

Transformer induced enhanced feature engineering for contextual similarity detection in text

Dakshinamoorthy Meenakshi, Abdul Rahim Mohamed Shanavas

Department of Computer Science, Jamal Mohamed College (Autonomous), Affiliated to Bharathidasan University, Trichy, Tamil Nadu, India

Article Info

Article history:

Received Oct 22, 2021

Revised May 17, 2022

Accepted Jun 27, 2022

Keywords:

Bagging

BERT

Contextual text analysis

Ensemble modelling

Similarity detection

Transformers

ABSTRACT

Availability of large data storage systems has resulted in digitization of information. Question and answering communities like Quora and stack overflow take advantage of such systems to provide information to users. However, as the amount of information stored gets larger, it becomes difficult to keep track of the existing information, especially information duplication. This work presents a similarity detection technique that can be used to identify similarity levels in textual data based on the context in which the information was provided. This work presents the transformer based contextual similarity detection (TCSD), which uses a combination of bidirectional encoder representations from transformers (BERT) and similarity metrics to derive features from the data. The derived features are used to train the ensemble model for similarity detection. Experiments were performed using the Quora question similarity data set. Results and comparisons indicate that the proposed model exhibits similarity detection with an accuracy of 92.5%, representing high efficiency.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Dakshinamoorthy Meenakshi

Department of Computer Science, Jammal Mohammed College (Autonomous)

Affiliated to Bharathidasan University

Trichy, Tamil Nadu, India

Email: rschlrmeena@yahoo.com

1. INTRODUCTION

Detecting duplicate records has been a problem since the beginning of the possibilities brought about by large data storage systems and data warehouses [1]. Duplicate detection is the process of identifying semantically similar entities even if they are syntactically or textually different [2]. Duplication in text or records arise due to several reasons like misspellings, abbreviations, missing data and typographical errors. Such data might prove to be near duplicates, even if not an exact duplicate of the actual data [3]. It becomes a huge challenge to identify this semantically similar data in large databases. Duplicate identification techniques are used in several areas like, approximate matching systems, object identification, information retrieval systems and machine learning methods [4], [5].

Question and answering communities like Quora and stack overflow are the domains that has most applications for duplicate identification techniques [6]. Such communities perform information exchange and knowledge collaboration [7]. The users of these communities post questions and request for answers from other existing users. Several users can post questions, hence, there is a huge possibility of duplication of questions. Duplicate questions exist in this scenario, as the same question can be framed in a different manner by another user. Hence, large number of question duplications exist in these applications [8]. Duplicated questions tend to dilute the results. For users searching for a solution, the search result tends to

yield several similar threads. Hence, a measure to control and eliminate or combine similar questions and their corresponding answers would prove to be a huge advantage for these communities [9]. Users will be able to find all answers in one place, and a user requesting to post questions can verify if the question they try to post already exists in the database [10].

A domain independent model for near duplicate detection has been proposed by Fellah [11]. This is a clustering based model that uses merge filter techniques. The model proposes three algorithms, constant threshold, variable threshold and function threshold. The model works based on the concept of divide and merge. A duplicate identification model mainly used for data cleaning and data merging has been proposed by Hernández and Stolfo [12]. The model uses sorted neighborhood method for duplicate identification. It localizes the neighborhoods do identify near duplicate text. A similar neighborhood based method was proposed by Yan *et al.* [13]. This method is a variation of the sorted neighborhood technique. A duplicate question per detection model using deep neural networks has been proposed by Imtiaz *et al.* [14]. This work uses long short term memory (LSTM) based neural networks for prediction.

A word embedding based duplicate question identification model has been proposed by Babu and Thara S [15]. Word embeddings are applied to the questions, and the numerical question vectors are compared using the cosine similarity metric. The questions are then ranked based on their similarity levels. A model to identify duplicate questions in stack overflow has been proposed by Zhang *et al.* [16]. This work uses the existing question set to identify duplicates from the questions provided by the user. The process of duplicate identification is done based on description, question tags and question titles. An interpretable model for question similarity detection has been proposed by Zhou *et al.* [17]. This technique uses two modularized deep learning models for prediction. Information relevancy is improved using filter operations and pre-trained word embeddings. The process of text matching is performed using vanilla attention and structured attention mechanisms. A neural network based model that uses what the word attention mechanism for natural language inference has been proposed by Rocktäschel *et al.* [18]. An LSTM based model for duplicate prediction was proposed by Chen *et al.* [19]. This work enriches the word level manipulations using attention mechanisms to provide duplicate prediction. Other similar LSTM based models include works by Ghaeini *et al.* [20] and Zhu *et al.* [21].

A hashing based algorithm for duplicate detection has been proposed by Bernstein *et al.* [22]. It proposes the SPEX algorithm to identify and filter non-unique data chunks in the text. A semi supervised model for duplicate detection was proposed by Hazimeh *et al.* [23]. The similarity levels are approximately identified and results are grouped based on the similarities. The similarity levels are identified using semi supervised classification based grouping. A graph theory based model to detect text duplicates has been proposed by Shakeel *et al.* [24]. This technique uses data augmentation strategy and proposes a multi cascaded model. A hidden markov model based technique for duplicate detection in bug reports has been proposed by Ebrahimi *et al.* [25]. Other models operating on semantic similarity levels include works by Li *et al.* [26], and Crouch *et al.* [27].

The key issues identified from the analysis of existing literature are: i) presence of huge amounts of data that is to be analysed; ii) existence of large number of duplicate entities, which exhibit semantic similarities; iii) low syntactic similarities between duplicate entries, making the detection process complex. This work presents a question duplicate identification technique based on ensemble modeling and Transformers. Transformers are used to encode the input data, which provides the features for the machine learning model.

2. TRANSFORMER BASED CONTEXTUAL SIMILARITY DETECTION (TCSD)

Information available in online has become overflowing due to the increased digitalization of information. However, it could also be noted that lots of duplicate information are available online. Content based similarity identification techniques are not sufficient to identify these duplicates, as they are contextually similar, but vary in the words that are used for representing the information. Synonyms for words are used to represent the same information. Hence, direct text matches tend to fail. This work presents a contextual similarity detection technique using transformers, TCSD. The proposed architecture is made up of the text cleaning face, transformer based feature engineering, similarity based feature creation, and bagging based similarity identification. The architecture of TCSD model is shown in Figure 1.

2.1. Text cleaning

Textual statements are the best data used for this process. Text contains punctuations and numerical characters. In the current trends smileys are also used in the text. Smileys are constructed by grouping several symbols. These symbols do not represent any context. Hence, they need to be eliminated from the text. In addition to symbols, extra spaces that are contained in the text are also eliminated. Further, null text is also identified and eliminated. Textual sentences are constructed with sentence cases, and capital letters for nouns.

According to automated models, uppercase and lowercase are treated as different entities. Hence, the entire text is converted to lowercase. At the completion of this process, the text contains only the significant logical entities. Tokens are created using these entities for feature construction.

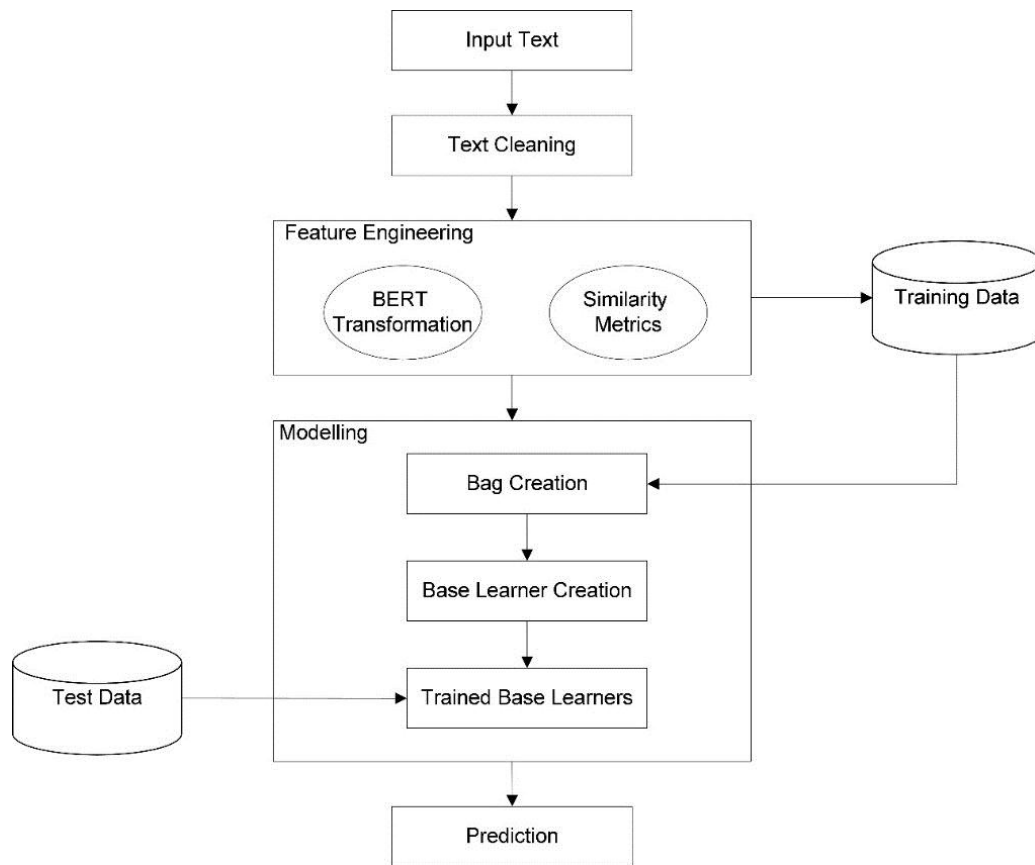


Figure 1. TCSD architecture

2.2. Bidirectional encoder representations from transformers transformer based feature engineering

Feature engineering is performed using the bidirectional encoder representations from transformers (BERT). BERT is a model that is used to encode text for better natural language understanding. Transformers are used to identify context relationships between elements in the text. They are used to perform intent classification. Intent classification requires a major understanding of the context, rather than the content present in the text.

BERT is a machine learning framework that has been developed by Google and was open sourced in 2018. BERT model is a deep learning based architecture in which every output element is connected to every input element and the connection weights are dynamically calculated. BERT comes in two versions, the 12 layer version called BERT base, and the 24 layer version called BERT large. BERT has been constructed using feedforward neural networks. BERT has been trained using a very large corpus of Wikipedia articles and 10,000 books on different genres. The major advantage of using BERT use that it requires list task specific fine tuning.

BERT is used for creating a contextual model from the given text. Representation for word in BERT is based on the word, and also other words contained in the sentence. Contextual modeling is created by capturing the relationships between words in a bi-directional manner. In this work BERT generates vectors of 128 features for each input text. The input vectors and the attention mask are extracted from the BERT model. The input vectors represent the actual textual data. These vectors have a dimension of 128 features. However, not all text will exhibit so many features. Hence, the additional space is padded. The number of padded components in each input vector is provided by the attention mask. The input vectors represent the embedding values. The embedding values and the attention mask are multiplied to obtain the masked embedding. The embedding values represent the features of the text.

2.3. Similarity based feature creation

Embedding vectors generated from the BERT model represents contextual relationship in a single text. Similarity identification is a domain that requires analysis of two textual sentences and identifying the similarity levels between them. Hence, features representing similarity levels are also added to the embedding vectors for better analysis. Cosine similarity levels, and distance metrics are added to the training data. Cosine similarity levels represent the level of match between two documents. The major advantage of cosine similarity metric is that it can effectively calculate the similarity levels even if the documents are of varied sizes. It measures the cosine value of the angle between the vectors created by the text. Cosine similarity between two vectors X and Y can be calculated by:

$$S_{X,Y} = \frac{X.Y}{|X||Y|} \quad (1)$$

Further, the cosine distance value is also considered as another similarity feature. The text embedding vectors obtain from the previous phase are combined with the similarity features obtained in this phase to obtain the training data.

2.4. Bagging based similarity prediction

Similarity prediction is a supervised classification process. Hence the training data is appended with the label that describes the similarity level off the two considered text. Prediction for text similarity is performed using a bagging ensemble model. The training data is divided into multiple overlapping subsets. Each subset contains a part of the data. Every subset has a partial overlap with most of the other subsets. Bagging ensemble technique creates multiple machine learning models, also known as base learners. The best learners are provided with the different data subsets that were created, and trained. Each trained base learner has been provided with a distinct set of training data. Hence, the decision rules obtained in each model is distinct. This work uses decision tree as the base learner. The bagging model created is homogeneous in nature. Hence, multiple instances of decision trees are created. Each decision tree is provided with a distinct subset of the data. During the prediction process, the test data instances are passed to all the models. Every model provides prediction based on the decision rules learned by them during the training phase. This results in multiple predictions for each instance. The final prediction is obtained using the voting technique. The prediction which exhibits maximum votes is considered as the final prediction.

3. RESULTS AND DISCUSSION

Experiments were performed using the Quora data set [28] used for question similarity prediction. The data set is composed of instances containing a question pair and a label depicting their similarity status. The textual questions were used for creating the embedding vectors. The vectors were integrated with similarity features to obtain the training data. The training data was used to model the bagging ensemble for prediction.

Performance of the proposed TCSD model is evaluated based on standard performance metrics such as; TPR, FPR, precision, recall, accuracy and F-score. The ROC plot depicting the true positive rate and false alarm levels of the proposed model is shown in Figure 2. The ROC plot shows that the proposed TCSD model exhibits high TPR levels greater than 0.9 and very low FPR levels less than 0.1. The high TPR levels indicate that the model is highly capable of predicting duplicate sentences. The low FPR levels indicate that the model exhibits low false alarm rates. Low false alarm levels represent that the model exhibits low errors in misclassifying non-duplicate entities as duplicates.

The PR plot representing precision and recall values is shown in Figure 3. Both precision and recall represent the effectiveness of a model in identifying positive classes. In this work positive classes represent duplicate sentences. High precision and recall levels indicate that the model is highly capable of identifying duplicate sentences. The proposed TCSD model exhibits precision levels greater than 0.85 and recall levels greater than 0.9. This indicates that the model can effectively identify duplicate entries in text.

Performance of the TCSD model is compared with the neural network based MaLSTM model proposed by Imtiaz *et al.* [14]. A comparison of the precision and recall levels is shown in Figure 4. The TCSD model exhibits better precision and recall levels compared to the MaLSTM model. Increase precision levels were observed to be approximately 10% and increase in recall levels where observed to be approximately 20%. This performance indicates the TCSD model exhibits better identification of similar question.

A comparison of the aggregated metrics; accuracy and F score as shown in Figure 5. These metrics describe the overall performance of the model in terms of detecting both similar and dissimilar questions. The performance comparison indicates TCSD model performs better in terms of both accuracy and F score.

A tabulated view of the performance is shown in table one. The best predictions as shown in bold. The TCSD model exhibits better prediction in terms of all the discussed metrics. TCSD exhibits and increased accuracy level of 8%, increased precision level of 9%, increased recall level of 21% and increased score of 14%. These performances show that the TCSD model exhibits effective similarity prediction levels. Table 1 shows the performance comparison

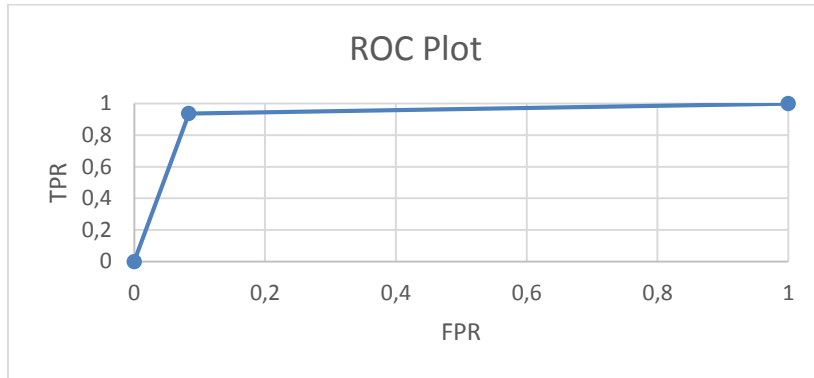


Figure 2. ROC plot

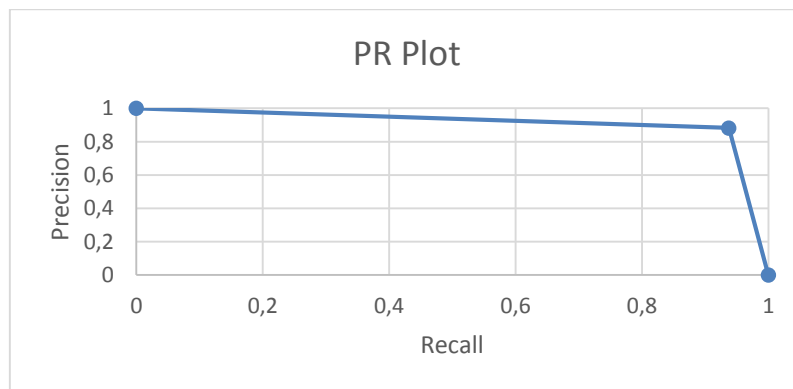


Figure 3. PR plot

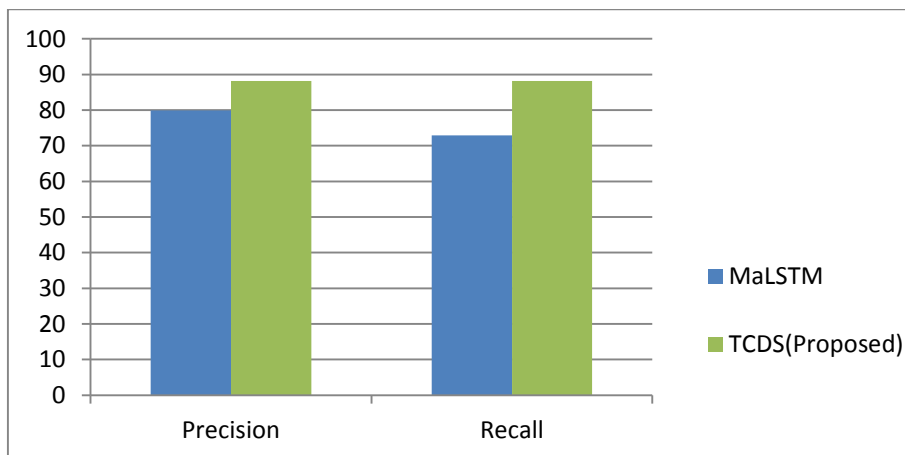


Figure 4. Precision and recall comparison

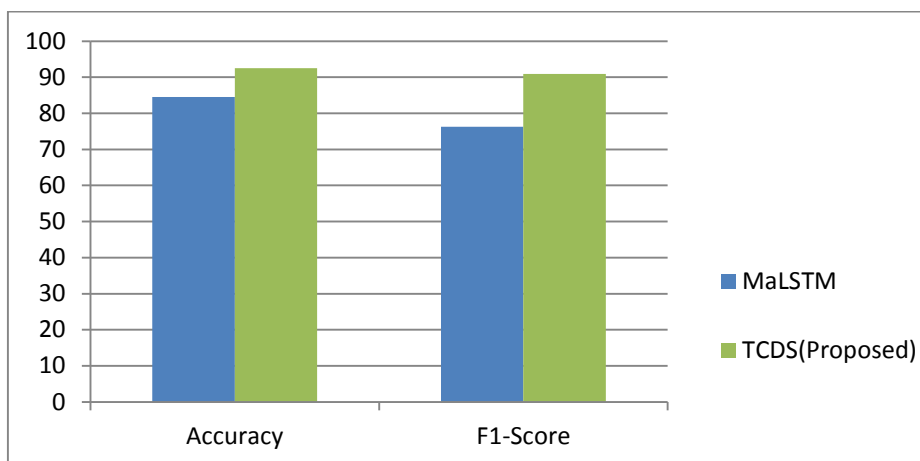


Figure 5. Aggregated metric comparison

Table 1. Performance comparison

Parameter	MaLSTM	TCSD
Accuracy	84.5	92.5
Precision	79.9	88.2
Recall	72.9	93.7
F-Score	76.3	90.9

4. CONCLUSION

Similarity prediction is one of the major requirements of any text processing system that operates on data provided by users. In specific, query answering community systems like Quora and stack overflow require duplicate detection to a large extent. This work presents an ensemble based duplicate detection model that creates text embeddings based on the context of a word in the text. The proposed TCSD model uses transformer based techniques to create the embedding vectors. The embedding vectors are created using Bert transformation, and additional similarity metrics are added to the features to improve the performance levels. The created features are used to train the bagging based ensemble model. Experiments on the Quora data set indicate that the TCSD model exhibits high performance in identifying similar text.




REFERENCES

- [1] H. Newcombe, J. Kennedy, S. Axford and A. James, "Automatic Linkage of Vital Records: Computers can be used to extract follow-up statistics of families from files of routine records," *Science*, vol. 130, no. 3381, 1959, pp. 954-959, 10.1126/science.130.3381.954.
- [2] M. A. Deshmukh and R. A. Gulhane, "Importance of Clustering in DataMining," *International Journal of Scientific Engineering Research*, vol. 7, no. 2, 2016, pp. 247-251.
- [3] M. Azimpour-Kivi and R. Azmi, "A webpage similarity measure for web sessions clustering using sequence alignment," *2011 International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 2011, pp. 20-24, doi: 10.1109/AISP.2011.5960993.
- [4] J. Rashid, S. M. A. Shah, and A. Irtaza, "Impact of Similarity Measures on Web-page Clustering," *American Association of Artificial Intelligence*, 2000, pp. 58-64.
- [5] Rashid, J., Shah, and S. M. A. Irtaza, "A Fuzzy topic modeling approach for text mining over short text," *Information Processing & Management*, vol. 56, no. 6, p. 102060, 2019, doi: 10.1016/j.ipm.2019.102060.
- [6] D. Bogdanova, C. N. dos Santos, L. Barbosa, and B. Zadrozny, "Detecting semantically equivalent questions in online user forums," *Proceedings of the 19th Conference on Computational Natural Language Learning*, 2015, pp. 123-131.
- [7] P. Christen, "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1537-1555, Sept. 2012, doi: 10.1109/TKDE.2011.127.
- [8] M. A. Jaro, "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414-420, 1989, doi: 10.2307/2289924.
- [9] A. E. Monge, "Adaptive detection of approximately duplicate database records and the database integration approach to information discovery," Ph.D. Thesis, Dept. of Comp. Sci. and Eng., Univ. of California, San Diego, 1997.
- [10] S. E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina, "Entity resolution with iterative blocking," in *Proceedings of the ACM international conference on management of data*, 2009, pp. 219-232, doi: 10.1145/1559845.1559870.
- [11] A. Fellah, "All-Three: Near-optimal and domain-independent algorithms for near-duplicate detection," *Array*, vol. 11, p. 100070, 2021, doi: 10.1016/j.array.2021.100070.
- [12] M. A. Hernández and S. J. Stolfo, "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Data Mining and Knowledge Discovery*, vol. 2, pp. 9-37, 1998, doi: 10.1023/A:1009761603038.




- [13] S. Yan, D. Lee, M. Kan, and L. C. Giles, "Adaptive sorted neighborhood methods for efficient record linkage," in *Proceedings of the 7th ACM/IEEE-CS joint conf. on Digital libraries*, 2007, pp. 185–94, doi: 10.1145/1255175.1255213.
- [14] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, "Duplicate Questions Pair Detection Using Siamese MaLSTM," *IEEE Access*, vol. 8, pp. 21932–21942, 2020, doi: 10.1109/ACCESS.2020.2969041.
- [15] J. Babu and Thara S, "Finding the Duplicate Questions in Stack Overflow using Word Embeddings," *Procedia Computer Science*, vol. 171, pp. 2729–2733, 2020, doi: 10.1016/j.procs.2020.04.296.
- [16] Y. Zhang, D. Lo, X. Xia, and Jian-Ling Su, "Multi-factor duplicate question detection in stack overflow," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 981–997, 2015, doi: 10.1007/s11390-015-1576-4.
- [17] Q. Zhou, X. Liu, and Q. Wang, "Interpretable duplicate question detection models based on attention mechanism," *Information Sciences*, vol. 543, pp. 259–272, 2021, doi: 10.1016/j.ins.2020.07.048.
- [18] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kocisky, and P. Blunsom, "Reasoning about entailment with neural attention," in *International Conference on Learning Representations (ICLR)*, 2016.
- [19] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 16571668, doi: 10.18653/v1/P17-1152.
- [20] R. Ghaeini *et al.*, "DR-BiLSTM: Dependent Reading Bidirectional LSTM for Natural Language Inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 1460–1469, 2018, doi: 10.18653/v1/N18-1132.
- [21] X. Zhu, P. Sobihani, and H. Guo, "Long short-term memory over recursive structures," *Proceedings of International Conference on Mach. Learning*, 2015, pp. 1604–1612.
- [22] Y. Bernstein and J. Zobel, "Accurate discovery of co-derivative documents via duplicate text detection," *Information Systems*, vol. 31, pp. 595–609, 2006, doi: 10.1016/j.is.2005.11.006.
- [23] H. Hazimeh *et al.*, "REDUCE: a semi-supervised scalable approach for REsult DUPLICATION detection in Search Engines," *Procedia Computer Science*, vol. 192, 2021, pp. 893–902.
- [24] M. H. Shakeel, A. Karim, and I. Khan, "A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts," *Information Processing & Management*, vol. 57, no. 3, p. 102204, 2020, doi: 10.1016/j.ipm.2020.102204.
- [25] N. Ebrahimi, A. Trabelsi, M. Islam, A. Hamou-Lhadj, and K. Khanmohammadi, "An HMM-based approach for automatic detection and classification of duplicate bug reports," *Information and Software Technology*, vol. 113, 2019, pp. 98–109, doi: 10.1016/j.infsof.2019.05.007.
- [26] J. Li, Y. Han, and Y. Niu, "A Similarity Detection Method Based on Distance Matrix Model with Row-Column Order penalty Factor," *Bulletin of Electrical Engineering and Informatics*, vol. 3, no. 4, pp. 285–292, 2014, doi: 10.11591/eei.v3i4.287.
- [27] N. Crouch and D. M. W. Powers, "Psycholargraph: A graph-based framework for indexing, searching and visualising relationships between academic papers," *The ANU Undergraduate Research Journal*, vol. 161, vol 6, pp. 161–173, 2015.
- [28] "Question Pairs Dataset," *Kaggle.com*, 2021. [Online]. Available: <https://www.kaggle.com/quora/question-pairs-dataset>. [Accessed: 16- Oct- 2021].

BIOGRAPHIES OF AUTHORS



Dakshinamoorthy Meenakshi    is currently a Research Scholar, in the Department of Computer Science, Jammal Mohammed College (Autonomous), Trichy, India. She received her M.Phil Degree in Bharathidasan University, Trichy, India in 2012 and also she is pursuing Ph.D (Computer Science) Bharathidasan University, Trichy. Her area of interest includes big data, cloud computing and so on. She can be contacted at email: rschlmeena@yahoo.com.



Abdul Rahim Mohamed Shanavas    working as an Associate Professor, in the Department of Computer Science, Department of Computer Science, Jamal Mohamed College (Autonomous), Trichy, India. He completed his PhD (Computer Science) Bharathidasan University, Trichy. His area of interest includes IoT, AI, cloud computing, data analytics, image processing and data mining. He can be contacted at email: dramshnavas@yahoo.com.