction, model training, and duplicate question detection using Word2Vec and neural network models. The system uses a Stack Overflow dataset for training and testing the model. It allows users to input question pairs, process them, classify them as duplicate or non-duplicate, and view the results through an interactive interface.

4.1 ALGORITHMS USED

4.1.1 CONVOLUTIONAL NEURAL NETWORK (CNN)

Deep Convolutional Neural Network (CNN) is a deep learning algorithm used to capture local patterns and features in textual data. It applies convolution operations to extract important features from word embeddings. In this project, CNN is used to identify similarities between question pairs by learning meaningful patterns from the input data, improving duplicate detection accuracy.

4.1.2 RECURRENT NEURAL NETWORK (RNN)

Recurrent Neural Network (RNN) is designed to process sequential data by maintaining information from previous inputs. It helps in understanding the context and order of words in a sentence. In this project, RNN is used to analyze the sequential nature of text and capture contextual relationships between words in question pairs.

4.1.3 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification tasks. It works by finding an optimal hyperplane that separates different classes. In this project, SVM is used as a baseline model to compare performance with deep learning approaches in detecting duplicates.

4.1.4 RANDOM FOREST (RF)

Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their outputs for final prediction. It helps improve accuracy and reduce overfitting. In this project, it is used to classify question pairs and compare results with deep learning-based models.

4.1.5 LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) is a special type of RNN that overcomes the limitations of standard RNN by handling long-term dependencies effectively. It uses memory cells to retain important information over longer sequences. In this project, LSTM is used to better understand semantic relationships in long and complex questions, improving classification performance

V. EXPERIMENTAL RESULTS AND DISCUSSION

The following screenshots represent the execution and performance of the proposed duplicate question detection system. These figures demonstrate the working of different modules such as data preprocessing, feature extraction, model training, and classification. The results clearly show how the system identifies duplicate and non-duplicate questions using Word2Vec and neural network models. It also highlights the effectiveness of the system in improving detection accuracy, reducing redundancy, and ensuring efficient retrieval of relevant information in question-answering platforms.

System Interface – Home Page:

Fig. 1. Home User Interface

The above figure shows the main interface of the system where users can perform all algorithms.

Fig. 2. Final Output Page

Fig. 3. Accuracy Graph

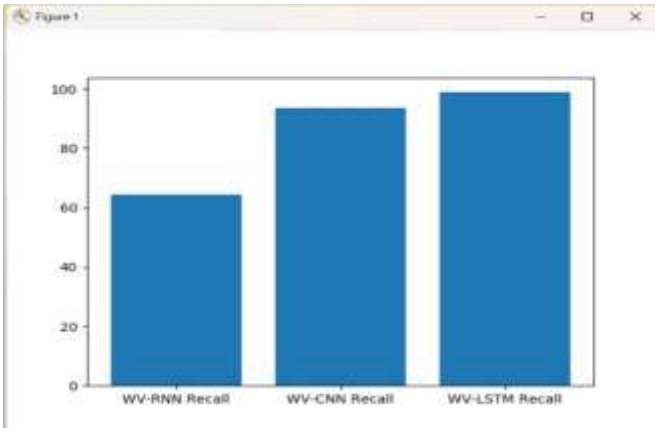

Fig. 3 shows the recall comparison graph of different deep learning models, providing a clear visual representation of their performance. The WV-RNN model achieves a recall of around 65%, indicating moderate performance in identifying duplicate questions. The WV-CNN model shows improved performance with a recall of approximately 95%, as it effectively captures important local features from the text data. The WV-LSTM model achieves the highest recall of nearly 99%, demonstrating its ability to capture long-term dependencies and semantic relationships in question pairs. This graph clearly indicates that the WV-LSTM model outperforms other models in terms of recall, making it more effective for duplicate question detection.

VI. CONCLUSION

This project presents an efficient duplicate question detection system using a deep learning approach that combines Word2Vec with neural network models such as CNN, RNN, and LSTM. The system effectively addresses challenges in traditional methods such as low accuracy, inability to capture semantic similarity, and handling large-scale textual data. By using a Stack Overflow dataset along with preprocessing and feature extraction techniques, the model improves detection performance for identifying both exact and semantically similar duplicate questions. The integration of Word2Vec with deep learning models enhances semantic understanding, improves accuracy, and provides better scalability. Overall, the proposed system offers a reliable and effective solution for duplicate question detection in community-based question answering platforms.

VII. FUTURE SCOPE

The proposed duplicate question detection system can be further enhanced by integrating advanced deep learning techniques for real-time duplicate detection and predictive analysis. Future improvements may include the use of more powerful architectures such as CNN, RNN, or LSTM to better capture complex and sequential patterns in textual data . The system can also be optimized to handle large-scale IoT environments more efficiently and improve detection performance on imbalanced datasets. Additionally, integrating real-time monitoring and automated response mechanisms can help in instantly mitigating detected attacks. These enhancements will make the system more accurate, scalable, and suitable for real-world cybersecurity applications.

VIII. ACKNOWLEDGMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **G Swathi** Asst. Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him/her shall

carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) coordinators **G SWARNALATHA** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **N.BHASKAR** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to **Sri. Ch. Gopal Reddy**, Chairman, **Smt. C. Vasantha Latha**, Secretary and
Sri. C. Abhinav Reddy, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

The guidance and support received from all the members of CMR Technical Campus who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

IX. REFERENCES

[1] Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994). *Network Intrusion Detection*. IEEE Network, 8(3), 26–41.

[2] Staudemeyer, R. C. (2015). *Applying Long Short-Term Memory Recurrent Neural Networks to Intrusion Detection*. South African Computer Journal, 56(1), 136–154.

[3] Vinayakumar, R., Alazab, M., Soman, K. P., & Poornachandran, P. (2019). *Deep Learning Approach for Intelligent Intrusion Detection System*. IEEE Access.

[4] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). *A Deep Learning Approach to Network Intrusion Detection*. IEEE Transactions on Emerging Topics in Computational Intelligence.

[5] Moustafa, N., & Slay, J. (2015). *The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Dataset*. IEEE.

[6] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A Detailed Analysis of the KDD CUP 99 Data Set*. IEEE Symposium on Computational Intelligence for Security and Defense Applications.