

Learning Improved Class Vector for Multi-Class Question Type Classification

Tanu Gupta^{1,*} Ela Kumar²

^{1,2} Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women

*Corresponding author. Email: tanu92gupta@gmail.com

ABSTRACT

Recent research in NLP has exploited word embedding to achieve outstanding results in various tasks such as; spam filtering, text classification and summarization and others. Present word embedding algorithms have power to capture semantic and syntactic knowledge about word, but not enough to portray the distinct meaning of polysemy word. Many work has utilized sense embeddings to integrate all possible meaning to word vector, which is computationally expensive. Context embedding is another way out to identify word's actual meaning, but it is hard to enumerate every context with a small size dataset. This paper has proposed a methodology to generate improved class-specific word vector that enhance the distinctive property of word in a class to tackle light polysemy problem in question classification. The proposed approach is compared with baseline approaches, tested using deep learning models upon TREC, Kaggle and Yahoo questions datasets and respectively attain 93.6%, 91.8% and 89.2% accuracy.

Keywords: *Class-specific vector, Deep Learning Models, Polysemy, Question Classification, Word2vec.*

1. INTRODUCTION

The massive growth of online data day by day has created the demand of automatic question answering system to retrieve accurate and precise answer for user's query. Many researchers have shown interest in developing QA systems for various applications such as Alexa and Siri. Question classification (QC) is an important element in designing an outstanding question answering system, which categorize question either on its answer type or to the related topic that helps in selecting relevant documents to answer it. Various previous studies [1,2] has used vector space model (VSM) to map documents into points in a vector space that embrace the relevance of its terms. Word2vec, Glove, Fasttext and many algorithms exist to implement VSM for text classification, but they are not capable to find exact sense of polysemy word. Word Embedding algorithms has produce one vector representation for a word, irrespective of its sense change with occurrence. The polysemy is a common issue in text classification, where a word can have multiple sense. This ambiguity problem is also seen in question classification. Because of polysemy words the actual meaning of user's question can be misunderstood and corresponding generated answer may not be accurate. Some acclaimed solutions for handling polysemy problem are present in existing

work such as context vector [3,4] and incorporating every possible sense vector [5,6] to word vector. Despite their significance these solutions suffer from sever drawbacks. From small dataset it's hard to interpret correct word sense from contextual information, therefore context2vec fails to identify the actual word meaning. On the other hand, sense embeddings try to capture all possible sense of word irrespective of application and domain. Then, all possible combinations of sense vectors are applied in NLP task which causes high training computation cost.

Sicong *et al.* [7] has proposed an idea to encapsulate class information into a vector to represent a class. This technique has proved successful in overcoming drawbacks of commonly used context and multiple sense embeddings. For each class a class vector is generated, those are further merged with general word embeddings to create single embedding for a word per class. In short each word has multiple embeddings according to the number of classes to solve polysemy problem in text classification.

Despite the advantage of above work in comparison of other existing approaches, we observe some critical drawbacks. First, the class specific vectors approaches target only binary categorized datasets for classification task. Secondly, approach is class label dependent, means based on meaning of class label.

This paper has introduced an approach that overcome the drawback of work mention in [7] and other commonly used techniques. Our approach has generated improved class vectors those are independent of class labels and also aim to classify multi-class categorized datasets. In contrast to previous work, the proposed approach can generate class vectors for questions labelled with its answer type such as yes/no, factoid, list and summary and also questions labelled with their domain such as business, entertainment and so on. For example, in previous work, tweets were collected form two classes; hurricane related and unrelated. The class vector for hurricane related tweet is based on the words having least cosine similarity with label 'hurricane'. This approach doesn't work for questions classified with their expected answer type. Therefore, a general methodology has been proposed in this paper to produce class vectors those are not depend on class label. Our work has been tested on both types of datasets; TREC questions are classified with its semantic class, Kaggle questions are classified with topics or issues related to python such as run time error, installation etc. and Yahoo Questions are classified with list, summary, factoid and yes/no answer type.

Our proposed approach has used modified skip gram model to generate word vectors and improved tf-idf score (*tf-idf-cf*) to express class information for question classification. With the experiments performed on mentioned datasets using improved class-specific vector representation, helps to infer the excellence of proposed approach to tackle light polysemy problem in question classification task. The brief of the contribution has been made in this paper:

- Our work is first to utilize class specific word embeddings for multi-class question classification.
- Unlike previous approach, class vectors are unrelated to class labels.
- Our work has targeted polysemy problem in question classification task.

The paper is structured as, section 2 has discussion about related literature survey, feature extraction methods are explained section 3, proposed approach is mentioned in section 4, experiment setup, datasets description and results is given in section 5, section 6 has made conclusion of the paper with insight into future work and then references.

2. RELATED WORK

Question classification is a vital step of question processing module in question answering system. This helps to find out the type of answer expected by the user and also it's related documents. This section describes the existing work in question classification. The techniques involved in classifying questions are placed in three groups: rule based, machine learning and hybrid techniques. The rule base or hand craft methodology is

the very basic approach for classification in which experts extract words or their combinations as features. In [8], Biswas and others has designed syntactic patterns of questions in TREC QA track. TREC QA track is the collection of questions which are categorized by Li and Roth into two levels; 6 coarse classes and 50 fine classes. The syntactic rule based approach has achieved ~98% accuracy for classification. Another rule based approach has added semantic hand crafted rule with syntactic rules in [9] on same dataset and tested these rules using machine learning models. The linear SVM gives 91.6% accuracy and manual approach has given 97% accuracy. Although rule based approaches are more accurate in classification task, but this practice is taking enormous efforts and time. With machine learning models, these drawbacks can be overcome and more latent features of text can be extracted. [10] has explored deep learning models such as CNN, LSTM, hybrid framework, for question classification using word2vec algorithm on Turkish translated UIUC English dataset. These models are tested upon word vector generated using skip gram embedding model and CBOW model. This study has investigated that skip gram model's word vectors has brought highest accuracy with CNN model. The RNN model and its variants has also gained popularity [11,12] in the same task. Stefan and others [11] has developed algorithm for accessing the question quality using sequence model. The questions are the collection of student feedback from a tutoring system, iSTART. Total collected questions are 4575 and are manually coded between: very shallow (1) to very deep (4). The experiments are implemented on multiple RNN models such as GRU, Bi-GRU and LSTM using Glove and FastText embeddings. The Bi-GRU model gives best performance with 81.22 % accuracy. Bi-LSTM has also given improved results on question collection of about daily meetings and conversation [12]. The study has evaluated accuracy and loss function value as 90.9% and 0.316 respectively. The question classification on Chinese dataset has tested word2vec algorithm with attention based deep learning model and compared results with other models such as CNN, LSTM and Bi-GRU [13]. From results, it is discovered that precision of attention based LSTM and Bi-GRU CNN model is same, but the f-score of attention based Bi-GRU CNN is highest among all models i.e. 0.784. Various previous works have contributed in word2vec algorithm [14,15,22] to give a new direction in performing different NLP classification tasks. But all the existing techniques generate one embedding per word, which face a fall in determining correct sense of word, when word actually has multiple senses called polysemy words.

To eliminate polysemy problem, several models have been introduced to induce multiple-embedding for a word, multiple-embeddings were trained on machine learning classifiers [17,16]. One way to discriminate between different senses of word is learning context of

target word. The k-means clustering algorithm used by Huang *et al.* [18] has empirically declared k-senses for each ambiguous word. Local contexts of word are grouped into k-clusters, which limits the knowledge that we can gather for distinguishing the related sense. Neelakantan *et al.* [17] has extended this idea and utilized skip-gram model for context clustering. The fusion of skip-gram model and context-clustering was proposed where cluster centroid is equivalent to sense vector and is sent to skip gram model for updating. This approach suffers through expensive training computation cost.

Some research work has also focus on morphology, i.e. sub-words level, to obtain multiple embeddings per word. Unlike previous work, Bojanowski *et al.* [19] explored the internal structure of word and modify skip-gram model with n-grams of word's character. Each character n-gram has a vector representation and their sum generate the word vector. This approach is combined with Gaussian mixture model, where each Gaussian component represents different sense of word [20,21].

The convolutional neural network has explored in many research work that help in producing context relevant embeddings. Jingyun *et al.* [23] has designed a two layer recurrent CNN model to capture context relevant concepts. The first layer presents the (word, concept) pair using pre-trained word vectors (local information) and second layer hidden states are concatenated using Bi-GRU according to the word input time (global information). Both information is aggregated at attention layer and generate context relevant word embedding for classifying short text datasets; TREC, MR (movie review) and AG news corpus. The GCN (Graph Convolutional Network) model variant of CNN has been explored to extract local information using lexical relations in language and global information from BERT model [24-26]. Both the knowledge is combined via attention mechanism through different layers of network. The hybrid of GCN and BERT model (VGCN-BERT) has performed outstanding in text classification on various datasets. SST-2, MR, CoLA, ArangoHate, FountaHate has acquired 91.93%, 86.49%, 83.68%, 88.43% and 81.26% f1-score respectively.

The class-specific vector representation of word [7] in corpus is another popular work for text classifying. The modified version of skip-gram model and continuous bag of word (CBOV) model has been proposed for generating class vectors. The linear compositionality property of word vector has been exploited for adding class information in general word vectors [35-38]. A parallel CNN framework is designed for classifying binary categorized datasets; SemEval 2013 and hurricane related tweets. With proposed features and deep learning model, SemEval 2013 attain 73.15% and hurricane related tweets attain 88.19%. The drawbacks observed in this work (see section 1) has been removed in our

proposed approach. We test our approach on TREC and Kaggle question dataset using deep learning models.

3. FEATURE EXTRACTION METHODS

3.1 Modified Skip Gram Model

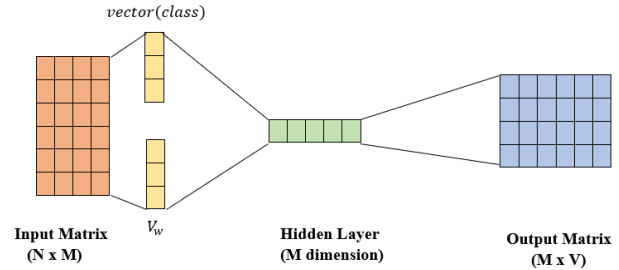


Figure 1 Architecture of Modified Skip-Gram Model. N is size of vocabulary; M is the context window size.

The skip gram model is a shallow neural network architecture to implement word2vec algorithm. It evaluates the context words embeddings from the given target word. The earliest version of model represents a word with single vector, which is not sufficient to address polysemy obstacle in text classification. In the proposed approach, modified skip gram model [7] is used to update context vectors of word using its class-specific embedding, it's architecture is given in Figure 1. The updated objective function for modified skip gram model is given in Equation 1.

$$L = \sum_{w \in C} \log p(V_{w,c}) \quad (1)$$

Where w is the target word and C is corpus.

3.2 Tf-idf-cf

The popular *tf-idf* algorithm is one of the most utilized statistical measure to find relevance of each word in dataset [27,28] Many researchers have continuously improved weighting formula to perform better in various NLP applications. We have used an improved *tf-idf* scoring formula to find relevant terms in each class. Liu *et al.* [29] has introduced in-class characteristic in *tf-idf* (term frequency- inverse document frequency) weighting formula to exploit its ability to differentiate documents among different classes. The in-class feature says that if term frequency is high and present in small portion of documents, then it is qualified to discriminate documents into classes. The formula for *tf-idf-cf* is written in Equation 2:

$$a_{ij} = tf_{ij} * \log \left(\frac{N}{n_j} \right) * \frac{n_{cij}}{N_{ci}} \quad (2)$$

Where tf_{ij} is the term frequency of term j in document i , N is the total count of documents, n_j represents the count of documents where term j appears, n_{cij} is the count of documents in which term j is present in the same class c document i belongs to and N_{ci} is the

number of documents within the same class document i belongs to. The improved formula with smoothing technique is given in Equation 3:

$$a_{ij} = \log (tf_{ij} + 1.0) * \log \left(\frac{N+1.0}{n_j} \right) * \frac{n_{cij}}{N_{ci}} \quad (3)$$

4. PROPOSED APPROACH

In this paper, a methodology for automatic question classification is proposed to handle light text polysemy problem. Using linear compositionality property of word embedding, different meanings of a word are captured in class information vector; one vector for each class [39]. This section describes the procedure of executing this approach, it's flowchart is shown in Figure 2.

(a) Input dataset

The details of question datasets used for this work is given in section 5.2. In TREC dataset, training dataset has questions with their labels and testing dataset has questions only. Whereas Kaggle dataset is split into training and testing dataset with 7:3 ratio and all questions are labelled.

(b) Pre-processing

Text pre-processing is an important step to convert raw textual data into more edible structure for machine learning models. Here the following are steps are taken in sequence to pre-process the questions: Tokenization, Stop Words Removal and Stemming. The output of pre-processing step gives necessary words those are given as input to word2vec algorithm [40-43].

(c) Vector Representation of Words

The question classification accuracy depends upon the quality of vector representation of words. In our proposed approach, improved class-specific word vectors are generated to tackle light polysemy problem in question classification. Figure 3 illustrates the flow of

proposed approach which based upon Sicong work [7]. After pre-processing, words belong to a class are separated and *tf-idf-cf* weight is calculated for each word in the class. Higher the *tf-idf-cf* score signifies the higher occurrence of the word in class, in short that word can represent the class. For generating word embeddings, set of words of class are given as input to modified Skip-Gram model given in Sicong work [7]. Then the *top-n* words with highest *tf-idf-cf* score are picked up and the average of these word vectors (Equation 4) gives class vector (*vector(class)*), approach has been tested for different values of n . So, this methodology generates class vectors equal to the number of classes in dataset.

$$vector(class) = \frac{1}{n} (vector(w_1) + vector(w_2) + \dots + vector(w_n)) \quad (4)$$

The representation of the class specific word embedding $V_{w,c}$ is achieved using linear compositionality property of vector with the summation of general word vector (V_w) and class vector (*vector(class)*) as shown in Equation 5.

$$V_{w,c} = V_w + vector(class) \quad (5)$$

$V_{w,c}$ for all the words of a class are given for training in modified skip gram model to obtain updated word vectors. These vector representations capture semantic as well as conceptual knowledge for each occurrence of word. Basically the number of embeddings per word depends on the number of classes in which word occur. Thus we get multiple class specific word embeddings for training classifiers for multi-class datasets.

(d) Question Classification

The proposed methodology to generate class-specific vector of word for classifying questions is evaluated using deep learning models with existing work of class vectors and others baseline approaches [44-48].

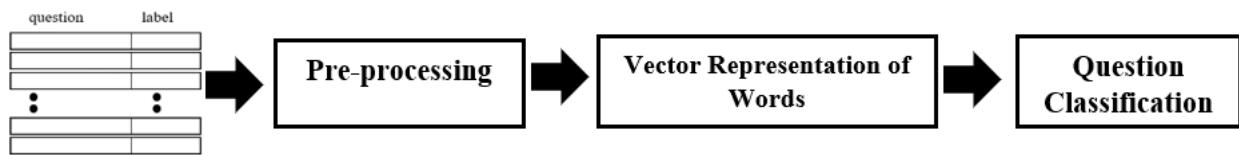


Figure 2 Flowchart of Proposed Methodology

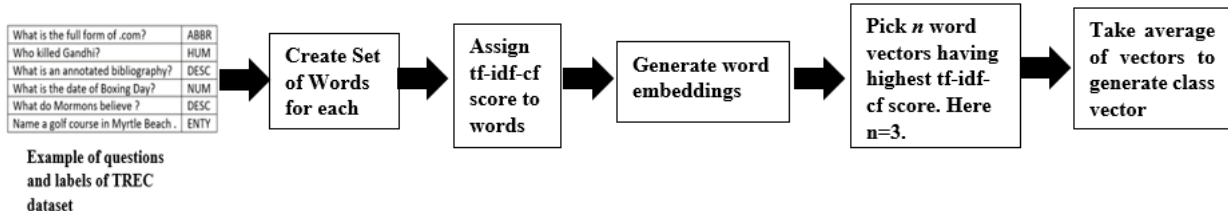


Figure 3 Proposed Approach to generate Class Vector

5. EXPERIMENT AND DATASET

5.1 Experimental Setup

The introduced approach for automatic question classification has been analyzed on system with 8GB main memory and intel core i5 processor and implemented on python 3.6 environment using keras [30]. The word2vec algorithm is implemented using NumPy library with following hyper-parameters values; (*window_size=3, embedding_dimension= 150, epochs=40, learning_rate=0.01*). The vector representation of questions and their labels are feed to classifiers after 10-fold cross validation to calculate accuracy of proposed approach. The confusion matrix has been calculated to find accuracy for classification on testing data with the below given formula (Equation 6):

$$Accuracy = \frac{\text{Correctly labelled questions}}{\text{Total number of questions}} \quad (6)$$

5.2 Datasets

Table 1. Number of Questions in TREC and Kaggle Datasets

Dataset Name	About and Categories	Total Samples	Training Samples	Testing Samples
TREC [32]	Open domain Questions/ 6 Coarse and 50 Fine Classes	6463	5453	1000
Kaggle [33]	Python question collection from stack Overflow/ 13378 classes	342184	239528	102656
Yahoo Questions [34]	Collection of questions posted on Yahoo/ 4 classes	43500	30450	13050

The proposed approach has been evaluated on three multi-class question datasets. The first one is TREC dataset; an open domain dataset which is classified into six main categories and further into 50 fine categories. The second is a Kaggle dataset; is collection of python related questions, which has 342184 questions categorized into 13378 classes. The third dataset is the collection of Yahoo Questions those are categorized on

the basis of expected answers such as yes/no, summary, factoid and list. The classifiers are feed with 70% of questions for training purpose and rest 30% are for testing purpose. The details of training and testing questions is given in table1.

5.3 Baseline Approaches

This section explains the commonly used methodologies used for question classification.

1. Word2vec [13,15]: With Word2vec algorithm, words are represented as a continuous vector. One word has one vector representation.
2. Word2Vec + tf-idf-cf [31]: This methodology has provided weights to word embeddings using tf-idf-cf score [29], which enables to calculate more than one embedding for a word. The higher the tf-idf-cf score of word, more the importance of word in class.
3. Word2vec + class vector [7]: The approach described in [7] has generated class vectors to represent a class, but with few drawbacks (mentioned in section 1). Our work has eliminated these gaps and increases the classification accuracy.

5.4 Results

Table 2. Results on TREC dataset

References	Classifiers		
	CNN [12]	Bi-LSTM [10]	ABBC [13]
Word2vec	82	83.5	85
Word2vec + tf-idf-cf	83.9	85.7	87.7
Word2vec+class vector	86.3	88.6	90.2
Proposed approach	88.2	91.7	93.6

*ABBC- Attention Based Bi-GRU CNN

Table 3. Results on Kaggle dataset

References	Classifiers		
	CNN [12]	Bi-LSTM [10]	ABBC [13]
Word 2vec	81.1	84.4	87.3
Word2vec+tf-idf-cf	83	86	87.3
Word2vec+class vector	85.8	87.4	89.1
Proposed approach	87.4	89.5	91.8

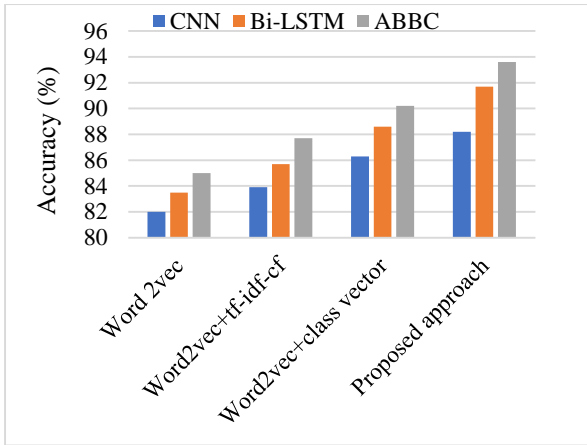


Figure 4 Graphical Representation of Results on TREC dataset.

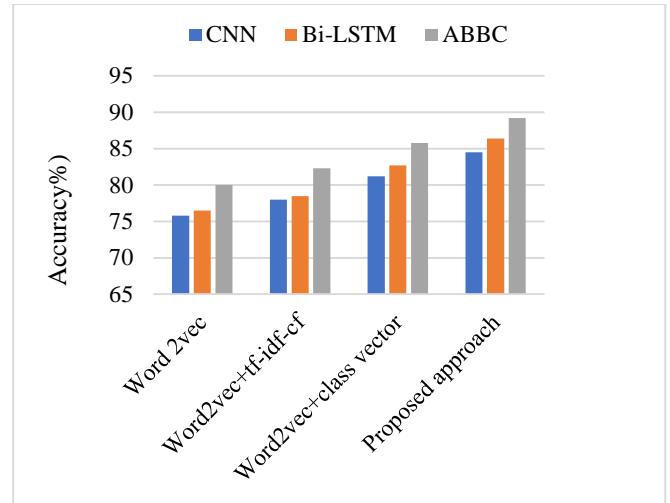


Figure 6 Graphical Representation of Results on Yahoo Questions dataset.

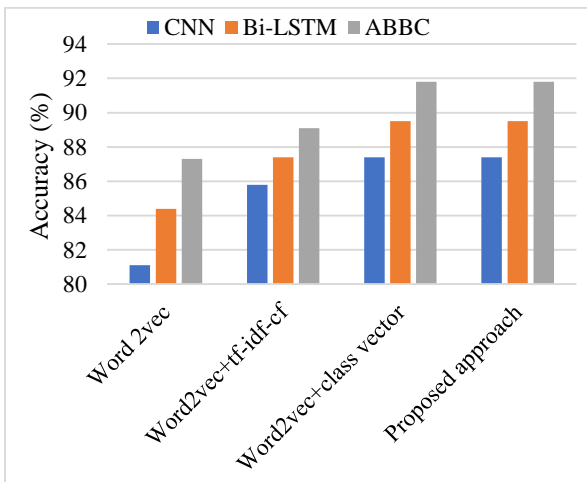


Figure 5 Graphical Representation of Results on Kaggle dataset.

Table 4. Results on Yahoo Questions dataset

References	Classifiers		
	CNN [12]	Bi-LSTM [10]	ABBC [13]
Word 2vec	75.8	76.5	80
Word2vec+tf-idf-cf	78	78.5	82.3
Word2vec+class vector	81.2	82.7	85.8
Proposed approach	84.5	86.4	89.2

6. CONCLUSION

We use CNN, Bi-LSTM and ABBC model with different feature sets to examine the efficiency of proposed approach. As the word embedding is the essential feature for classification, but addition of class information with word vectors has increase its recognition in dataset. The proposed class vector in last section are useful to update context word embedding of a word in class and also expand the distinctive property of a word for each appearance.

The elementary target of our work is to determine the efficacy of question classification task utilizing proposed improved class-specific word embeddings. The graphical representation (figures 4- 6) of results on TREC, Kaggle and Yahoo Questions dataset attained from proposed work shows that proposed approach has achieved competent and better results when compared with baseline approaches.

The proposed approach gives 93.6%, 91.8 and 89.2% classification accuracy on TREC, Kaggle and Yahoo Question datasets respectively on ABBC model, which shows improvement with respect to other baseline approaches (tables 2-4). The comparative analysis has been made with different vector features using deep learning models conclude that ABBC framework is best to extract semantic and contextual information from class-specific vectors to handle light polysemy problem in question classification. When compared with main baseline approach, proposed work has shown ~3% improvement.

The *tf-idf-cf* weight of word has also shown its significance in question classification when seen in second baseline approach and proposed approach. The *word2vec+tf-idf-cf* feature has increase accuracy by ~2.5% for all datasets when compare with *word2vec*

feature. It means *tf-idf-cf* is an efficient method to calculate statistical power of word in every class.

REFERENCES

- [1] T. Georgieva-Trifonova, Text classification based on enriched vector space model, in: B. Rachev, A. Smrikarov (Eds.), Proceedings of the 18th International Conference on Computer Systems and Technologies, 2017, pp. 103-110. DOI: <https://doi.org/10.1145/3134302.3134343>
- [2] Bhuvaneshwary, N., S. Prabu, S. Karthikeyan, R. Kathirvel, and T. Saraswathi. "Low Power Reversible Parallel and Serial Binary Adder/Subtractor." *Further Advances in Internet of Things in Biomedical and Cyber Physical Systems* (2021): 151.
- [3] Prabu, S., Balamurugan Velan, F. V. Jayasudha, P. Visu, and K. Janarthanan. "Mobile technologies for contact tracing and prevention of COVID-19 positive cases: a cross-sectional study." *International Journal of Pervasive Computing and Communications* (2020).
- [4] Pham, Dung V., Giang L. Nguyen, Tu N. Nguyen, Canh V. Pham, and Anh V. Nguyen. "Multi-topic misinformation blocking with budget constraint on online social networks." *IEEE Access* 8 (2020): 78879-78889.
- [5] Naeem, Muhammad Ali, Tu N. Nguyen, Rashid Ali, Korhan Cengiz, Yahui Meng, and Tahir Khurshaid. "Hybrid Cache Management in IoT-based Named Data Networking." *IEEE Internet of Things Journal* (2021).
- [6] M. Pelevina, N. Arefyev, C. Biemann, C., A. Panchenko, Making sense of word embeddings, arXiv preprint arXiv:1708.03390, 2017. DOI: 10.18653/v1/W16-1620.
- [7] S. Kuang, B.D. Davison, Learning class-specific word embeddings, The Journal of Supercomputing, 2020, vol. 76, no. 10, pp. 8265-8292. DOI: 10.1007/s11227-019-03024-z.
- [8] P. Biswas, A. Sharan, R. Kumar, Question Classification using syntactic and rule based approach. in: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2014, pp. 1033-1038. DOI: 10.1109/ICACCI.2014.6968434 .
- [9] H.T. Madabushi, M. Lee, High accuracy rule-based question classification using question syntax and semantics, in: H. Watanabe (Eds.), Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1220-1230.
- [10] S. Yilmaz, S. Toklu, A deep learning analysis on question classification task using Word2vec representations, Neural Computing and Applications, 2020, vol. 32, pp. 1-20. DOI: 10.1007/s00521-020-04725-w
- [11] S. Ruseti, M. Dascalu, A.M. Johnson, R. Balyan, K.J. Kopp, D.S. McNamara, S.A. Crossley, S. Trausan-Matu, Predicting question quality using recurrent neural networks, in: C.P. Rosé, R. Martínez-Maldonado, U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, B. du Boulay, International conference on artificial intelligence in education, 2018, pp. 491-502. Springer. DOI: 10.1007/978-3-319-93843-1_36
- [12] R. Anhar, T.B. Adji, N. Setiawan, Question Classification on Question-Answer System using Bidirectional-LSTM, in: 5th International Conference on Science and Technology (ICST), 2019, vol. 1, pp. 1-5, IEEE. DOI: 10.1109/ICST47872.2019.9166190
- [13] J. Liu, Y. Yang, S. Lv, J. Wang, H. Chen, Attention-based BiGRU-CNN for Chinese question classification. Journal of Ambient Intelligence and Humanized Computing, 2019, pp. 1-12. DOI: 10.1007/s12652-019-01344-9
- [14] M.A. Fauzi, Word2Vec model for sentiment analysis of product reviews in Indonesian language. International Journal of Electrical and Computer Engineering, 2019, vol. 9, pp. 525. DOI:10.11591/ijece.v9i1.pp525-530
- [15] B. Jang, M. Kim, G. Harerimana, S.U. Kang, J.W. Kim, Bi-LSTM model to increase accuracy in text classification: combining Word2vec CNN and attention mechanism. Applied Sciences, 2020, vol. 10, pp. 5841. DOI:10.3390/app10175841
- [16] A. Trask, P. Michalak, J. Liu, sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. arXiv preprint arXiv:1511.06388, 2015.
- [17] A. Neelakantan, J. Shankar, A. Passos, A. McCallum, Efficient non-parametric estimation of multiple embeddings per word in vector space. arXiv preprint arXiv:1504.06654, 2015. DOI: 10.3115/v1/D14-1113
- [18] E.H. Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: H. Li, C.Y. Lin, M. Osborne, G.G. Lee, J.C. Park, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 2012, vol. 1, pp. 873-882.
- [19] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics, 2017, vol. 5, pp. 135-146. DOI: 10.1162/tacl_a_00051

- [20] B. Athiwaratkun, A.G. Wilson, A. Anandkumar, Probabilistic fasttext for multi-sense word embeddings, arXiv preprint arXiv:1806.02901, 2018. DOI: 10.18653/v1/P18-1001
- [21] D.A. Reynolds, Gaussian Mixture Models. Encyclopedia of biometrics, 2009, vol. 741, pp.659-663. DOI: https://doi.org/10.1007/978-0-387-73003-5_196
- [22] X. Yang, C. Macdonald, I. Ounis, Using word embeddings in twitter election classification, Information Retrieval Journal, 2018, vol. 21, pp. 183-207. DOI: 10.1007/s10791-017-9319-5
- [23] J. Xu, Y. Cai, X. Wu, X. Lei, Q. Huang, H.F. Leung, H. F, Q. Li, Incorporating context-relevant concepts into convolutional neural networks for short text classification, Neurocomputing, 2020, vol. 386, pp. 42-53. DOI: <https://doi.org/10.1609/aaai.v33i01.330110067>
- [24] Z. Lu, P. Du, J.Y. Nie, VGCN-BERT: augmenting BERT with graph embedding for text classification, in: J.M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M.J. Silva, F. Martins, European Conference on Information Retrieval, 2020, pp. 369-382. Springer, Cham. DOI: 10.1007/978-3-030-45439-5_25
- [25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, arXiv preprint arXiv:1310.4546, 2013
- [26] J. Lilleberg, Y. Zhu, Y. Zhang, Support vector machines and word2vec for text classification with semantic features, In 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015, pp. 136-140. DOI:10.1109/ICCI-CC.2015.7259377.
- [27] S. Jabri, A. Dahbi, T. Gadi, A. Bassir, Ranking of text documents using TF-IDF weighting and association rules mining, In 2018 4th International Conference on Optimization and Applications (ICOA), 2018, pp. 1-6. DOI: 10.1109/ICOA.2018.8370597
- [28] R.K. Roul, O.R. Devanand, S.K. Sahay, Web document clustering and ranking using tf-idf based apriori approach. arXiv preprint arXiv:1406.5617, 2014
- [29] M. Liu, J. Yang, An improvement of TFIDF weighting in text categorization, in: M. Othman, International proceedings of computer science and information technology, 2012, vol. 47, pp. 44-47
- [30] Deep Learning for Natural Language Processing Using word2vec-keras, <https://towardsdatascience.com/deep-learning-for-natural-language-processing-using-word2vec-keras-d9a240c7bb9d>, last accessed: 2021/05/18
- [31] M. Razzaghnoori, H. Sajedi, I.K. Jazani, Question classification in Persian using word vectors and frequencies, Cognitive Systems Research, 2018, vol. 47, pp.16-27. DOI:10.1016/j.cogsys.2017.07.002
- [32] X. Li, D. Roth, D. Learning question classifiers, In COLING 2002: The 19th International Conference on Computational Linguistics, 2002.
- [33] Python Questions from Stack Overflow, <https://www.kaggle.com/stackoverflow/pythonquestions>, last accessed: 2021/05/18
- [34] Yahoo Answers Dataset, <https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset>, last accessed: 2021/05/19.
- [35] J. Sang, S. Pang, Y. Zha, F. Yang, Design and analysis of a general vector space model for data classification in Internet of Things, EURASIP Journal on Wireless Communications and Networking, 2019, vol. 1, pp: 1-10. DOI:10.1186/s13638-019-1581-3.
- [36] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365, 2018. DOI: 10.18653/v1/N18-1202.
- [37] O. Melamud, J. Goldberger, I. Dagan, context2vec: Learning generic context embedding with bidirectional lstm, in: S. Riezler, Y. Goldberg (Eds.), Proceedings of the 20th SIGNLL conference on computational natural language learning, 2016, pp. 51-61. DOI: 10.18653/v1/K16-1006.
- [38] A. Trask, P. Michalak, J. Liu, sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings, arXiv preprint arXiv:1511.06388, 2015.
- [39] Rajendran, Ganesh B., Uma M. Kumarasamy, Chiara Zarro, Parameshchhari B. Divakarachari, and Silvia L. Ullo. "Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM Classifier on hybrid pre-processing remote-sensing images." *Remote Sensing* 12, no. 24 (2020): 4135.
- [40] Le, Ngoc Tuyen, Jing-Wein Wang, Duc Huy Le, Chih-Chiang Wang, and Tu N. Nguyen. "Fingerprint enhancement based on tensor of wavelet subbands for classification." *IEEE Access* 8 (2020): 6602-6615.
- [41] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi and T. Sato, "Deep-Learning-Empowered Breast Cancer Auxiliary Diagnosis for 5GB Remote E-Health,"

- IEEE Wireless Communications, vol. 28, no. 3, pp. 54-61, June 2021, doi: 10.1109/MWC.001.2000374.
- [42] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A. K. Bashir, F. A. Khan, "Securing Critical Infrastructures: Deep Learning-based Threat Detection in the IIoT", IEEE Communications Magazine, 2021.
- [43] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, T. Sato, "Secure Artificial Intelligence of Things for Implicit Group Recommendations", IEEE Internet of Things Journal, 2021, doi: 10.1109/JIOT.2021.3079574.
- [44] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin and G. Srivastava, "An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things," IEEE Journal of Biomedical and Health Informatics, 2021, doi: 10.1109/JBHI.2021.3075995.
- [45] Subramani, Prabu, Ganesh Babu Rajendran, Jewel Sengupta, Rocío Pérez de Prado, and Parameshachari Bidare Divakarachari. "A block bi-diagonalization-based pre-coding for indoor multiple-input-multiple-output-visible light communication system." *Energies* 13, no. 13 (2020): 3466.
- [46] Rajendrakumar, Shiny, and V. K. Parvati. "Automation of irrigation system through embedded computing technology." In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp. 289-293. 2019.
- [47] K. Yu, L. Tan, X. Shang, J. Huang, G. Srivastava and P. Chatterjee, "Efficient and Privacy-Preserving Medical Research Support Platform Against COVID-19: A Blockchain-Based Approach", IEEE Consumer Electronics Magazine, doi: 10.1109/MCE.2020.3035520.
- [48] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang and K. Yu, "Robust Spammer Detection Using Collaborative Neural Network in Internet of Thing Applications", IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9549-9558, 15 June 2021, doi: 10.1109/JIOT.2020.3003802