MIKIDQD: Machine intelligence and knowledge infused duplicate question detection

Vaibhava lakshmi R
Department of Computer Technology, Madras Institute of Technology
Chennai, India
vaibhavi18092002@gmail.com

Gerard Deepak
Manipal Institute of Technology, Bengaluru, Manipal Academy of Higher Education
Manipal, India
gerard.deepak.christuni@gmail.com

Santhanavijayan A
Department of Computer Science and Engineering, National Institute of Technology
Tiruchirapalli, India
vijayana@nitt.edu

Radha S
School of Advanced Sciences, Vellore Institute of Technology
Chennai, India
radha.s@vit.ac.in

***Abstract*—Quora is a flourishing online question answer forum where millions of users post questions and answer questions in the subjects they have knowledge. A plethora of questions are posted by users and some of them can be similarly worded. This would make it tedious for the users as answering similarly worded questions can be time consuming. Hence, there is a need to detect similarly worded questions or duplicate questions. In this paper, we propose the MIKIDQD framework for detecting duplicate questions in the Quora forum using the Quora dataset. The knowledge-centric, ontologybased and semantically-driven strategy incorporated in our model makes it surpass the other baseline models in terms of the performance measures. Our proffered model yields the highest accuracy value of 98.07% and the least False Negative Rate of 0.01 indicating the best performance.**

***Index Terms*—Machine Intelligence, Auxiliary knowledge, Ontology**

I. Introduction

Online discussion forums are sites where users post questions, contribute insights, answer questions posted by others and assess the answers provided by others. Quora is one such global online platform for asking questions and posting answers for them. It is visited by around 300 million users every month. Often the questions posted in Quora might be previously asked. Hence, there is a need to ameliorate user experience by identifying duplicate entries. This can help users to find questions that have been answered already and help community members to avoid answering same questions multiple times. Duplicate Question Detection is a special case of the more general question-question similarity problem. Finding and labeling such potential duplicates manually can be cumbersome. Hence, automatic methods for detecting duplicate questions must be used. These methods mostly incorporate strategies amalgamating Natural Language Processing, Deep Learning and Machine Learning to furnish best results. The traditional Machine learning techniques sometimes perform better than Deep learning techniques. But in general, Deep learning methods are more effectual as they can record the semantics at document level.

Identify applicable funding agency here. If none, delete this.

A. Contribution

In this paper, we have put forth the knowledge-centric and ontology-based scheme for Duplicate Question Detection. Experimentation was done using the Quora dataset. A high density of auxiliary knowledge incorporated from three sources: Google's KG API, LOD cloud and ODP ensure that an enriched taxonomy set is generated from the entities yielded after pre-processing. Later, feature selection is done using Morisita index and classification using Logistic regression classifier is followed which ensures elimination of outliers. The subject-object association after generation of RDF triplets gives large density of auxiliary knowledge. Apart from this, ontologies are generated from the Quora dataset for which the LSI topic modeling framework is applied for all the concepts and sub-concepts which enhances the density of auxiliary knowledge included in the model.

B. Organization

This paper contains content as mentioned: Section II poses the related works and the extant methodologies. In section III, the architecture of the proposed framework is elucidated along with the experimental environment details. In section IV, results of the proffered methodology are discussed and compared with the baseline models. Finally, conclusions are put forth in section V.

II. Related works

Yun Zhang et. al. [1] have proffered an automated methodology named DupPredictor that takes new questions as inputs and detects the possible duplicates by taking into account

multiple factors. This framework incorporates four similarity measures by comparing the titles, topics, descriptions and tags of each pair of questions. These four similarity scores are combined to obtain a new similarity score to yield final results. Di Liang et. al. [2] have proposed an answer information-enhanced adaptive multi-attention network to detect duplicate questions. The strategy involves taking advantage of the semantic details of the paired answers and assuaging noise issues caused by adding the answers. Damar Adi Prabowo et. al. [3] have put forth a model which incorporates semantic similarity of questions using the Glove pre-trained word embeddings. This framework is optimized using Stochastic Gradient Descent. The outputs are compared with Siamese Networks. Nina Poerner et. al. [4] have addressed issue of detecting duplicate questions in Community discussion forums which are domain-specific. The multi-view model MV-DASE amalgamates an ensemble of sentence encoders: generic and domain-specific averaged word embeddings, domain-finetuned BERT and the Universal Sentence Encoder through Generalized Canonical Correlation Analysis, using unlabeled data only. Heng Zhang et. al. [5] have used ensemble learning treating disparate neural networks as individual learners along with a voting strategy to obtain better detection accuracy. Multi-head attention diminishes performance gap and correlation betwixt disparate models. Zhuojia Xu et. al. [6] have proffered a Semantic matching framework amalgamated with the multi-task transfer learning strategy for multi-domain forum duplicate question detection. This model can automatically choose to ignore or give attention to potential kindred words by developing wordto-sentence interaction strategy based on the word-to-word interaction. Liting Wang et. al. [7] have build three techniques namely WV-CNN, WV-RNN and WV-LSTM, which are built by integrating Word2Vec with CNN, RNN and LSTM respectively. The results obtained show that the WV-CNN, WV-RNN, and WV-LSTM models outperform four machine learning approaches based on Support Vector Machine, Logic Regression, Random Forest and eXtreme Gradient Boosting. Qifeng Zhou et. al. [8] have proffered two componentized and comprehensible deep neural network models for detection of duplicate questions. The attention visualization incorporated yields detailed representation at word and sentence level. Alami Hamza et. al. [9] have constructed a DQD framework based on contextual word representation, question classification and forward/backward structured self attention. It incorporates data augmentation technique based on equivalence relations to ameliorate the generalization of the model. Zainab Imtiaz et. al. [10] have incorporated three kinds of word embeddings involving Google news vector embedding, FastText crawl embedding with 300 dimensions, and FastText crawl sub words embedding with 300 dimensions separately to vectorize all the questions and train the model. The final features used for detection are combination of these three kinds of word embeddings. Jeena Jacob [11] has proffered a multi-task learning technique based on Caps-Net for text classification. This framework diminishes interference experienced among disparate tasks in the multi-task learning. Nishesh Awale et. al. [12] have put forth a Machine learning approach using XGBoost model, code style similarity and similarity score of n-grams for detecting plagiarism in programming assignments.

III. Proposed architecture

The architecture of the proffered MIKIDQD framework is depicted in Fig.1. It is a knowledge-centric model for detecting duplicate questions in the Quora website. The Quora dataset is used for the implementation. It is initially subjected to parsing and then questions are extracted. The questions are then pre-processed. Pre-processing involves tokenization, lemmatization, stop word removal, named entity recognition. Also, Word sense disambiguation is done. The dataset is subjected to pre-processing so as to eliminate inconsistencies in the form of punctuations, stop words and typos. For pre-processing, Python's NLTK (Natural Language Tool Kit) was used and for lemmatization, the WordNet 3.0 Lemmatizer was used. A white space special character and period punctuated tokenizer was used. For stop word removal, RegEx based stop word matching algorithm was incorporated. Thesaurus based NER (Named Entity Recognition) was achieved. T The keywords are yielded after preprocessing. Keyword taxonomy formulation is followed. The keywords are subjected to computation of Renyi entropy. The Renyi entropy is an entropy measure computed using logarithm of diversity indices. It can be mathematically represented as:

$$H_\alpha(p) = \frac{1}{1-\alpha} log(\sum_{i=1}^{n} p_i^\alpha) \tag{1}$$

where
p = probability distribution over n € N elements
and
H_α = Renyi entropy of p at order α

The Renyi entropy has a step deviant value of 0.25,i.e, the difference between the Renyi entropy values of two entities must be within 0.25. Based on the values of Renyi entropy for the keywords, a taxonomy is created. The keyword taxonomy is further enriched using three distinct knowledge sources namely Linked Open Data (LOD) cloud through SPARQL using a multi-agent arrangement, Google's Knowledge Graph API and Open Data Project (ODP). From each of these knowledge sources, the entities are furnished and are further formulated into an enriched taxonomy set by again computing the Renyi entropy. Subsequently, the Quora dataset is used to generate the Ontology using OntoCollab. All the generated ontologies are subjected to Latent Semantic Indexing (LSI) for topic modeling of all the core concepts and sub-concepts eliminating the individuals. The individuals are not being subjected to LSI. Auxiliary knowledge is assimilated into the localized framework as a result of Latent Semantic Indexing. The Quora dataset is sent through the RDF distiller to generate the RDF triplet structure which comprises of predicates, subjects and objects. The predicates are heterogeneous and can be a link or sometimes text. The

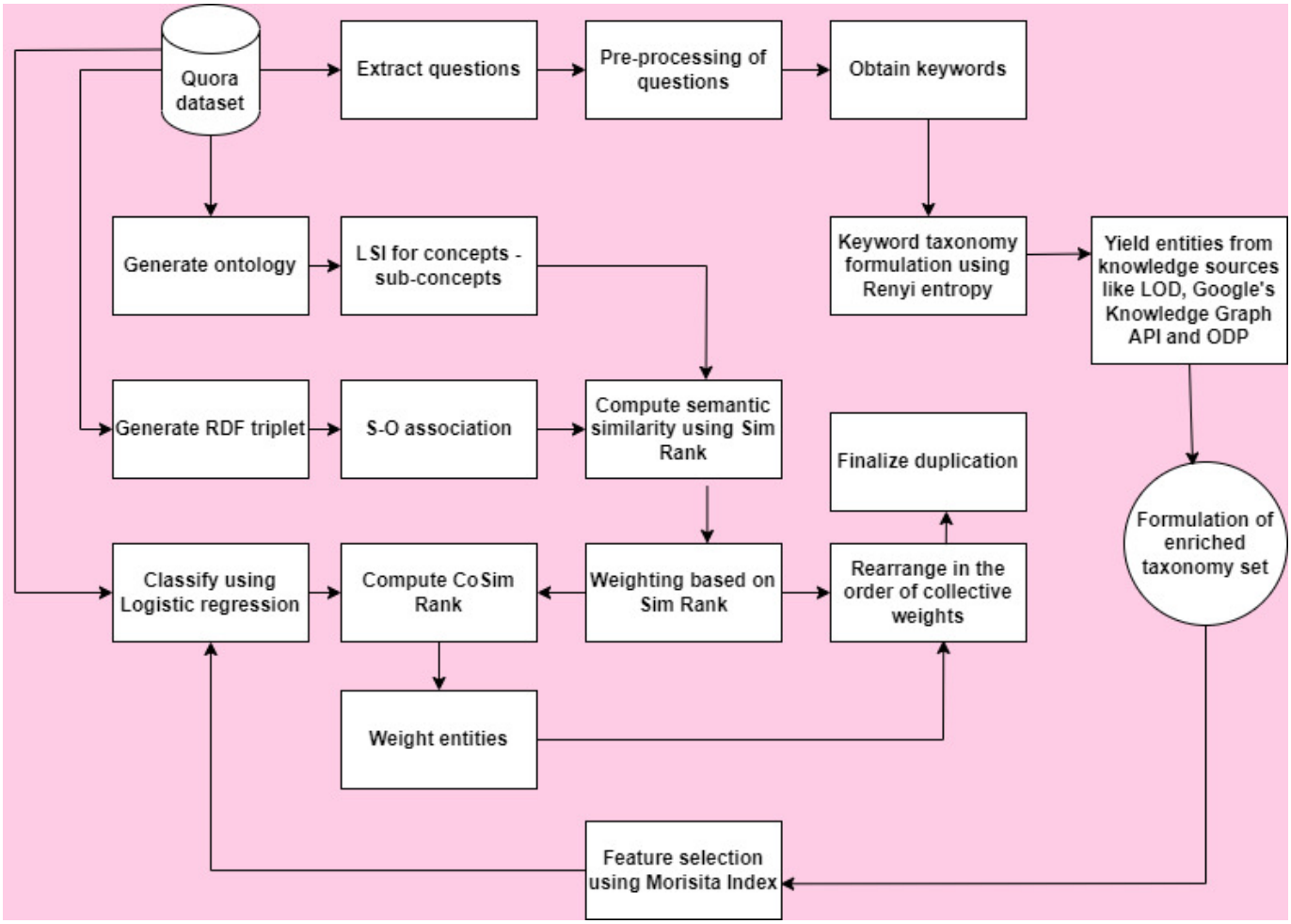

Fig. 1. Architecture of the proposed MIKIDQD framework

semantic similarity is computed between the subjects and objects obtained from LSI by SimRank computation. SimRank is a similarity measure which classifies two entities as similar if they are referenced by similar entities. The equation for the SimRank measure between two entities a and b is given as:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \qquad (2)$$

where C is a constant between 0 and 1.
I(a) and I(b) denote the set of in-neighbours of a and b.

The SimRank score is set to a threshold value of 50%. The weighting of the entities is then done based on the increasing value of SimRank. For every step of 0.05 SimRank values, distinct weights are assigned to the entities. The lowest value of SimRank corresponds to the highest weight assigned. The Quora dataset is also subjected to classification using the Logistic regression classifier. This is a preliminary step towards detecting duplicate questions. The enriched taxonomy set is subjected to feature selection using the Morisita Index and then classification is done using the Logistic regression classifier. The Logistic regression is a supervised classification algorithm in which the target output value can take only discrete values for a given set of features. A decision threshold value is used which is set based on the classification problem itself. The Morisita Index is a measure of how kindred or different two sets of data are. It can be mathematically given as:

$$C_D = \frac{2\sum_{i=1}^{S} x_i y_i}{(D_x + D_y)XY} \qquad (3)$$

where
S = number of unique items
D_x and D_y are the Simpson' s Diversity indices for samples 1 and 2
x_i and y_i are the number of times an item appears in samples 1 and 2
The outcome of the Logistic regression classifier and the entities weighted based on SimRank are further used to compute the CoSimRank. The CoSimRank is a variant of SimRank measure. The CoSimRank measure can be represented in matrix form as:

$$S = C.(A^{\top}.S.A) + I \tag{4}$$

where
S is the similarity matrix with $S_{[a,b]}$ denoting the similarity score between entities a and b.
A is column normalized adjacency matrix and I is the identity matrix.
The CoSim rank is set to a threshold of 75%. The entities furnished after calculation of the CoSimRank are weighted corresponding to the increasing order of CoSimRank values. The entities are then rearranged using collective weights according to the weighting done using CoSimRank and SimRank. Finally, all the questions having entities with equal weights are selected as duplicates and then sent for review.

IV. Performance Evaluation and Results

TABLE I
Comparison of Performance of the Proposed MIKIDQD with other Approaches

Model	Precision	Recall	Accuracy	F-measure	FDR
MDQD	92.23	93.06	92.64	92.64	0.07
MANQD	92.01	93.45	92.73	92.72	0.07
DQDCNN	93.04	94.13	93.58	93.58	0.06
MDASDQD	93.95	94.85	94.40	94.39	0.06
Proposed MIKIDQD	97.03	99.12	98.07	98.06	0.01

The performance of the proffered MIKIDQD framework is evaluated using Precision, Recall, Accuracy and Fmeasure for quantifying the relevance of the results, and the FNR (False Negative Rate) measure for finding the probability that a true positive will be missed by the test.

$$Accuracy = \frac{TP + FP}{TP + FP + TN + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where
TP = number of positive classes correctly predicted
FP = number of positive classes incorrectly predicted
TN = number of negative classes correctly predicted
FN = number of negative classes incorrectly predicted

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

$$FNR = \frac{FN}{TP + FN} \tag{9}$$

where
TP = number of positive classes correctly predicted
FP = number of positive classes incorrectly predicted
TN = number of negative classes correctly predicted
FN = number of negative classes incorrectly predicted

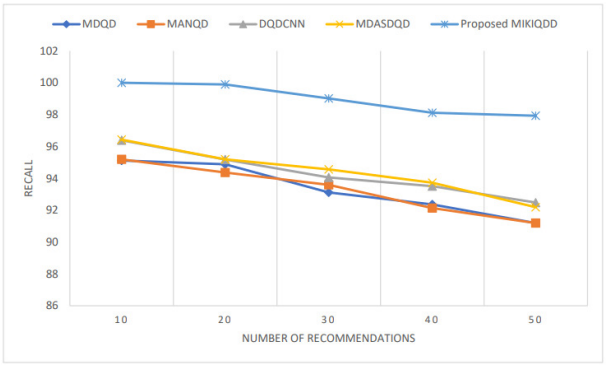

Fig. 2. Recall versus number of recommendations curve

In order to rate the performance of the proffered model, it is baselined with the MDQG [1], MANQD [2], DQDCNN [3] and MDASDQD [4] models respectively. It is indicated from Table.1. that the proffered MIKIDQD framework furnishes highest precision, recall, accuracy and F-measure percentages of 97.03, 99.12, 98.07 and 98.06 respectively and the lowest FDR of 0.01. The reason why the MIKIDQD framework outperforms the baseline models is due to the fact that it incorporates semantically inclined strategy with machine intelligence. Machine intelligence is incorporated by using Logistic Regression which has a feature controlling mechanism which is integrated with several intelligence estimation techniques like the topic modeling for enhancing auxiliary knowledge, the SimRank and CoSim Rank model for computing the semantic similarity. CoSim Rank and Sim Rank are set with a specific threshold value which incorporate inferencing. Owing to this reason, a Machine intelligence strategy is incorporated in the MIKIDQD framework. The MIKIDQD model is RDF-driven in which a triadic tuple is proffered. RDF has subject, predicate and object to yield strong lateral knowledge entities. However, the predicate is dropped owing to the presence of large amount of heterogeneity. The subject and object co-occurrence is retained in the RDF triple. The subject-object association itself provides large density of auxiliary knowledge. Apart from this, ontologies are generated from the Quora dataset for which the LSI topic modeling framework is applied for all the concepts and sub-concepts which enhances the density of auxiliary knowledge included in the model. The integration LOD cloud, Google's KG API and ODP to harvest entities with a high degree of heterogeneity to formulate an enriched taxonomy set ensures high degree of lateral knowledge. The weighting of entities from the CoSim Rank and the Sim Rank ensures a weighted scheme for yielding collective weights per questions in order to predict the duplicates much more accurately. The Logistic regression classifier which is a Machine learning classifier is chosen in place of a Deep learning classifier because the latter has a tendency to learn outliers from the dataset since the features are not controlled. Hence, the Logistic regression classifier would furnish results with no deviants.
The reason why the MDQD model doesn't perform as good as the proffered model is that, although features like titles,

descriptions, topics and tags are considered, there is a lot of monotony. Similarity scores have been used comprehensively. However, there is no heterogeneity in the semantic score threshold regulation. Only inferencing is incorporated with no auxiliary knowledge-based learning. The reason why the MANQD model is transcended by the proffered framework as it is computationally complex and learning takes place from dataset alone. There is no controlled mechanism for feature selection, i.e., implicit feature selection takes place. There is a high possibility of learning outliers. Knowledge incorporation is sparse. Also, there is no relevance computation mechanism involved. The DQDCNN framework is also transcended by the proffered MIKIDQD model. Although the deep learning CNN classifier is used along with Glove pre-trained word embeddings which incorporates partial auxiliary knowledge, outliers won't be eliminated. Our proposed model also outperforms the MDASDQD framework. The sentence embeddings in the latter strategy ensures incorporation of some amount of subsidiary knowledge. But relevance computation mechanism must be stronger.

V. Conclusion

With the advent of the Internet, sundry question-answer forums have come up where people can share knowledge. The interactive nature of the Quora question and answer forum causes digital flourishment through content generated by users, whilst simultaneously giving customers a platform to deliver official, authorized, problem-solving solutions in the form of answers which are accepted. To avoid ambiguity for users while using Quora, there is a need for detecting duplicate questions. We have put forth the MIKIDQD framework which is ontology-driven, knowledge-centric and Semantically-inclined. The incorporation of three different auxiliary knowledge sources namely the LOD cloud, Google's KG API and ODP ensure that there is high volume of knowledge incorporated. Also, the relevance computation mechanism used is very efficacious. Hence, our model transcends other baseline models by yielding an accuracy score of 98.07%.

References

[1] Zhang, Y., Lo, D., Xia, X., Sun, J. L. (2015). Multi-factor duplicate question detection in stack overflow. Journal of Computer Science and Technology, 30(5), 981-997.

[2] Liang, D., Zhang, F., Zhang, W., Zhang, Q., Fu, J., Peng, M., ... Huang, X. (2019, July). Adaptive multi-attention network incorporating answer information for duplicate question detection. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 95-104).

[3] Prabowo, D. A., Herwanto, G. B. (2019, July). Duplicate question detection in question answer website using convolutional neural network. In 2019 5th International Conference on Science and Technology (ICST) (Vol. 1, pp. 1-6). IEEE.

[4] Poerner, N., Schütze, H. (2019, November). Multi-view domain adapted sentence embeddings for low-resource unsupervised duplicate question detection. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 1630-1641).

[5] Zhang, H., Chen, L. (2019, November). Duplicate Question Detection based on Neural Networks and Multi-head Attention. In 2019 International Conference on Asian Language Processing (IALP) (pp. 13-18). IEEE.

[6] Xu, Z., Yuan, H. (2020). Forum duplicate question detection by domain adaptive semantic matching. IEEE Access, 8, 56029-56038.

[7] Wang, L., Zhang, L., Jiang, J. (2020). Duplicate question detection with deep learning in stack overflow. IEEE Access, 8, 25964-25975.

[8] Zhou, Q., Liu, X., Wang, Q. (2021). Interpretable duplicate question detection models based on attention mechanism. Information Sciences, 543, 259-272.

[9] Hamza, A., Ouatik, S. E. A., Zidani, K. A., En-Nahnahi, N. (2020). Arabic duplicate questions detection based on contextual representation, class label matching, and structured self attention. Journal of King Saud University-Computer and Information Sciences.

[10] Imtiaz, Z., Umer, M., Ahmad, M., Ullah, S., Choi, G. S., Mehmood, A. (2020). Duplicate questions pair detection using siamese malstm. IEEE Access, 8, 21932-21942

[11] Jacob, I. Jeena. "Performance evaluation of caps-net based multitask learning architecture for text classification." Journal of Artificial Intelligence 2, no. 01 (2020): 1-10.

[12] Awale, Nishesh, Mitesh Pandey, Anish Dulal, and Bibek Timsina. "Plagiarism Detection in Programming Assignments using Machine Learning." Journal of Artificial Intelligence and Capsule Networks 2, no. 3 (2020): 177-184.