International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 6, Nov-Dec-2020, ISSN (Online): 2395-566X

Collation: Features Extraction in Text Classification

Shireen MT, Nahan Rahaman MK
Dept. of Computer Science &Engineering
MGM College of Engineering and Pharmaceutical Sciences
Valanchey, Kerala, India
shireenmt@gmail.com, rahmannahan48@gmail.com

Abstract- **In the last few years, a difference of deep learning models has been applied to natural language processing (NLP) to improve the ability of computers to understand human speech as it is spoken. NLP helps to analyse, understand, and derive meaning from human language in a smart and useful way. For impressive performance of deep learning methods on challenging NLP processing problems we use some text classification features extraction technique.it means that distribution representation for text or words that allows the words with same meaning to have similar representation as real valued vectors in a predefined vector space. In this paper we are comparing the learning technique like Bag of words (BOW), Term frequency-inverse document frequency(tf-idf), word2vec, Glove in word embedding and also explaining the advantage and disadvantage.**

Index words- Deep learning, NLP, Bow, Tf-idf, word2vec, Glov

I.INTRODUCTION

Unstructured text is all over the place, for example, messages, talk discussions, sites, and online media however it's difficult to remove an incentive from this information except if it's coordinated with a specific goal in mind. Doing so used to be a troublesome and costly cycle since it required investing energy and assets to physically sort the information or making carefully assembled decides that are hard to keep up. Text classifiers with NLP have demonstrated to be an extraordinary choice to structure printed information in a quick, practical, and versatile way. Nowadays Deep Learning uses supervised learning to train neural networks using unstructured and unlabelled data set [1]. In deep learning features extraction technique for representing data sets like sentences, text, or words that have the same meaning into converting similar representation of vectors. This approach to speaking to words and reports will be considered one of the key breakthroughs of deep learning on challenging NLP issues.

But machines basically cannot prepare content information in unprotected or unstructured form. They need us to break down the text into a numerical format that is easily readable by the machine that NLP is using. NLP is concerned with how computers can process, analyse, and understand human languages. It makes use of diverse techniques such as statistical methods, ML algorithms, and rule-based approaches [2]. Using these methods, NLP breaks down natural languages into shorter elements, tries to understand the relationships between these pieces, and explores how they fit together to create meaning. For converting text into vectors, we use the concepts of Bag-of-Words (BoW) and TF-IDF to come into play. Both BoW and TF-IDF are techniques that help us convert text sentences into numeric vectors, the most commonly used models for word embeddings are word2vec and GloVe which are both unsupervised approaches based on the words that occur in the same contexts tend to have similar meanings. In this paper, comparing Word embedding and feature learning techniques in natural language processing (NLP) andexplaining the advantage and disadvantage. some techniques are Count Vectorizing, Bag of words, TF-IDF encoding, Word2Vec, and Glove. Before we should understand count vectorizing it is the most basic method for transforming words into vectors is to count the occurrence of each word in each document. Such an approach is called count vectorizing, the main principle of this method is to collect a set of documents they can be words, sentences, and count the occurrence of every word in each document.

Paragraphs or even articles and count the occurrence of every word in each document [3]. The main drawback is it will give a vector with the number of times each word appears in the document. This leads to a few problems mainly that common words like "at", "the", “it" etc will appear most of the time, and less frequency in important words in the document repetitions of a non-important word is higher as compared to another term. Second one is the bag-of-words model it a way of representing text data when modelling text with machine learning algorithms [4]. The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modelling and document classification. Next technique tf-idf it is a statistical measure that assesses how relevant a word is to a report in an assortment of records. In deep learning calculations, as Word2Vec or GloVe are additionally utilized to get better vector representation for words and improve the exactness of classifiers prepared with traditional machine learning algorithm [5].

Applications of text classification has a huge number of utilization cases and is applied to a wide scope of assignments. Now and again, information grouping instruments work in the background to upgrade application highlights we associate with consistently (like email spam separating). In some different cases, classifiers are utilized by advertisers, item administrators, specialists, and sales reps to robotize business cycles and spare many long periods of manual information preparing. This task is shut identified with spam sifting and assessment examination. Aim of this paper is to identify raw data and convert the data into vector form by using features extraction technique. The rest of the paper is organized as follows; InSection II we explain about some review of literature survey paper used the features extraction methods. InSectionIII, Methods and Comparison, In SectionIV ,We conclude with a summary of our methods and Analysis.

II.LITERATURE SURVEY

1 Text Classification Algorithms: A Survey

In this paper, a brief overview of text classification algorithms is discussed. This overview covers different text feature extractions, dimensionality reduction methods, existing algorithms and techniques, and evaluations methods. Finally, the limitations of each technique and their application in real-world problems are discussed. Text classification problems have been widely studied and addressed in many real applications [6-8] over the last few decades. Especially with recent breakthroughs in Natural Language Processing (NLP) and text mining, many researchers are now interested in developing applications that leverage text classification methods.

Most text classification and document categorization systems can be deconstructed into the following four phases: Feature extraction, dimension reductions, classifier selection, and evaluations. In this paper, discuss the structure and technical implementations of text classification systems in terms of the pipeline illustrated. Text Preprocessing Feature extraction and pre-processing are crucial steps for text classification applications, here discussing introduce methods for cleaning text data sets, thus removing implicit noise and allowing for informative featurization. The classification task is one of the most indispensable problems in machine learning. As text and document data sets proliferate, the development and documentation of supervised machine learning algorithms becomes an imperative issue, especially for text classification. Having a better document categorization system for this information requires discerning these algorithms. However, the existing text classification algorithms work more efficiently if we have a better understanding of feature extraction methods.

2. Text Classification

The data and information that are shared over the internet is increasing day by day and it is practically impossible for the human to annotate the text and process it. Information is being shared on the fly, so it is highly necessary to recognize the toxic comments immediately when they are being shared and remove them. Machine Learning and Deep Learning algorithms are capable of identifying the patterns present in data and are capable of solving this problem. Many social platforms don't allow the user to express the opinions as it is difficult for the company to combat any abusive or targeted comments being shared over their platform. This paper deals with understanding the comments shared on social platforms and categorizing them to the possible subgroups as toxicity such as threats, obscenity, insults, and identity-based.

The data is supervised machine learning problem, where the target labels are available for training the algorithm. The data is from kaggle is a data science platform. Data consists of a large number of Wikipedia comments that are labeled by human raters for toxic behavior. The input text may fall into one or more categories with regard to the context. Modeling has been carried out using various text pre-processing techniques and converting text to vectors and applied various machine learning and deep learning techniques. In the end, the models are evaluated on the test data, not used during modeling. Concatenated models are proposed to overcome identify and classify the comments.

This project also uses the Recurrent Neural Network architecture is used to capture the sequential information present in the sentences. The modeling uses the Recurrent Neural Network architectures such as LSTM is long short-term memory and Bi-Directional Recurrent Neural Network and Graded Recurrent Neural Network to perform the classification task. Modeling techniques are fruitful in classifying the multi-label classification challenge of categorizing the Wikipedia comments into 6 categories. This paper proposes a new concatenated approach of combining both sparse represented word vectors with the word embeddings in modeling the classification problem. The sparse matrix is TF-IDF vectorized matrix consists of sentences represented as vectors using the TF-IDF vectorizer and are reduced to lower dimension using the dimensionality reduction technique called as principal component analysis. The sparse matrix of large size is reduced to lower size to be computationally effective. Along with the reduced lower-dimensional features, the concatenated approach uses the word embedding as initial layers of the deep learning architecture. In the intermediate layers, the output of the word embedding network is combined with the lower dimensional TF-IDF vectors to produce a new set of vectors. The concatenated input is combined and fed to the neural network layers and a 6 layered sigmoid activation function is used at the end of the output layer. The reason behind using the sigmoid layer

is at any instance the output may fall into more than one category. The threshold of 0.5 is set, to identify the output values greater than 0.5 are classified as 1 and less than equals to 0.5 are classified as 0. The same threshold level is used for all the 6 categorical labels and outputs are labeled. To overcome the effect of over-fitting in the neural network architecture, dropout layers are used in between the layers.

3. Text Feature Extraction Based on Deep Learning: A Review

Selection of text feature item is a basic and important matter for text mining and information retrieval. Traditional methods of feature extraction require handcrafted features. To hand-design, an effective feature is a lengthy process, but aiming at new applications, deep learning enables to acquire new effective feature representation from training data. As a new feature extraction method, deep learning has made achievements in text mining. The major difference between deep learning and conventional methods is that deep learning automatically learns features from big data, instead of adopting handcrafted features, which mainly depends on prior knowledge of designers and is highly impossible to take the advantage of big data. Deep learning can automatically learn feature representation from big data, including millions of parameters. This paper outlines the common methods used in text feature extraction first, and then expands frequently used deep learning methods in text feature extraction and its applications, and forecasts the application of deep learning in feature extraction.

Text feature extraction that extracts text information is an extraction to represent a text message, it is the basis of a large number of text processing [10]. The basic unit of the feature is called text features [11]. Selecting a set of features from some effective ways to reduce the dimension of feature space, the purpose of this process is called feature extraction [12]. During feature extraction, uncorrelated or superfluous features will be deleted. As a method of data preprocessing of learning algorithm, feature extraction can better improve the accuracy of learning algorithm and shorten the time. Selection from the document part can reflect the information on the content words, and the calculation of weight is called the text feature extraction [12]. Common methods of text feature extraction include filtration, fusion, mapping, and clustering method. Traditional methods of feature extraction require handcrafted features. To hand-design an effective feature is a lengthy process, and deep learning can be aimed at new applications and quickly acquire new effective characteristic representation from training data.

This paper outlines the common methods used in text feature extraction first, and then expands frequently used deep learning methods in text feature extraction and its applications, and forecasts the application of deep learning in feature extraction.

4. Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method

Any mistake in writing of a document will cause the information to be told falsely. These days, most of the document is written with a computer. For that reason, spelling correction is needed to solve any writing mistakes. This design process discusses about the making of spelling correction for document text in Indonesian language with document's text as its input and a .txt file as its output. For the realization, 5 000 news articles have been used as training data. Methods used includes Finite State Automata (FSA), Levenshtein distance, and N-gram. The results of this designing process are shown by perplexity evaluation, correction hit rate and false positive rate. Perplexity with the smallest value is a unigram with value 1.14. On the other hand, the highest percentage of correction hit rate is bigram and trigram with value 71.20 %, but bigram is superior in processing time average which is 01:21.23 min. The false positive rate of unigram, bigram, and trigram has the same percentage which is 4.15 %. Due to the disadvantages at using FSA method, modification is done and produce bigram's correction hit rate as high as 85.44 %.

The designed system is a web-based spelling correction for text documents. The system is designed to allow users to manually and automatically perform the correction. Users can input text documents in the spelling correction system, then the system will display the contents of the document and mark the location of the error word with red colour.

Manual correction can be done by clicking on a word that has a mark and the system will give word suggestions for the error word, where the user can choose the word suggestion that he/she consider as the correct word. Automatic correction can be done only by clicking a button, then manual or automatic correction will result in the output of a text document whose error has been corrected. Spelling correction system is designed using System Development Life Cycle (SDLC). Systems development methods are methods, procedures, job concepts, rules that will be used as guidelines for how and what to do during this development. Method is a systematic way to do things. A step-by-step list of instructions for solving any instance of the problem is known as algorithm [13]. Based on the test results, it can be concluded as follows:

- Text document spelling correction can resolve non-word error, using FSA and Levenstein distance methods. Bigram has greater correction hit rate than unigram and trigram, with the adjoining word error that is 75 %. While the non-adjoining word error, bigram and trigram have the same correction hit rate that is 71.20 %.

- FSA method can be used to shorten the spelling correction processing time.
- The modification results show an increase in correction hit rate from 71.20 % to 85.80 %, but the average processing time becomes longer by 29.57 s.

5. Character-level Convolutional Networks for Text Classification

In this paper explore treating text as a kind of raw signal at character level, and applying temporal (one-dimensional) ConvNets to it. This paper only used a classification task as a way to exemplify ConvNets' ability to understand texts. Historically know that ConvNets usually require large-scale datasets to work, therefore here also build several of them. An extensive set of comparisons is offered with traditional models and other deep learning models. Applying convolutional networks to text classification or natural language processing at large was explored in literature. It has been shown that ConvNets can be directly applied to distributed [14] [15] or discrete [16] embedding of words, without any knowledge on the syntactic or semantic structures of a language. These approaches have been proven to be competitive to traditional models.

There are also related works that use character-level features for language processing. These include using character-level n-grams with linear classifiers [17], and incorporating character-level features to ConvNets [18] [19]. In particular, these ConvNet approaches use words as a basis, in which character-level features extracted at word [18] or word n-gram [19] level form a distributed representation. Improvements for part-of-speech tagging and information retrieval were observed. This article is the first to apply ConvNets only on characters. Here showing that when trained on largescale datasets, deep ConvNets do not require the knowledge of words. This paper offers an empirical study on character-level convolutional networks for text classification. When compared with a large number of traditional and deep learning models using several largescale datasets.

On one hand, analysis shows that character-level ConvNet is an effective method. On the other hand, how well our model performs in comparisons depends on many factors, such as dataset size, whether the texts are curated and choice of alphabet.

6. Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding

Fake news has recently leveraged the power and scale of online social media to effectively spread misinformation which not only erodes the trust of people on traditional presses and journalisms, but also manipulates the opinions and sentiments of the public. Detecting fake news is a daunting challenge due to subtle difference between real and fake news. As a first step of fighting with fake news, this paper characterizes hundreds of popular fake and real news measured by shares, reactions, and comments on Facebook from two perspectives: domain reputations and content understanding. Here domain reputation analysis reveals that the Web sites of the fake and real news publishers exhibit diverse registration behaviours, registration timing, domain rankings, and domain popularity. In addition, fake news tends to disappear from the Web after a certain amount of time. The content characterizations on the fake and real news corpus suggest that simply applying term frequency-inverse document frequency (tf-idf) and Latent Dirichlet Allocation (LDA) topic modelling is inefficient in detecting fake news, while exploring document similarity with the term and word vectors is a very promising direction for predicting fake and real news. To the best of our knowledge, this is the first effort to systematically study domain reputations and content characteristics of fake and real news, which will provide key insights for effectively detecting fake news on social media.

This paper characterizes hundreds of very popular fake and real news from a variety of perspectives including the domains and reputations of the news publishers, as well as the important terms of each news and their word embeddings. Here analysis showing that the fake and real news exhibit substantial differences on the reputations and domain characteristics of the news publishers. On the other hands, the difference on the topics and word embeddings shows little or subtle difference between fake and real news. Future work is centered on exploring the word2vec algorithm [20] , a computationally-efficient predictive model based on neural networks for learning the representations of words in the high-dimensional vector space, to learn word embedding of the important words or terms discovered via the aforementioned tf-idf analysis. Rather than comparing the few important words of each new article, word2vec will allow to compare the entire vector and embeddings of each word for broadly capturing the similarity and dissimilarity of the content.

III.METHODS AND COMPARISON

Rather than depending on physically made principles, machine learning text classification learns how to mention arrangements dependent on past objective facts. By utilizing pre-marked models as preparing information, machine learning algorithms can get familiar with the various relationship between bits of text, and that a specific output (i.e., labels) is normal for a specific input (i.e., text). A "tag" is the pre-decided characterization or class that any given text could fall into. The initial move towards preparing amachine learning NLP classifier is include extraction: a technique is utilized to change every content into a mathematical portrayal as a vector. One of the most oftentimes utilized methodologies is Bag of words, where a vector speaks to the recurrence of a word in a predefined word reference of words.

1. Bag of Words (Bow)

One of the most every now and again utilized feature extraction technique is Bag of words, where a vector speaks to the recurrence of a word in a predefined word reference of words. BOW work when contain text data ie sentimental analysis we can covert to vector. During text pre-processor we should lower the sentence stop keywords like "he", "is"," a", "she"

For example(3.1), consider 3 sentences

Sent 1: He is good boy

Sent 2: She is a good girl

Sent 3: Boys and girls are good

Avoid stop keywords then

Sent 1: good, boy

Sent 2: good, girl

Sent 3: Boys, girls, good

Now apply BOW where convert word to vectors:

Table 1 shows example for BOW

	F1	f2	f3
	Good	Boy	girl
Sent 1	1	1	0
Sent 2	1	0	1
Sent 3	1	1	1

Where F1 ,F2,F3 are independent features Here we can get only the vectors 0 and 1 so in BOW semantic are same and can't identify which word is important have equal weightage. To overcome this, we can use TF-IDF technique.

2.Term-Frequency=Inverse Document Frequency (Tf=Idf)

In TF-IDF (term Frequency inverse document frequency) can manage this difficult better. TF-IDF brings down the weight of regularly utilized words and raises the weight of uncommon words that happen just in current report. TF-IDF equation resembles this:

For example:

Sent 1: He is good boy

Sent 2: She is a good girl

Sent 3: Boys and girls are good

After stemming and lemmatization, avoid stop keywords then

Sent 1: good, boy

Sent 2: good, girl

Sent 3: Boys, girls, good

Term

Frequency=no of repetition of words in a sentence ÷ no of words in sentence (equation 3.2.1)

Consider Table 2 to find out Term frequency

	Sent 1	Sent 2	Sent 3
Good	1/2	1/2	1/3
Boy	1/2	0	1/3
Girl	0	1/2	1/3

Then find out Inverse Document Frequency=log[Noofsentences ÷ Noofsentencescontaingwords](equation :3.2.2)

Let as consider above example.

Table 3 to find out Inverse Document frequency

Words	Inverse Document Frequency (IDF)
Good	log (3/3)
Boy	log (3/2)
Girl	log (3/2)

In this Example we are considering no of sentences so here sent 1 sent 2 sent 3 so total sentences are there 3 sentences and no of repeated words in the sentences consider an example from the sentences boy here boy is repeated 2 time so according to equation (3.2.2) Boy=log (3/2) Finally multiply equation (3.3.2) *equation (3.2.2) ie TF*IDF finally.

Table 4 to find out TF*IDF

	F1	F2	F3
	Good	Boy	Girl
Sent1	1/2*log(3/3)	1/2*log(3/2)	0*log (3/2)
Sent2	1/2*log (3/3)	0*log(3/2)	1/2*log(3/2)
Sent 3	1/3*log (3/3)	1/3*log (3/2)	1/3*log (3/2)

In TF-IDF is more efficient as compare to BOW here semantic are not same identify which word is important.

3. Word 2vec and Glove

Deep learning calculations, as Word2Vec or GloVe are likewise utilized to get better vector portrayals for words and improve the exactness of classifiers prepared withwith traditional machine learning algorithms.

The most ordinarily utilized models for word embeddings are word2vec and GloVe which are both unaided methodologies dependent on the distributional speculation (words that happen in similar settings will in general have comparative implications). Word2Vec word embeddings are vector portrayals of words, that are regularly learnt by an unaided model when taken care of with a lot of text as information (for example Wikipedia, science, news, articles and so on) These portrayals of words catch semantic closeness between words among different properties. Word2Vec word embeddings are scholarly in a such manner, that separation between vectors for words with close implications like (men and women)are closer than distance for words with completely different implication (men and animal)Word2Vec vectors even allow some mathematic operations on vectors. Another word embedding technique is Glove ("Global Vectors"). It depends on matrix factorization methods on the word-context matrix. It first develops a huge matrix of (words x

context) co-event data, for example for each "word" (the lines), you tally how every now and again we see this word in some "specific circumstance" (the sections) in a huge corpus. At that point this matrix is factorized to a lower-dimensional (word x features) matrix, where each line presently stores a vector portrayal for each word. As a rule, this is finished by limiting a "recreation loss". This loss attempts to discover the lower-dimensional portrayals which can clarify the majority of the change in the high-dimensional information.Here similarity of men and women havehigher value as compare to men and animal. Here we can find out the relationship of words and important

Table 5 Feature extraction comparison

Model	Advantage	Disadvantage
Bag of Words (Bow)	Computation is Easy Easy to process the comparability between 2 reports utilizing it Works with obscure word (e.g., New words in dialects)	It doesn't catch position in text (syntactic) It doesn't catch importance in text (semantics) Common words impact on the outcomes (e.g., "am", "is", and so on)
TF-IDF	Easy to process Easy to register the comparability between 2 reports utilizing it Basic measurement to remove the most distinct terms in an archive Common words don't impact the outcomes because ofIDF (e.g., "am", "is", and so on)	It doesn't catch position in text (syntactic) It doesn't catch significance in text (semantics).
Word2Vec	It captures position of the words in the content (syntactic) It catches significance in the words (semantics)	It can't catch the importance of the word from the content (neglects to catch polysemy) It can't catch out-of-jargon words from corpus
GloVe	It catches position of the words in the content (syntactic) It catches importance in the words (semantics) Trained on colossal corpus	It can't catch the significance of the word from the content (neglects to catch polysemy) Memory utilization for capacity It can't catch out-of-jargon words from corpus

IV. CONCLUSION

The text classification is one of the most essential issues in deep learning thathas been applied to natural language processing (NLP) to improve the ability of computers to understand human speech as it is spoken. NLP helps to analyse, understand, and derive meaning from human language in a smart and useful way. In this paper we are comparing the learning technique like Bag of words (BOW), Term frequency-inverse document frequency(tf-idf), word2vec, Glove in word embedding and also explaining the advantage and disadvantage As text classification is application in for example, E-business, news offices, content keepers, websites, catalogs, and likes can utilize computerized advancements to group and label substance and items.

sREFERENCES

[1] https://ikipdia.org/wiki/Deep_learning
[2]. https://en.wikipedia.g/wiki/Natural language processg
[3] https://www.quora.com/Why-TF-IDF-transformation-works-better-than-Count-Vectorizer-in-Machine-Learning
[4] https:// [5] https://monkeylearn.com/text-classification/
[6] Jiang, M.; Liang, Y.; Feng, X.; Fan, X.; Pei, Z.; Xue, Y.; Guan, R. Text classification based on deep belief network and softmax regression. Neural Comput. Appl. 2018, 29, 61–70. [CrossRef]
[7] Kowsari, K.; Brown, D.E.; Heidarysafa, M.; Jafari Meimandi, K.; Gerber, M.S.; Barnes, L.E. HDLTex: Hierarchical Deep Learning for Text Classification. Machine Learning and Applications (ICMLA). In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017.
[8] McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, USA, 26–27 July 1998; Volume 752, pp. 41–48
[9] Liang, H.; Sun, X.; Sun, Y.; Gao, Y. Text feature extraction based on deep learning: A review. EURASIP J. Wirel. Commun. Netw. 2017, 2017, 211. [CrossRef]
[10] V Singh, B Kumar, T Patnaik, Feature extraction techniques for handwritten text in various scripts: a

survey. International Journal of Soft Computing and Engineering 3(1), 238–241 (2013)
[11] Z Wang, X Cui, L Gao, et al., A hybrid model of sentimental entity recognition on mobile social media. Eurasip Journal on Wireless Communications and Networking 2016(1), 253 (2016)
[12] D Trier, AK Jain, T Taxt, Feature extraction methods for character recognition—a survey. Pattern Recogn. 29(4), 641–662 (1996)
[13] B. Miller, D. Ranum. Problem Solving with Algorithms and Data Structures. [Online] from https://www.cs.auckland.ac.nz/courses/compsci105s1c/resources/ProblemSolvingwith AlgorithmsandDataStructures.pdf (2013). [Accessed on 25 January 2017].
[14] C. dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics
[15] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.
[16] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. CoRR, abs/1412.1058, 2014.
[17] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos. Words versus character n-grams for anti-spam filtering. International Journal on Artificial Intelligence Tools, 16(06):1047–1067, 2007
[18] C. D. Santos and B. Zadrozny. Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1818–1826, 2014.
[19] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pages 101–110. ACM, 2014
[20] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, Efficient estimation of word representations in vector space, in Proc. Int. Conf. on Learning Representations, Scottsdale, AZ, USA, 2013.

About Author

Shireen MTreceived the B.Tech degree in Computer Science and Engineering from University of Calicut, Kerala, India, in 2012. Presently, she is doing her Post Graduation in the Department of Computer Science and Engineering, at MGM College of Engineering and Pharmaceutical Sciences, Valanchery, Kerala,India. Her latest research includes Image processing, Natural Language Processing, Deep Learning and Machine Learning

Nahan Rahman MK obtained her Bachelor's degree in Computer Science and Engineering from MEA Engineering College Perinthalmanna, Kerala, India and Master's degree in Computer science and Engineering at Cochin College of Engineering and Technology, Valanchery, Kerala. Currently, she is working as Assistant Professor in the department of Computer Science and EngineeringMGM College of Engineering and Pharmaceutical Sciences, Valanchery, Kerala, India. Her current research includes Artificial Intelligence, Natural language Processing and InformationRetrieval.