IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 19, 2021, accepted December 2, 2021, date of publication December 7, 2021,
date of current version December 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133651

A Deep Learning Model Based on BERT and Sentence Transformer for Semantic Keyphrase Extraction on Big Social Data

R. DEVIKA[1], SUBRAMANIYASWAMY VAIRAVASUNDARAM[1], C. SAKTHI JAY MAHENTHAR[1], VIJAYAKUMAR VARADARAJAN[2], AND KETAN KOTECHA[3]

[1]School of Computing, SASTRA Deemed University, Thanjavur 613401, India
[2]School of Computer Science and Engineering, University of New South Wales, Sydney, Kensington, NSW 2052, Australia
[3]Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Lavale, Pune, Maharashtra 412115, India

Corresponding author: Ketan Kotecha (head@scaai.siu.edu.in)

This work was supported in part by the Science and Engineering Research Board (SERB), Department of Science and Technology, India, through the Mathematical Research Impact Centric Support (MATRICS) Scheme under Grant MTR/2019/000542; in part by SASTRA Deemed University, Thanjavur; and in part by Symbiosis International University.

ABSTRACT In the evolution of the Internet, social media platform like Twitter has permitted the public user to share information such as famous current affairs, events, opinions, news, and experiences. Extracting and analyzing keyphrases in Twitter content is an essential and challenging task. Keyphrases can become precise the main contribution of Twitter content as well as it is a vital issue in vast Natural Language Processing (NLP) application. Extracting keyphrases is not only a time-consuming process but also requires much effort. The current works are on graph-based models or machine learning models. The performance of these models relies on feature extraction or statistical measures. In recent year, the application of deep learning algorithms to Twitter data have more insight due to automatic feature extraction can improve the performance of several tasks. This work aims to extract the keyphrase from Big social data using a sentence transformer with Bidirectional Encoder Representation Transformers (BERT) deep learning model. This BERT representation retains semantic and syntactic connectivity between tweets, enhancing performance in every NLP task on large data sets. It can automatically extract the most typical phrases in the Tweets. The proposed Semkey-BERT model shows that BERT with sentence transformer accuracy of 86% is higher than the other existing models.

INDEX TERMS Attention layer, BERT, deep learning, keyphrase extraction, social data.

I. INTRODUCTION

In recent days, extracting keyphrases from online social data has played a critical role. On big social media with a large amount of unstructured information in Twitter is exponentially increased daily. A human can't prepare keyphrase mining manually. The automatic key phrase automatically generates the most critical documents, allowing people to search more accessible, faster, and more effectively. It helps us to decide whether to proceed with the upcoming search for further information [6]. For different kinds of tasks such as text classification [1], named entity recognition [2], parts of speech [3], [4], sentiment analysis [5], and many other aspects of the text used Deep learning techniques. There is a subtle difference between keyword and keyphrase extraction. A keyword is actually ''a solitary word that is most relevant'' A key phrase is ''one or more words that are observed greatly relevant [7]. A keyword is a unigram, while a key phrase is an N-gram word. For example, 'apple' is a fruit name, whereas 'Apple iPhone' could be the brand name and thus context is crucial.

While extracting keyphrase from a vast corpus is easy, dragging in a short text/sentence is difficult. Major existing works successfully generate keyphrases, but their performance is comparatively less for short sentences [8]. In machine learning approaches namely unsupervised and supervised methods are widely used for keyword extraction. Supervised keyphrase extractions follow the binary classification while unsupervised keyphrase extraction by ranking methods [9], [10]. Supervised methods need linguistic

The associate editor coordinating the review of this manuscript and approving it for publication was Shen Yin.

knowledge, and they implicitly depend on language tools; hence, they extract language-dependent features specific to the training set [11].

Many studies on keyphrase extraction are using supervised learning methods like Naïve Bayes, Support Vector Machine (SVM), and unsupervised methods like Term Frequency - Inverse Document Frequency (TF-IDF) has witnessed a good performance [12]. But these methods depend on feature extraction efforts [13].

Deep learning-based approaches widely provided significant benefits for the task of keyphrase extracting an NLP task. Both CNN and deep CNN, deep RNN, are slow in training due to sequential encoding [14]. Most supervised work is limited to the human-annotated corpus. There are two stages in supervised methods. The first stage is extracting the candidate phrases, making use of the heuristic rule, and in the next step, the train classification model to predict if the candidate phrase is a key phrase or not[15].

The most significant and effective way is to use pre-trained sentence transformers like Generative Pre-trained Transformer (GPT-1) [25], BERT [26], Transformer-XL model pre-trained (XLNet) [27], Robustly Optimized BERT Pre-training Approach (Roberta) [28], Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) [29], Text-to-Text Transfer Transformer (T5) [30], Bidirectional and Auto-Regressive Transformer (BART) [32] to get tweets embedding and then use similarity metric to compute similarity score. Transformer-based pre-trained models have attained remarkable success is mainly due to their ability to learn universal language representation from a massive corpus of unlabeled text data. The downstream task uses this knowledge.

This paper presents an attention-based deep learning model for contextual key phrases with Bidirectional Encoder Representation from Transformer.

The main contributions of this work are fivefold.

- It leverages BERT embeddings to create key phrases that are most similar to Tweets.
- To compute semantic similarity, we use sentence transformers as the embedding model.
- We use three different sentence transformer models to extract keyphrases of the 3-gram range.
- Then, we combine all the keyphrases (obtained from 3 different models) in a single document, apply normalization and threshold values and rank them using the rank aggregation method to get the best keyphrases.
- Our experiments on Twitter data of research papers show that the proposed semkey-BERT work surpasses preceding state-of-the-art approaches.

The rest of the paper is structured in this way. Section 2 provides the related research works. Section 3, set forth the proposed Semkey-BERT model. Section 4 put forward the experimental results of this study, including four datasets. At last, the conclusion is in Section 5.

II. RELATED WORK

Extractive methods select critical words from the original document and collect them to provide a smaller version in an abstractive approach, instead of simply extracting important words, paraphrases, or using more new words to generate a summary. Existing works on keyphrase extraction mainly concentrate on documents in diverse disciplines and events, news, and web text [16]. The supervised method [17] is considered a classification problem for keyphrase extraction on social data. Applying some rules first extract candidate key phrases, and the model trained to predict a keyphrase—feature extraction used for this task with TF-IDF, position, and structural and syntactic features. Unsupervised methods follow pipeline approach such as preprocessing, candidate key phrase generation and then score is calculated for each candidate key phrase. For removing near-duplicates, with some post-processing work selected, the highest keyphrase.

To extract keyphrase on Twitter, Graph-based methods [18] is also widely used in the unsupervised approach. They don't need to be trained on large corpus and don't need any pre-trained rules. It uses statistical features such as degree, Clustering coefficient, Eigenvalues, betweenness, etc. Here text documents are considered nodes, and two nodes are linked together if they have high correlations. SentiWordNet [8] gives the highest sentiment polarity of a specified word. Edge weight is by the number of outgoing edges from the nodes. In-text graph, the real value is added as rank. Then apply statistical equation until the rank value gets converged.

As Scientific information has inadequate human-annotated corpus, existing works on its extraction are bounded. In [19] introduce three categories of the dataset of scientific abstracts. Automatic keyphrase extraction in the Pattern-based bootstrapping approach. But the performance is improved in [20], which uses hand-designed attributes within a supervised bootstrapping structure. SemEval 2017 has three sub works: keyphrase recognition, classification, and the link among extracted keyphrases. To estimate system performance, it uses a science dataset as a benchmark. In [21] includes keyphrase recognition and classification, like the Named Entity Recognition(NER). It follows a sequence labeling problem for keyphrase extraction.

RNN [22] and LSTM-RNN [23] are the most recommended sequence labeling problems. In bidirectional LSTM-RNN with conditional random fields uses NER, which gives a promising performance in keyphrase extraction. In recent years, some pre-trained techniques such as Embeddings from Language Models (ELMO) [35], XLNet45 were used, and it has improved text semantic representation and enhanced versions in various NLP tasks. Word2Vec and Glove algorithms are used to extract syntactic and semantic [33] features of a given text.

III. PROPOSED SEMKEY-BERT MODEL

This research develops and evaluates the deep learning mechanism with three different sentence transformers: xlm-r-

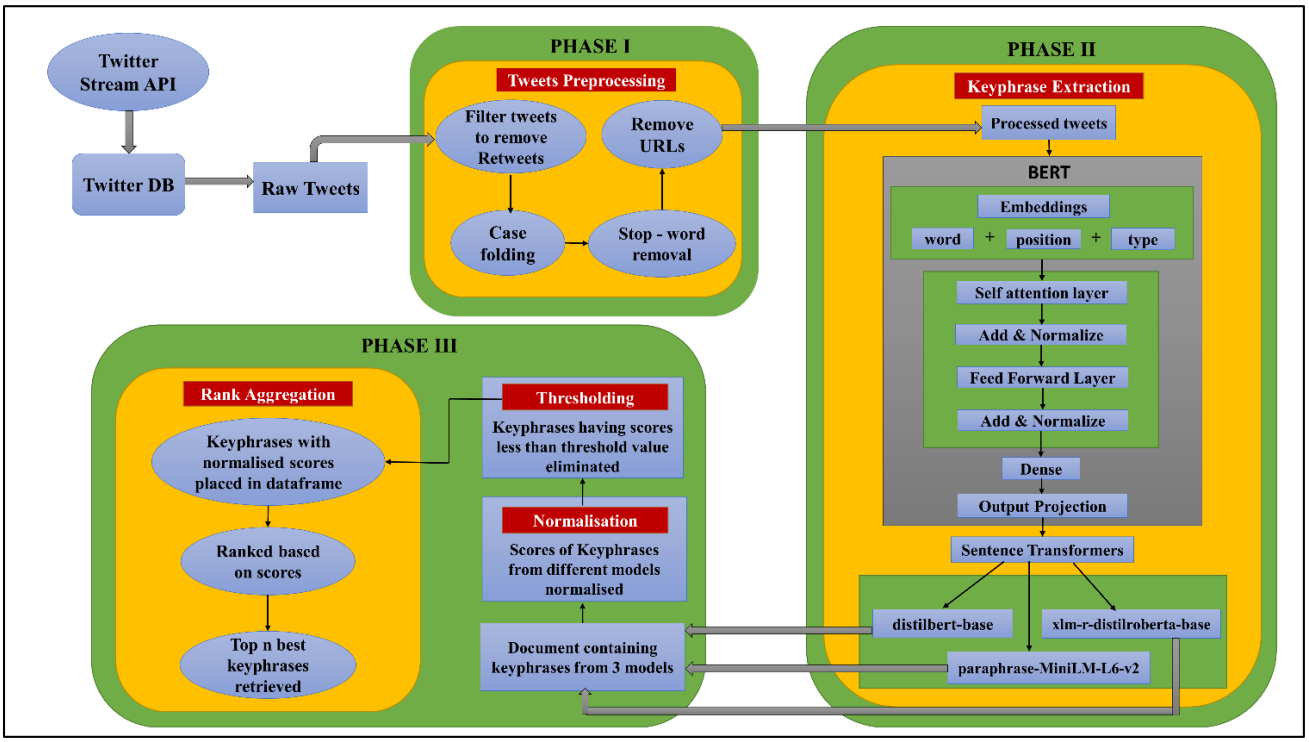


FIGURE 1. Proposed semkey-BERT architecture diagram.

distilroberto, distilbert_base, and paraphrase_minilm_l6_v2, respectively, for keyphrase strategies. This work has leveraged BERT model architecture and a multilayer transformer encoder library with bidirectional self-attention. It addresses self-attention, which can only operate in left or right in the given Tweets. To compute semantic similarity, we use sentence transformers as the embedding model. Finally, we combine all the keyphrases obtained from three different models as a single document, apply normalization and threshold values and rank them using rank aggregation methods to get the top k key phrases as shown in Fig1. The following phases describe how three-sentence transformer models select the candidate keyphrases and ranking methodology for keyphrase extraction.

A. PRELIMINARIES FOR SEMKEY-BERT

Let X be a set of Twitter users. Let $Y = \left\{\{a_{x,n}\}_{n=1}^{N_x}\right\}_{x \in X}$ be a collection of tweets generated by X, where N_x is the overall count of tweets generated by user x and $a_{x,n}$ is the nth tweet of user x.$a_{x,n}$ consists of sequence of words $\left(b_{x,n,1}, b_{x,n,2}, b_{x,n,3}, \ldots\ldots, b_{x,n,M_{x,n}}\right)$ where $M_{x,n}$ is the total count of words in $a_{x,n}$. Let $E = \{e_p\}_{p=1}^{P}$ be the collection of embeddings extracted with BERT for N-gram phrases, where P is the total number of embeddings extracted. We use sentence transformer models that use cosine similarity to find phrases most similar to the document. Let K= $\{k_n, s_n\}_{n=1}^{N}$ be the list of keyphrases, where s_n is the score of each key phrase. Min-Max Normalization normalizes the scores obtained from different models, and a threshold value is applied such that we select only those keyphrases that satisfy the condition $(s_n \geq \sigma)$.

B. PHASE I: PREPROCESSING

For extracting keyphrases, we must process the tweets. First, we collect all the tweets made by the Twitter user for a particular timeframe. The collected tweets, along with the username and TweetId, are stored in a data frame. We removed stop words, URLs, retweets, and case folding to get clean and normalized tweets. We used NLTK (Natural Language Toolkit) python library for preprocessing the tweets, which removes Punctuations, tags, special characters, and digits from the extracted tweets. In preprocessing, it uses the NLP text processors such as tokenization, stemming, and lemmatization.

C. PHASE II: CANDIDATE KEYPHRASE EXTRACTION USING BERT AND SENTENCE TRANSFORMER

The primary task in keyphrase extraction is generating a possible keyphrase candidate. Those candidates come from the given datasets. In NLP applications, the N-gram model is widely used to extract sequences of words, where N denotes the number of words in a row. In this work, we use word-based N-gram models with N=3 in Twitter data analysis. The BERT model uses basic and powerful techniques for extracting keyphrases. It creates embeddings for each word into vectors to get a document-level representation. Then, these embeddings from BERT input to three distinct sentence

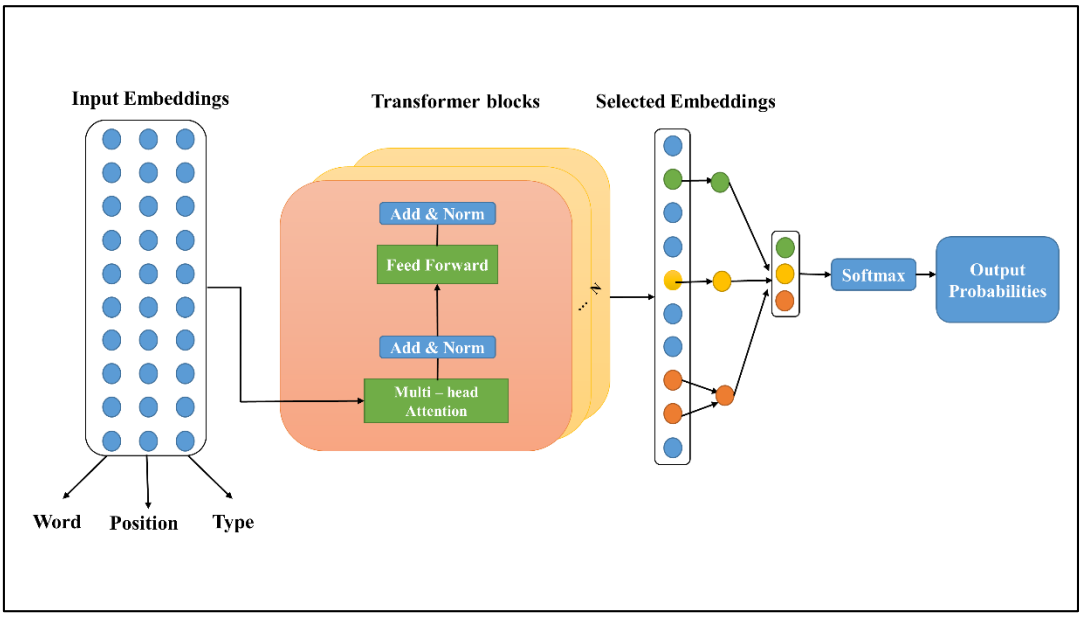


FIGURE 2. Architecture diagram of sentence transformer.

TABLE 1. Hyperparameter values used for sentence transformers.

Hyperparameter	Value
Maximum sequence length	256
Maximum number of training epochs	5
Training batch size	15
Learning rate	5e-5
Early stopping patience	3
Manual seed	4

transformers that use semantic similarity to find phrases most similar to the document. The transformer model consists of stacked layers where each layer contains a multi-head attention layer succeeded by a position-wise feed-forward network, as shown in Fig 2. In [24], the softmax function normalizes the scaled value obtained for each keyphrase. After applying softmax, all the values become positive and add up to 1. The most similar phrases are that best describe the entire tweets.

A vital part of successfully training a good model is to get the hyperparameters right. Most pre-trained Transformer models will converge and give good results if the hyperparameter values are reasonable. Therefore, the values given below are considered for the three-sentence transformer models we used.

D. PHASE III: KEYPHRASE SCORING AND RANKING

The keyphrases fetched from three distinct sentence transformers in Phase II were filtered to discard redundant sub phrases. The key phrases have to be normalized based on their scores as extracted from different models. The goal of normalization is to change the scores of each keyphrase to a standard scale without distorting differences in the ranges of values. We applied Min-max normalization to normalize the scores. Let x be the score of each key phrase. Then the normalized score of x is given as described in equation (1).

$$x' = \frac{x - \text{least}(x)}{\text{most(x)} - \text{least}(x)} \tag{1}$$

where least(x) is the lowest score and most(x) is the highest score of the keyphrases obtained in phase II. Now, we eliminated those keyphrases whose scores are less than a threshold value σ. To fetch only the phrases with high scores to get more defined keyphrases eliminating the insignificant ones. Let k denote keyphrases obtained after normalization. To find the threshold value σ(k), we used the average distance between occurrences of keyphrases d(k) and standard deviation s(k) as described in the formula given below.

$$\text{d(k)} = \frac{(p_1 - p_0) + (p_2 - p_1) + \ldots + (p_{n+1} - p_n)}{n+1}, \tag{2}$$

$$\text{s(k)} = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n}((p_{i+1} - p_i) - d\,(k))}, \tag{3}$$

$$\text{Threshold value } \sigma\text{(k)} = \frac{s(k)}{d(k)} \tag{4}$$

where p_i denotes the score of each keyphrase and n represents the total count of keyphrases obtained. Then, we placed the selected keyphrases in a pandas data frame and used the Rank Aggregation method to rank the key phrases based on their normalized scores. The following Algorithm 1 illustrates the working principle of the Proposed semkey-BERT.

First, we give the dataset as an input. Then we call a fit() function to learn vocabulary from our dataset. Once

Algorithm 1 Keyphrase Extraction and Selection

Input: dataset containing preprocessed tweets
Output: 3 gram range phrases fetched

```
Step 1: Read the dataset;
Step 2: Check for candidates;
        if candidate is None
            call fit() to learn vocabularies;
        end
        else
            set n gram range to 3;
            call CountVectorizer class to transform
            text to vector of token counts;
        end
Step 3: Extract embeddings with BERT;
Step 4: Pass embeddings to sentence transformers;
Step 5: Combine Keyphrases in one document;
Step 6: for each of keyphrase x_i do
            Call Min-Max Normalize(x_i);
            Return normalized score;
        end
Step 7: Eliminate insignificant phrases: using threshold
        value;
Step 8: for each keyphrase do
            Rank by normalized score;
        end
Step 9: Sort the phrases by rank;
Step 10: return top_n keyphrases.
```

when vocabularies are learned, we create an instance of the CountVectorizer class. It transforms corpora of text to a vector of term / token counts. We extract the embeddings using BERT and fetch 3-gram keyphrases using the three-sentence transformer models. Then, the scores of the keyphrases are normalized, insignificant phrases are eliminated using threshold value, and finally ranked based on the normalized score to get the best keyphrases. Table 2 shows the top 10 keyphrases extracted with their normalized score from five real-time datasets.

IV. RESULT AND DISCUSSION

A. DATA SET DESCRIPTION

We carried out the experiments out on the real-time Twitter datasets. Twitter users convey their message in 140 characters and can connect with some topics using the shortened or joined words or word groups called hashtags, which start with the "#" character. In this way, their tweets can be listed in a general search of a particular hashtag[32]. We collected tweets related to the following hashtags Tokyo Olympics 2021, National Education Policy, cybercrime, human rights, and covid-19. To extract tweets from Twitter API, we used the python Tweepy library. To use Twitter API, first, we created a Twitter Developer account. Then after creating an app, we got our API Keys and Access Tokens, which helped us retrieve data from Twitter. We used specific keywords that are related to the topic to fetch the tweets. Every dataset uses a timeframe

TABLE 2. TOP 10 keyphrases from each dataset.

Datasets	Keyphrase	Normalized Score	Rank
Tokyo Olympics	ruleplease treat equally	1.000	1.0
	teambrazil womensvolleyball olympics	0.921	2.0
	olympicgames tokyoolympics womensgymnastics	0.911	3.0
	archery olympicgames usabasketball	0.908	4.0
	tokyoolympics softball olympicgames	0.884	5.0
	cheer4india olympics2020 tokyoolympics	0.872	6.0
	olympics cheer4india tokyoolympics	0.870	7.0
	cheers indian olympic	0.861	8.0
	tokyoolympics olympics2021 bestgameever	0.855	9.0
	olympicgames tennis tokyo2020	0.846	10.0
NEP	transformed education indianeweducationpolicy	1.000	1.0
	educationpolicyhoping better india	0.993	2.0
	india pmnational educationpolicy	0.974	3.0
	india universities need	0.918	4.0
	india neweducationpolicy nep2020	0.910	5.0
	nep2020 transforming india	0.895	6.0
	change indian educationsystem	0.885	7.0
	revolutionise education nep	0.871	8.0
	neweducationpolicy india multidisciplinary	0.865	9.0
	research needed india	0.857	10.0
Cybercrime	cybersecurity ethicalhacking hackers	1.000	1.0
	hackers hacking cyberpunk2077dominos	0.981	2.0
	hackers training hackingtools	0.969	3.0
	cyberattack russian hackers	0.938	4.0
	hackers hacked 5gtechnologyforum	0.927	5.0
	hackers ethicalhacking cyberattacks	0.921	6.0
	solarwinds hackcybersecurity infosec	0.916	7.0
	cyberattacks hacking cybercrimemicrosoft	0.915	8.0
	cybercrime cybersecurity tools	0.902	9.0
	cybercrime dataprotection exclusive	0.901	10.0
Human rights	morality chinaglobal threat	1.000	1.0
	china humanrights defenders	0.955	2.0
	liberate hongkong saveourdemocracy	0.949	3.0

TABLE 2. *(Continued.)* **TOP 10 keyphrases from each dataset.**

	chinese christians persecution	0.932	4.0
	chinesevirus makechina accountable	0.912	5.0
	ccp hongkong political	0.911	6.0
	chinafree assange humanityfirst	0.892	7.0
	need humanrights china	0.887	8.0
	fightfor freedom taiwan	0.774	9.0
	harvesting chinahumanrights violation	0.773	10.0
Covid - 19	vaccine passports covidvaccine	1.000	1.0
	getvaccinated for covid19	0.881	2.0
	covidvaccine vaccinated everybody	0.762	3.0
	daily covid cases	0.760	4.0
	covidvaccine lifesaving protect	0.726	5.0
	bengaluru covidvaccine availability	0.719	6.0
	maharastra fullyvaccinate people	0.711	7.0
	tested covid positive	0.710	8.0
	covid pandemic lockdown	0.703	9.0
	masking coronavirus pandemic	0.691	10.0

TABLE 3. **Datasets used in proposed Semkey-Bert model.**

Dataset	Number of tweets	Size of dataset
Tokyo Olympics	15605	1.46MB
National Education Policy	14973	3.25MB
Cybercrime	12001	2.00MB
Human rights	12001	2.19MB
Covid19	11000	1.11MB

using since and until parameters to extract the tweets. Only the essential information such as the Datetime, Tweet Id, Text, Username kept in a data frame. We validate our proposed semkey-BERT model with the datasets mentioned above of varied topics—Table 3 below shows the number of tweets collected and the size of each dataset.

B. EXPERIMENTAL RESULTS AND COMPARISON

We employed three pre-trained sentence transformer models for extracting N-gram keyphrases from the real-time datasets. First, we used a distilbert-base transformer that consists of 768 hidden sizes, 12 hidden layers, 12 self-attention heads, and 110M parameters. It runs 60% faster and has 40% fewer parameters than Bert-base-uncased as measured on the GLUE language understanding benchmark.

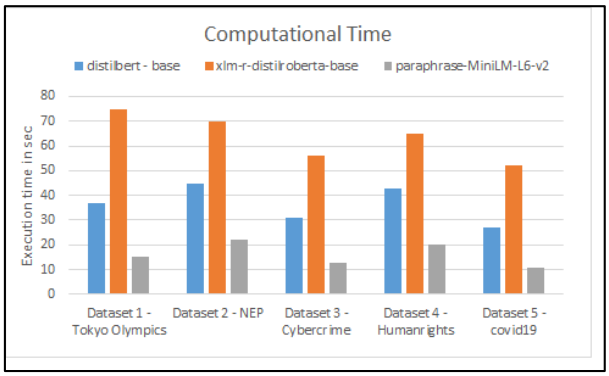


FIGURE 3. **Comparison chart of computational time for three sentence transformer.**

Then we employed an xlm-r-distilroberto-base that maps sentences and paragraphs to a 768-dimensional dense vector space. Lastly, we used the paraphrase-MiniLM-L6-v2 model, which is of size 80MB that has 384 hidden layers. It has a high encoding speed of 14200 sentences per second on a V100 Graphics processing unit (GPU), clearly inferred from Fig 3. The datasets collected were trained using the transformer models mentioned above. Experiments are carried out on google colaboratory notebooks that support GPU and Tensor Processing Unit (TPU) instances, making it a perfect tool for deep learning and data analytics enthusiasts because of computational limitations on local machines. The maximum capacity of RAM is 12.69 GB, and the disk space is 107.72 GB. Google Colab Pro provides even additional memory and disk space. The graph given below shows the computational time of 3 models on the five preprocessed

We used Precision, Recall, F1-score, Accuracy, and Error rate as described in equations 5 to 9 to evaluate the models' performance.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{F1 - Score} = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{7}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$\text{Error rate} = \frac{FP + FN}{TP + FP + TN + FN} \tag{9}$$

where TP, TN, FP, and FN are truly positive, true negative, false positive, and false negative, respectively. The obtained results for each dataset are shown from Table 4 to Table 8.

From the Tables 4-8, we can infer that the average F1-score of all datasets for semkey-BERT is nearly 28 percent higher than other models. Moreover, our proposed model has good Accuracy of more than 75 percent for all the datasets and a low error rate compared to other models. Therefore, the proposed semkey-BERT outperforms the other three-sentence transformer models.

TABLE 4. Dataset–Tokyo olympics.

Model	Precision	Recall	F1 - score	Accuracy	Error rate
distilbert-base	0.720	0.617	0.692	0.707	0.265
xlm-r-distilroberto	0.800	0.500	0.615	0.638	0.352
MiniLM-L6-v2	0.560	0.6428	0.595 8	0.614	0.367
Proposed semkey-BERT	**0.900**	**0.667**	**0.766**	**0.785**	**0.196**

TABLE 5. Dataset–NEP.

Model	Precision	Recall	F1 - score	Accuracy	Error rate
distilbert-base	0.720	0.611	0.661	0.659	0.295
xlm-r-distilroberto	0.760	0.6315	0.6898	0.715	0.277
MiniLM-L6-v2	0.840	0.714	0.77189	0.764	0.192
Proposed semkey-BERT	**0.900**	**0.890**	**0.8949**	**0.902**	**0.071**

TABLE 6. Dataset–cybercrime.

Model	Precision	Recall	F1 - score	Accuracy	Error rate
distilbert-base	0.674	0.587	0.6274	0.651	0.326
xlm-r-distilroberto	0.712	0.607	0.655	0.676	0.287
MiniLM-L6-v2	0.593	0.622	0.606	0.599	0.351
Proposed semkey-BERT	**0.750**	**0.790**	**0.7694**	**0.756**	**0.218**

TABLE 7. Dataset–humanrights.

Model	Precision	Recall	F1 - score	Accuracy	Error rate
distilbert-base	0.776	0.695	0.7333	0.745	0.227
xlm-r-distilroberto	0.544	0.6548	0.5942	0.641	0.351
MiniLM-L6-v2	0.7436	0.713	0.7277	0.709	0.263
Proposed semkey-BERT	**0.850**	**0.838**	**0.8439**	**0.864**	**0.095**

We have already evaluated the proposed semkey-BERT using the F1 score. Now, to assess the quality of predicted keyphrases, F-measure may not be an ideal one. The keyphrases obtained can receive a low score for F1-measure and Accuracy though they are semantically similar to the author assigned phrases. However, the key phrases have to be scored for predicting something close to standard gold phrases. Thus, to evaluate the quality of keyphrases, we go with the Greedy Matching approach.

For every predicted keyphrase, we calculated the cosine similarity scores against each key phrase in the set of author-assigned phrases in the first stage. We then took the highest of all measured cosine-similarity scores for each predicted keyphrase in the next stage, and it must correlate to the top similar keyphrase in author-assigned phrases. Now all the

TABLE 8. Dataset–covid-19.

Model	Precision	Recall	F1 - score	Accuracy	Error rate
distilbert-base	0.560	0.5714	0.5656	0.592	0.369
xlm-r-distilroberto	0.600	0.667	0.6317	0.657	0.313
MiniLM-L6-v2	0.640	0.625	0.6324	0.6216	0.372
Proposed semkey-BERT	**0.700**	**0.714**	**0.8757**	**0.864**	**0.126**

TABLE 9. GM(p,au) values of each model for five datasets.

Model	Tokyo olympics	NEP	cybercrime	humanrights	covid-19
distilbert-base	0.756	0.789	0.656	0.644	0.613
xlm-r-distilroberto	0.414	0.539	0.577	0.371	0.342
MiniLM-L6-v2	0.799	0.747	0.645	0.542	0.657
Proposed semkey-BERT	**0.927**	**0.813**	**0.691**	**0.774**	**0.785**

TABLE 10. GM(au,p) values of each model for five datasets.

Model	Tokyo olympics	NEP	cybercrime	humanrights	covid-19
distilbert-base	0.511	0.638	0.583	0.577	0.605
xlm-r-distilroberto	0.632	0.714	0.525	0.623	0.592
MiniLM-L6-v2	0.587	0.696	0.6585	0.715	0.476
Proposed semkey-BERT	**0.645**	**0.733**	**0.731**	**0.915**	**0.764**

predicted keyphrases scores are averaged to get a final score.

$$\text{GM(p,au)} = \frac{\sum_{k=1}^{m} score(p_k, au)}{m} \tag{10}$$

where m is the total count of predicted keyphrases, GM is the scoring function of the Greedy Matching approach, p and au are the collection of predicted and author-assigned keyphrases, respectively. Table 9 below gives the score assigned by GM(p,au) to the proposed semkey-BERT and the sentence transformer models for all datasets.

But, the above scoring is asymmetric. The reason is that the value obtained from GM(p,au) need not be the same as GM(au,p). Thus, for asymmetric scoring, we used the following formula [23].

$$\text{S} = \text{GM(p,au)} + \text{GM(au,p)} \tag{11}$$

$$\text{S\_GM(p,au)} = \frac{S}{2} \tag{12}$$

There is one more advantage of symmetric scoring. If we use GM(p,au) alone, the model will receive a maximum score. However, only one keyphrase in the predicted collection matches perfectly with just one author assigned keyphrase, among various other gold keyphrases which weren't predicted.

Likewise, suppose GM(au,p) is used alone. In that case, it gives a high score to the model, although only one author

TABLE 11. S_GM(p,au) values of each model for five datasets.

Model	Tokyo olympics	NEP	cybercrime	humanrights	covid-19
distilbert-base	0.6335	0.7135	0.6195	0.6105	0.609
xlm-r-distilroberto	0.523	0.6455	0.551	0.497	0.467
MiniLM-L6-v2	0.693	0.7215	0.672	0.6285	0.5665
Proposed semkey-BERT	**0.786**	**0.773**	**0.711**	**0.8445**	**0.7745**

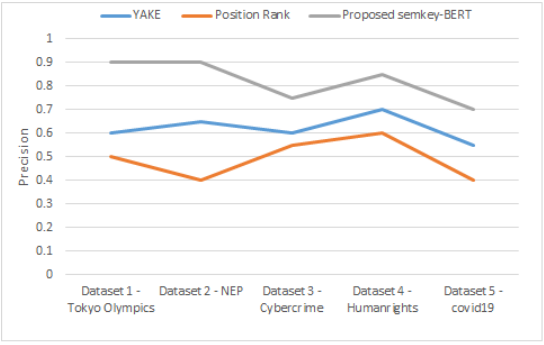


FIGURE 4. Comparison of different models using precision.

assigned keyphrase that matches remarkably with hardly one predicted phrase, amidst several other expected key phrases, which do not match with gold. These drawbacks are resolved using S_GM(p,au). Table 10 and Table 11 shown below give the score assigned by GM(au,p) and S_GM(p,au) to the proposed semkey-BERT and the sentence transformer models for all datasets, respectively.

From Table 11, we infer that semkey-BERT gets a good score for all the datasets for the Greedy Matching approach compared to other models. So, this implies that the keyphrases obtained from the proposed semkey-BERT are highly semantic similar to the ground truth keyphrases.

Now, we compare the proposed semkey-BERT with the unsupervised models for keyphrase extraction. Statistical models like the TfIdf, KPMiner, Yet Another Keyword Extractor (YAKE), Rapid Automatic Keyword Extraction (RAKE), and Graph-based models [34] like TextRank, SingleRank, TopicRank, TopicalPageRank, PositionRank, MultipartiteRank are some of the unsupervised models.

Figure 4 above shows the variation in the Precision for the proposed semkey-BERT against other unsupervised models. The plot implies that the average Precision of all datasets for semkey-BERT is nearly 30 percent higher than other models.

Figure 5 shows the variation in the Recall for the proposed semkey-BERT against other unsupervised models. From the plot, it is evident that average Recall of all datasets for semkey-BERT is almost 20 percent higher than other models.

Figure 6 shows the variation in the F1-score for the proposed semkey-BERT against other unsupervised models. The result implies that the average F1-score of all datasets for semkey-BERT is nearly 30 percent higher than other models.

Figure 7 shows the variation in the Accuracy for the proposed semkey-BERT against other unsupervised models.

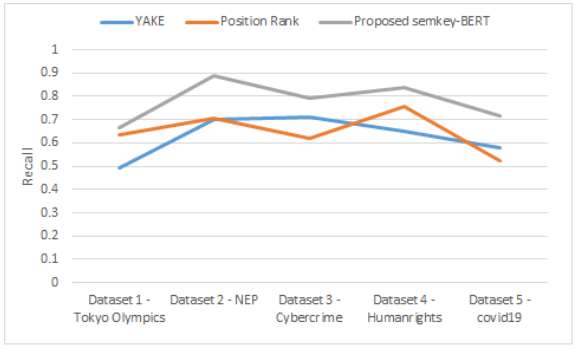


FIGURE 5. Comparison of different models using recall.

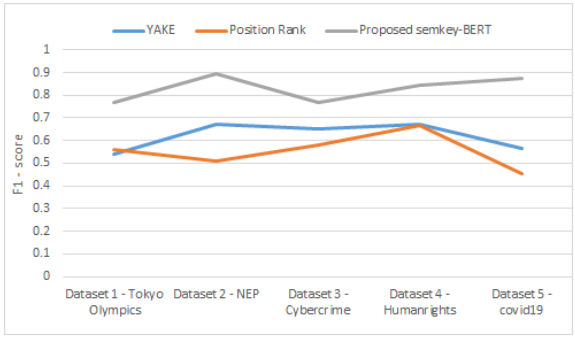


FIGURE 6. Comparison of different models using F1-score.

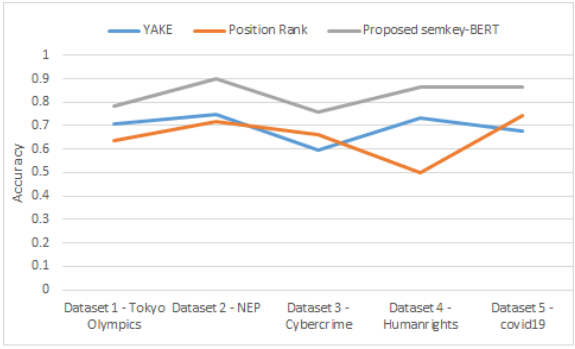


FIGURE 7. Comparison of different models using accuracy.

From the results, it is evident that average Accuracy of all datasets for semkey-BERT is almost 20 percent higher than other models.

We can infer that PositionRank and YAKE models give many redundant keyphrases. The key phrases are not so defined, i.e., they have more repeated words in their phrases. For large datasets, these models did not produce good results. Moreover, we understand that YAKE and PositionRank models didn't use semantic similarity approach to fetch keyphrases. But semkey-BERT had well-defined and diverse phrases even for large datasets. Thus, the proposed semkey-BERT model outperforms other unsupervised models.

V. CONCLUSION

In this work, we represented a deep learning model for extracting keyphrases from five different Twitter big social data datasets. We have carried out the state-of-the-art BERT with a three-sentence transformer model for

keyphrase extraction. After extracting candidate keyphrases, we employed rank aggregation methods to extract the key phrases with top scores. The resultant keyphrase is unique, and we used min-max normalization and threshold to increase the diversity in the selected keyphrases. The proposed semkey-BERT model yield outperforms on five Twitter datasets. From this time forth, we aim to expand our work by considering the N-gram range. We believe, in the future, to identify the optimal BERT-based fine-tuning deep learning model for keyphrase extraction to improve our keyphrase.

VI. DECLARATION OF COMPETING INTEREST

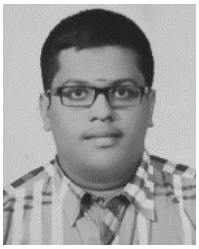
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*.
[2] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 160–167.
[3] M. F. Kabir, K. Abdullah-Al-Mamun, and M. N. Huda, "Deep learning-based parts of speech tagger for Bengali," in *Proc. 5th Int. Conf. Inform., Electron. Vis. (ICIEV)*, May 2016, pp. 26–29.
[4] J. Li, R. Li, and E. Hovy, "Recursive deep models for discourse parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 2061–2069.
[5] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, Jul. 2018, Art. no. e1253.
[6] Z. A. Merrouni, B. Frikh, and B. Ouhbi, "Automatic keyphrase extraction: A survey and trends," *J. Intell. Inf. Syst.*, vol. 54, pp. 391–424, May 2019.
[7] K. M. Hammouda, D. N. Matute, and M. S. Kamel, "Corephrase: Keyphrase extraction for document clustering," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, Jul. 2005, pp. 265–274.
[8] P. Sharma and Y. Li, "Self-supervised contextual keyword and keyphrase retrieval with self-labelling," Preprints, 2019, doi: 10.20944/preprints201908.0073.v1.
[9] A. Bougouin, F. Boudin, and B. Daille, "Topic rank: Graph-based topic ranking for keyphrase extraction," in *Proc. Int. Joint Conf. Natural Lang. Process. (IJCNLP)*, Oct. 2013, pp. 543–551.
[10] F. Bulgarov and C. Caragea, "A comparison of supervised keyphrase extraction models," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 13–14.
[11] S. Yang, B. Ju, and H. Chung, "IJDC| Peer-reviewed paper," *Int. J. Digit. Curation*, vol. 14, no. 1, pp. 62–87, 2019.
[12] K. S. Hasan and V. Ng, "Automatic keyphrase extraction: A survey of the state of the art," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Jun. 2014, pp. 1262–1273.
[13] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Automatic keyphrase extraction from scientific articles," *Lang. Resour. Eval.*, vol. 47, no. 3, pp. 723–742, Sep. 2013.
[14] Y. Zhang, M. J. Er, R. Zhao, and M. Pratama, "Multiview convolutional neural networks for multidocument extractive summarization," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3230–3242, Oct. 2017.
[15] O. Medelyan, E. Frank, and I. H. Witten, "Human-competitive tagging using automatic keyphrase extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Aug. 2009, pp. 1318–1327.
[16] X. Wan and J. Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *Proc. AAAI*, vol. 8, Jul. 2008, pp. 855–860.
[17] W. T. Yih, J. Goodman, and V. R. Carvalho, "Finding advertising keywords on web pages," in *Proc. 15th Int. Conf. World Wide Web*, May 2006, pp. 213–222.
[18] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Jul. 2004, pp. 404–411.
[19] S. Gupta and C. D. Manning, "Analyzing the dynamics of research by extracting key aspects of scientific papers," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, Nov. 2019, pp. 1–9.
[20] C. T. Tsai, G. Kundu, and D. Roth, "Concept-based analysis of scientific literature," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2013, pp. 1733–1738.
[21] X. Zhu, C. Lyu, D. Ji, H. Liao, and F. Li, "Deep neural model with self-training for scientific keyphrase extraction," *PLoS ONE*, vol. 15, no. 5, May 2020, Art. no. e0232547.
[22] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNN's-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1064–1074.
[23] V. Rus and M. Lintean, "A comparison of greedy and optimal assessment of natural language Student input using word-to-word similarity metrics," in *Proc. 7th Workshop Building Educ. Appl. Using NLP*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 157–162.
[24] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3 pp. 1–40, Apr. 2021.
[25] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Blog, 2018.
[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[27] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5753–5763.
[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[29] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–18.
[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, *arXiv:1910.10683*.
[31] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[32] İ. Türker and E. E. Sulak, "A multilayer network analysis of hashtags in Twitter via co-occurrence and semantic links," *Int. J. Mod. Phys. B*, vol. 32, no. 4, 2018, Art. no. 1850029.
[33] H. Liu, L. Wang, P. Zhao, and X. Wu, "Document specific supervised keyphrase extraction with strong semantic relations," *IEEE Access*, vol. 7, pp. 167507–167520, 2019, doi: 10.1109/ACCESS.2019.2948891.
[34] R. B. Asrori, R. Setyawan, and M. Muljono, "Performance analysis graph-based keyphrase extraction in Indonesia scientific paper," in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*, Sep. 2020, pp. 185–190, doi: 10.1109/iSemantic50169.2020.9234231.
[35] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang, "SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model," *IEEE Access*, vol. 8, pp. 10896–10906, 2020, doi: 10.1109/ACCESS.2020.2965087.

R. DEVIKA received the M.Sc. degree in computer science from Bharathidasan University, India, the M.Phil. degree in computer science from Alagappa University, and the M.Tech. degree in computer science and engineering from SASTRA Deemed University, Thanjavur, India, where she is currently pursuing the Ph.D. degree. She is currently working as an Assistant Professor with the School of Computing, SASTRA Deemed University. She has 12 years of experience in teaching. Her research interests include text analysis, big data analytics, and social networks analysis.

SUBRAMANIYASWAMY VAIRAVASUNDARAM received the B.E. degree in computer science and engineering from Bharathidasan University, India, the M.Tech. degree in information technology from Sathyabama University, India, and the Ph.D. degree from Anna University, India. He is currently working as a Professor with SASTRA Deemed University, Thanjavur, India. He has 17 years of experience in academia. He is continuing the extension work with the support of the Department of Science and Technology as a Young Scientist Award Holder. He has been contributing papers and chapters for many high-quality technology journals and books that are being edited by internationally acclaimed professors and professionals. He is a Research Supervisor and a Visiting Expert to various universities in India. His technical competencies lie in recommender systems, social networks, the Internet of Things, information security, and big data analytics. He is on the reviewer board of several international journals and a member of the program committee for several international/national conferences and workshops. He also serves as a guest editor for various special issues of reputed international journals.



C. SAKTHI JAY MAHENTHAR received the B.A. degree in hindi from DBHPS, in 2014. He is currently pursuing the Bachelor of Technology degree in the field of computer science and engineering with SASTRA Deemed University, Thanjavur. His research interests include big data analysis on social media and natural language processing.

VIJAYAKUMAR VARADARAJAN is currently working as an Adjunct Professor with the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He has more than 18 years of experience which includes ten years in teaching and eight years in the industry. He is also a Coordinator with the Cloud Computing Research Group and Internship in India and Worldwide. In VIT, he involved in many research and development activities, where he has also organized many national/international seminars/workshops/symposiums/conferences/special sessions in the area of cloud computing and big data, which includes ISBCC'14, ISBCC'15, ISBCC'16, ISBCC'16, ICBCC'18 in India, CCCA'14 in Vietnam, and CCNC'14 in USA. His research interests include grid computing, cloud computing, big data, web semantics and also involved in the domain like biomedical application, mammogram, autism, immune systems and other areas like key management, security issues in cloud, and grid computing.

KETAN KOTECHA is currently an Administrator and a Teacher of deep learning. He has expertise and experience in cutting-edge research and projects in AI and deep learning for the last 25 years. He has published more than 100 widely in several excellent peer-reviewed journals on various topics ranging from cutting edge AI, education policies, teaching-learning practices, and AI for all. He has published three patents and delivered keynote speeches at various national and international forums, including at the Machine Intelligence Laboratory, USA, IIT Bombay, under the World Bank Project, the International Indian Science Festival Organized by the Department of Science and Technology, Government of India, and many more. His research interests include artificial intelligence, computer algorithms, machine learning, and deep learning. He was a recipient of the two SPARC projects worth INR 166 lakhs from the MHRD Government of India in AI in collaboration with Arizona State University, USA, and The University of Queensland, Australia. He was also a recipient of numerous prestigious awards, like the Erasmus+ Faculty Mobility Grant to Poland, the DUO-India Professors Fellowship for research in responsible AI in collaboration with Brunel University, U.K., the LEAP Grant at Cambridge University, U.K., the UKIERI Grant with Aston University, U.K., and a Grant from the Royal Academy of Engineering, U.K., under Newton Bhabha Fund. He is an Associate Editor of IEEE Access.