International Journal of Applied Science and Engineering

Comprehensive analysis of deep learning based text classification models and applications

Niraj Kumar*, Subhash Chandra Yadav

Department of Computer Science and Engineering, Central University of Jharkhand, Ranchi, India

ABSTRACT

The social media platform has become one of the prime modes of interaction between different natures of peoples, where they used to share their feelings in the form of textual messages. Due to the easy availability of plenty of social media tools like Twitter, Flickr, Imgur, Facebook etc. more and more people are indulging themselves in propagating enormous amounts of information on diverse nature of topics/various issues, and that has become a huge source of data to be analyzed by the researchers to extract useful information. This research article comprises a brief study of different text classification models, which uses deep learning algorithm in Natural Language Processing task. However, it remains a challenging issue for most of the researchers to get absolute architecture, layout and appropriate techniques for classifying text data. Further, the study reveals a brief discussion on the relevance of various deep learning models available for text classification along with their feature assessment also a comparative study of the various available deep-learning models have also been done during the work.

Keywords: Text classification, Neural network, Attention mechanism, Transformer, RNN, Deep learning.

OPEN ACCESS

Received: December 9, 2022
Revised: January 5, 2023
Accepted: March 8, 2023

Corresponding Author:
Niraj Kumar
niraj.kumar@cuj.ac.in

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394(Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

A text classification task is a process of assigning a text message or document to its pre-defined appropriate category. The defined categories depend on the type of dataset in which it is applied and can be different for related topics. Text classification is a widely used task in the processing of natural language. Its goal is to automatically classify a document, text messages, and blogs into one or more pre-defined classes or categories. As an example, an online news portal that posted a news article can be assigned the categories of the posted article as sports, technology, society, environment etc. as mathematically represented in Fig. 1.

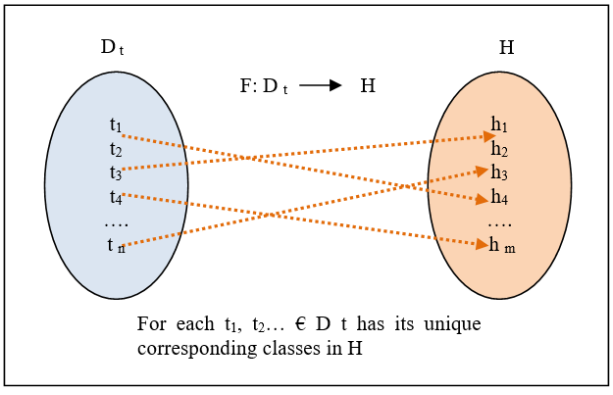

Fig. 1. Mathematical representation of the assignment for the class

In other word suppose a document set $D_t = t_1, t_2....., t_n$ called as training set, in this document t_i is labeled in h_j which belongs to the set of headings $H = h_1, h_2....., h_m$ classification model determines by classifying any document t_k under appropriate category from the set H. The issue is to discover a linear function mapping from D_t to H. Text-based data can be classified in two ways. In the first approach, a few rules can be made where a collection of words may be grouped; which decide the sentiment of input text. By doing this texts can be categorized into their specific classes. This approach can be used for a handful of data in the case when data is very large this process is neither efficient nor cost-effective.

Whereas, in the second approach for large data sets, it is better to utilize the natural language processing technique (Fahad and Yahya, 2018) and texts data can be classified with machine learning / deep learning architecture. The fundamental component for machine-based text classification is the embedding model, which maps text in a feature vector of low dimension. And after giving input to the classifier we get output as to which category text belongs. Classification of text data can be used for spam detection of e-mail, targeting customer needs, news summarizing etc... In this era where text data is generated every second of the day, which becomes an asset for any organization and business need.

The remaining part of this paper is organized in six parts. Section 2 describes text classification methodologies in more detail. Section 3 reveals various approaches for text classification with architecture. The application area of text classification is detailed in Section 4. Classification evaluation methods are discussed in Section 5. The conclusion is presented in Section 6.

1.1 Deep Learning

Deep learning approaches apply deep neural networks, and become very popular because of their high performance in solving the complex task of Artificial Intelligence (AI). Deep learning accomplishes higher power with flexibility due to the ability to process huge amounts of a feature in unstructured data.

Osindero et al. (2006) introduced a Deep Neural Network (DNN), the concept was based on a neural network, it works on different layers stacked one after another, and during the learning phase, it updates weights calculated by optimization function. Hinton et al. (2006) named the layer of a neural network a neuron. Because of the deep learning capability, it trained network models properly, have resulted in great achievement in many of the challenges in regression and classification problems. Due to its learning capability, there is a great demand in the area of data science and AI. Deep learning not only learns mapping in the middle of the input layers and outputs layers but also structures the underlying data as input vectors Karhunen et al. (2015). In nutshell, DL is a special variant of machine learning (ML) that accomplishes great power and acquiescence by learning new representation of the world in the encapsulated hierarchy of concept and representations, and accompanied by concept describe in relation to simpler concepts. Different elements of the Deep learning model are depicted in Fig. 2.

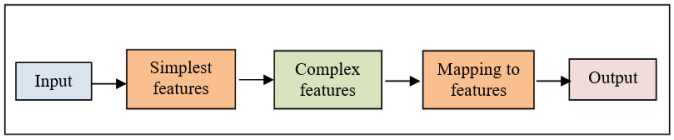

Fig. 2. Elements of deep learning model

1.2 Related Work

Several deep learning-based models have been developed for classifying text data and Natural Language Processing (NLP) related tasks. However, many approaches have applied to develop a good architecture for classifying text data. This section presents a review of some of the previously developed models. Joulin et al. (2016) proposed a very efficient text classifier called FastText. Treated text containers of words use n-gram features for capturing words and other information. Le et al. (2014) introduced doc2vec, to learn feature descriptions of text in a fixed-length piece like paragraphs, sentences and documents by use of an unsupervised algorithm. Tai et al. (2015) invented a model based on Tree-LSTM structure, where concept of LSTM was utilized as a tree-like network structure. Liu et al. (2016) implemented three different architectures based on Recurrent Neural Networks (RNNs) integrated with a multi-learning framework for text data classification. Johnson et al. (2016) examine embedding method for text region using LSTM. Kalchbrenner et al. (2014) proposed the first CNN structured model. Model uses the dynamic max- pooling technique (K-max) called DCNN (dynamic CNN). Chen (2015) invented a simple CNN structure model for classification of Text data with comparison to DCNN. Zhang et al. (2015) propose Character-based CNN for text classification first time. Yang et al. (2016) suggested a hierarchical attention network for text classification. Zhou et al. (2016) developed sentiment classification models of different language by using the LSTM network. Santos et al. (2018) suggested the Attentive Pooling (AP) two-way attention mechanism for pair wise ranking. Kowsari et al. (2017) perform text classification by integrating multiple deep learning methods stacked in a hierarchical manner named as HDL-Tex.

1.3 Motivation

Introductory neural networks came with perceptions at neural layers and computations were limited. In the second generation, the propagation error rate was calculated and the back propagation of error was considered to make learning easier. After that, other neural networks evolved gradually (Wang and Raj, 2017; Miikkulainen et al., 2019). The main strength of deep learning is to handle the complex data and also establish efficient relations among them; which is then used to classify both labeled and unlabeled

data. It analyses the data, correlates and merges the data for quick learning. Furthermore, a deep learning framework can perform the execution of thousands of data in a fraction of the time if properly trained. Apart from this deep learning can perform feature engineering on its own, which is not available with machine learning.

In case of limited data availability, the deployed model cannot train effectively which affect the performance and leads to inaccurate prediction; because deep learning architecture learns systematically, and enormous volumes of the collection of data are essentially required for training the model. In deep learning based models, it is hard to understand and fix the process of decision making as it permit only to view input and predicted output without emphasizing steps involved in data processing.

1.4 Emergence of Deep Learning from Machine Learning

Automated machine learning methods were very popular over the last few years. Many Automated machine learning founded models classify text data in a two-step plan of action (Bijaksana and Algarni, 2013). In the beginning, some domain knowledge based features were picked from the text data. In the other step, the same features are given to the classifier as an input to process the machine and come up with a predicted output. Some of the Renault knowledge based features are BoW (Bag of Words) and their supplements (Yang and Pedersen, 1997). Some of the popular algorithms of classification are the Bayes model, SVM, Probability tree model, and Multi probability tree model (Joachims, 2005; Chen et al. 2009; Haddoud, 2016). The automated two-step architecture classifier has many limitations. Believing in Knowledge based feature extraction and obtaining good results require great subject knowledge and designing accurate features for the classifier. Through this process, it is very difficult to generalise the new classification task. Finally, these models did not come forward when the training data is on large scale, due to the difficulty in determining features. To address all these limitations Deep Learning approaches have been traverse. The basic object of this architecture is embedding methods, which draw text data in a low dimensional sparse vector, where knowledge based features are not required. Fig. 3 visualizes the procedures of text classification in relation with machine learning and deep learning.

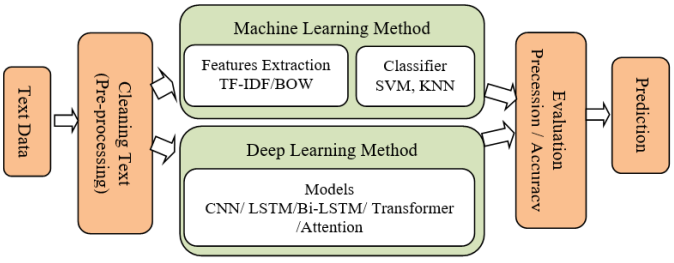

Fig. 3. Text classification from ML to DL

2. TEXT CLASSIFICATION METHODOLOGIES

Automatic text classification includes numerous machine learning algorithms, NLP and other AI-enabled techniques for the classification of text data, which is more faster and accurate than manual process and also cost-effective. This paper, discuss automatic text classification using a deep learning algorithm. Automatic Text Classification (TC) comes under the category of supervised machine learning task in which process of learning is done by utilizing a collection of classified text data. The text classification steps are shown in Fig. 4.

This technique provides one or additional pre-defined class labels to the subject. Supervised classification architecture utilizes a collection of pre-defined category of text data to process the learning of the model. The important and main steps involved in the process of text classification are data document collection, pre-processing of text data, model training and predicting output from the model.

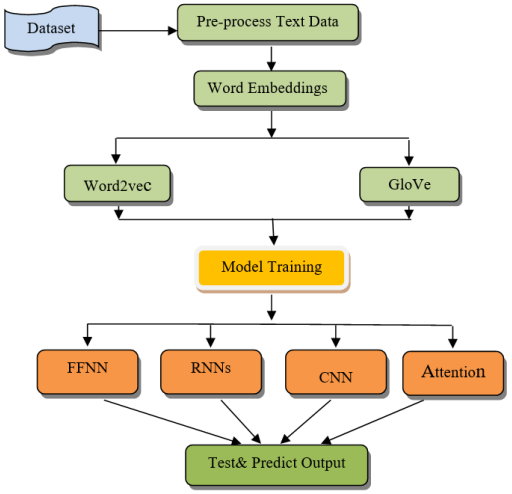

Fig. 4. Text classification steps of deep learning model

2.1 Dataset for Text Classification

Data is the main component of the AI model and the sole of the popularity of any machine learning or deep learning model that we ear-witness today. It is essential to collect data in the form of a dataset. Table 1 describes some of the broadly used datasets for the various text classification and related task.

2.2 Pre-processing of Textual Data

Pre-process and feature extraction are the important steps during text classification of textual data. Generally, in a text document-based dataset carry needless words, misspell words, slang words etc. Deep learning model performance is adversely affected by these noisy words have no significance in the classification process. This section will explain methods and techniques for pre-processing text data before providing it to the model.

https://doi.org/10.6703/IJASE.202309_20(3).0023

Table 1. Dataset for text classification

Dataset name	Size	Application area
Yelp-5 (Kaggle, 2021a)	Training samples-650,000 test samples-50,000	Sentiment analysis
IMDB Dataset (Kaggle, 2021b)	Training sets-25,000 and test set-25,000	Sentiment analysis
AG News Dataset (Zhang et al., 2015)	Training samples-120,000 test samples-7,600	News classification
20 Newsgroups (Mikolov et al., 2013)	Samples-18,821	News classification
Dbpedia dataset (Lehmann et al., 2015)	Training samples-560,000 test samples-70,000	Topic classification
EUR-Lex (Loza and Fürnkranz, 2008)	Documents-19,314 categories-3,956	Topic classification
SQuAD 1.1 (Rajpurkar et al., 2018)	Question-answer 107,785 pair	Question-answer
WikiQA (Yang et al., 2015)	Samples-3,047 questions answer	Question-answer

- Tokenization: Tokenization is the pre-processing process to break a full-length text into symbols; phrases; words; and other meaningful parts of the textual data that is usually known as token (Gupta and Malhotra, 2015).
- Stop Words: Text data includes numerous words have no significance in the text classification algorithm for example (‘is’, ‘am’, ‘are’, ‘about’, ‘after’). The general way to deal with all these words is to apply the method to remove these from text data (Saif et al., 2014).
- Capitalization: While creating a text or document capitalization is used in diverse ways. The way to overcome this is to convert every word into a small case letter to make identical feature space (Gupta and Lehal, 2009).
- Slang word and Abbreviation: Slang words and abbreviations are supplemental words used in a text document are important to handle. An abbreviation used is a short form of the word, which mostly contain word-starting letter such as NN stand for Neural Network. This can be resolved by converting it into formal language (Dhuliawala et al., 2016).
- Noise Removing: For human understanding, many of the text document contains unnecessary characters like comma, full stop, semicolon, question mark, and punctuation which has no use of machine so they can be removed before feeding to the classification algorithm (Pahwa et al., 2018).
- Spelling Correction: Miss Spell is a very common problem in text data it is an optional task in pre-processing step. Many methods and algorithms are available to address this issue in NLP (Mawardi et al., 2018).
- Stemming: In a text document, one word can be reappear in its different grammatical forms, as an example, maybe in singular or plural but have the same semantic meaning. NLP provide a method to convert a different form of a word in the same space called stemming; Sampson (2005) for example “paying” is stem to “play”.
- Lemmatization: The process to convert a word to its dictionary meaning or removing the suffix of a word called lemmatization. It provides basic form of the word known as lemma (Sampson, 2005).

2.3 Word Embedding Technique for Text Classification

The word embedding technique is a feature learning process of mapping every word and phrase from vocabulary to an n-dimension vector. Different word embedding techniques developed to convert uni-grams into DL model understandable input. Here we will discuss some most important word embedding methods such as Fast-Text, Glove and Word2Vec, which has been use in the DL model for better output.

2.3.1 Word2Vec

An improved word representation technique Word2Vec proposed by (Mikolov et al., 2013) the architecture uses two shallow effective models Continuous- Bag-of-Words and Skip Gram model which is used for learning vector space re-presentation of the words or phrases by considering the co-occurrence frequency and consequently similar meaning; there are two related vectors within embedding vector-space. First, one is known as Continuous-Bag of Words Model that represent given target words in multiple words. Such as the word, “study” and “books” are context words of “Education”. It uses distributed continuous representation in context words (Mikolov et al., 2013). This model used mostly to represent an unordered huge collection of words in the form of a vector. First, it creates a vocabulary of all the unique words of a corpus. The output will be the predicted word given in its context. The setting of the window size will determine the number of words used. Whereas the second one is Continuous Skip Gram Model that represents a similar architecture like CBOW (Mikolov et al., 2013) model. The skip-gram model attempt to maximize the classification of a particular word based on the other distinct words in the sentence, instead of foretelling the present word built on its background. Both CBOW and Skip-Gram models preserve the syntactic and semantic knowledge of the sentence used in the Deep learning algorithm.

2.3.2 Global Vectors to Represent Word

In the hierarchy of embedding techniques, another dynamic word embedding method used in the text data classification is name as GloVe (Pennington et al., 2014). The processing of words is almost alike to the Word2vec method, here every words are represented in high-dimension vectors after that they are trained built on nearby words with a huge corpus. The pretrained embedding of words are used to accomplish many other tasks based on 400000 words of vocabulary, which was

trained with Wikipedia 2014 as a corpus. The GloVe provides the pre-trained vectorization of words with 50, 100, 200, 300 dimensions trained over a very big corpus, which also includes the content of Twitter data.

2.3.3 FastText to Represent Word

Most of the word embedding models disregards the words morphology by assigning each of the word in a distinct vector (Bojanowski et al., 2017). A novel technique discovered by the Face book research lab and come up with the issue of the morphology of words by innovating a new embedding method known as FastText. Face book has released pre-trained word embedding vectors that is trained with Wikipedia using FastText of 300 dimensions in 294 different languages. Whereas the Skip-Gram model with default settings is used by FastText (Bojanowski et al., 2017).

3. TEXT CLASSIFICATION APPROACHES

Based on the internal structures, text classification models can be divided into various categories mainly.

- Feed-Forward Neural Network (FFNN)
- Recurrent Neural Network (RNN) based neural network
- Convolutional Neural Network (CNN) based neural network
- Neural Network with Attention Mechanism
- Hybrid deep learning approach combines RNN's, CNN, Attention mechanism and others
- Transformer Model

3.1 Feed-Forward Neural Network

Feed-Forward Neural Networks consider the text as a collection of a word. It is one of the first and very useful algorithms also known as deep network or multi-layer perceptron or neural network. In this architecture, data passes through the mess artificial network in one or more layers. Each layer filters the outline, identifies similar entities and throws the output to the next layer.

Fig. 5 shows the architecture of Feed-Forward Neural Network (FFNN) with the multi-layer along with one hidden layer. In this architecture, one hidden neuron is placed between the output layer and the input layer. The functionality of a hidden neuron is to interpose between the output and external input networks in a meaningful style. Network enables to extracts high-order statistics when one or more than one hidden layers are present in the network. For example the Fig. 5 mentioned above depict one hidden layer and the complete network is said to be of 8-6-4-network, where network comprises of eight input neurons, six hidden neurons and four output neurons.

The advantage associated with feed forward neural networks is the potential to acquire knowledge and to established relations to represent text in a non-linear relationship.

It learns from the given input with their relationship and can model new relationships on new unseen data. However, FFNN is hardware dependent that requires expensive processing units for the execution of the task. It has also observed that solutions provided by FFNN are of black box-type and after generating the prediction, it is very difficult to find how it has been generated. Application area of FFNN is described in Table 2.

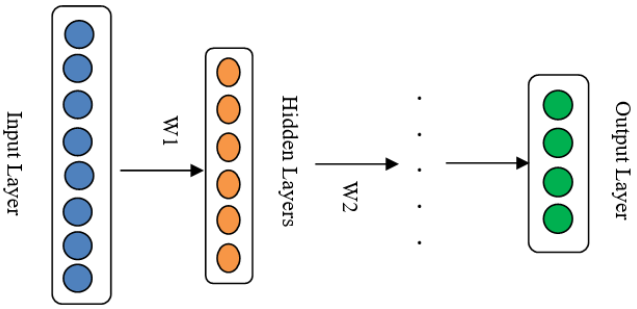

Fig. 5. Architecture of FFNN

Table 2. Applications areas of FFNN

Application area	Architecture	References
Text classification, Sentiment Analysis	Vector representation technique used: Word2vec, GloVe, Classifier used: Naïve Bayes, SVM, logistic regression	Chen et al. (2013); Mikolov et al. (2013); Le et al. (2014); Joulin et al. (2018)

3.2 RNN Based Neural Network

RNN-founded model views text data as a sequence of a word and captures word sequence dependency and its structures. Fig. 6 depicts the basic architecture of an RNN. The working procedures of the RNN model are completed in three stages. In the first stage, the neuron moves forward across the hidden layer to make a prediction.

In the next second stage, it uses the loss function to compare the prediction with its true value. The loss function determines how the model is performing. The model is better when the loss function value is lower. At the final stage, the gradient of each node is determined by using the error value in back propagation. The value of the gradient used to restructure the weight of the Neural Network at each node. RNNs used for the analysis of sequential data. The reason behind this is that models built on layers, which provides the model with short-term memory. By utilizing this memory, the model predicts the next data to process more accurately. For the previous information, which is not more important will be kept in memory or not depends on the allotted weight to it. RNN includes LSTM and all its variants. Table 3 lists the application area of RNN.

RNN transmits the information acquired previously in the later state, which means the present output is a sequence that is related to the previous output. RNN connect the semantics of the context in text processing and makes text

classification more accurate. However, RNNs faces the problem of long-term memory dependency, which is then resolved by introducing LSTM (Long Short Terms Memory).

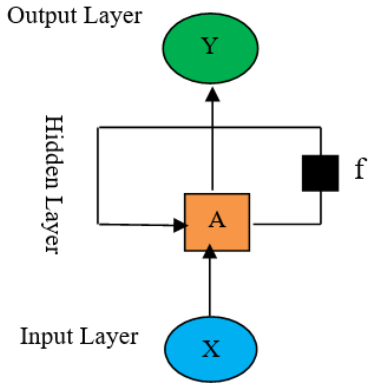

Fig. 6. Basic RNN architecture

Table 3. Application areas of RNN

Application area	Architecture	References
Sentiment classification, Emotion analysis, Sentences comparison, Question classification, Subjectivity classification and Newsgroup classification	Vector representation technique: Word2vec, Classifier used: LSTM, Bi-LSTM	Tai et al. (2015); Cheng et al. (2016); Zhou et al. (2016); Mahto and Yadav (2022)

3.3 CNN Based Neural Network

CNN based Text classification model trained to identify patterns in textual data and identify key phrases for classification. The functionality of CNN is similar to the visual peridium of the animal brain. CNN gives promising results in a text classification task. It is very much similar to the image classification task the difference is that in image classification it takes pixel value whereas in text classification word vectors matrix is used. CNN Model functionality completed in three layers. Here, the very first is embedding layer, where words are converted in embedding vectors. Then the entire processing take place at convolution layer that is the second layer. Pre-defined filters pass through the sentence matrix and convert it to the matrix of lower dimension. The block diagram of CNN is given in Fig. 7.

The third layer called the soft-max layer here sentence matrix is reduces to its lower dimension by the down sampling technique. The function embedding lookup is for converting word embedding from the sentence. The defined filter used to reduce the word matrix and capture convolved features. These capture features are further reduced and the generated output is passed through the max pooling layer and there is a further down sample to predict the output. The CNN application areas are listed in Table 4.

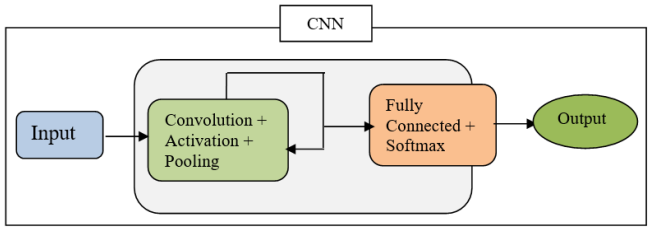

Fig. 7. Block diagram of CNN architecture

Table 4. Application areas of CNN

Application area	Architecture	References
Sentence Modelling, Character-based classification, Text modelling	Embedding technique used: Word2Vec, Classifier used: CNN, DCNN	Kalchbrenner et al. (2014); Zhang et al. (2015); Prusa et al. (2016)

The main advantages of CNN are that it extracts important content from the sequence of text, assigns weight and shares it in the network. It minimizes computational cost in comparison to systematic neural networks. It works well with image processing which is different from text processing. In text classification, many words have different meanings depending on the context of the text. CNN cannot transfer ample information in the context of the text.

3.4 Neural Network with Attention Mechanism

The attention Technique in the model identifies and correlates words in text data. Now a day it has become a useful technique in the development of DL models. The attention mechanism is influenced by the functioning of the human biological system which when working with a large volume of information focuses on the important part of the information. Neural Network architecture with attention mechanism is broadly used in different NLP problems to learn a sequence of data or information for a long time.

Fig. 8 represents the seq2seq model, which is integrated by an attention mechanism. In RNN, the encoder undertakes the input sequence in the form of a context vector. To create a mapping between the context vector and the entire input we can introduce an attention mechanism where the weights of the mapping connection are changeable for each output. Due to the mapping connection between the context vector and input sequence, the context vector has access to all the input sequence this resolve the problem of forgetting long sequence information. By using the attention mechanism in a neural network context vectors have the following information- Hidden state of Encoder; Hidden state of Decoder; Alignment between input and target. Table 5 displays the application areas of the neural network with the attention mechanism.

The attention mechanism has great advantages in computing the NLP tasks by recognizing the information from the input layer. The attention mechanism utilizes all the intermediate phases of the encoder to compute the

context vectors for decoding and predicting the output without throwing the intermediate phases. On the other hand, the disadvantage of the attention mechanism is, it increases the computation.

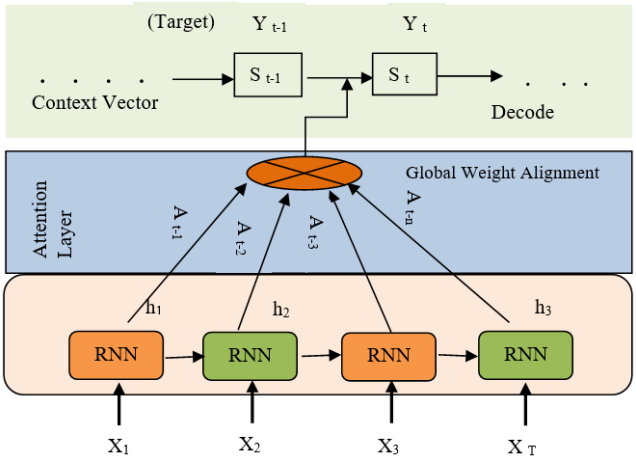

Fig. 8. Seq2seq model integration with attention mechanisms

Table 5. Application areas of attention mechanism integrated neural network

Application area	Architecture	References
Text Classification, Sentiment analysis, Question-Answering	Embedding technique used: Word2Vec, GloVe, Classifier used: Visual attention Mechanism	Liu et al. (2016); Yang et al. (2016); Zhou et al. (2016); Wang et al. (2018)

3.5 Hybrid Deep Learning Approach

Hybrid Deep Learning approach combines different variants of RNN, CNN and attention mechanisms to express global and local features of input sentences or documents. Fig. 9 displays a model using a hybrid deep learning approach.

It has shown excellent concurrence in terms of its achievement and happening over a large area of application with a different data type. The model combines two or more types of variants to grasp the strength of all the variants. In other words, it is the approach to incorporate the advantage of two or more different models in one by combining all; therefore, the hybrid model can address the potential pitfall of one single model if any. The integrated model's effectiveness may vary according to the task to done. For example the aim of combining LSTM, CNN and SVM to take advantage of two deep neural network architecture by using SVM algorithms when performing emotion analysis of social media data of the different domain. Application areas of the model using hybrid deep learning are shown in Table 6.

Hybrid model utilizes the strength of two or more deep learning methods for solving complex tasks. However, due to expensive network and its very complex design, it requires more effort to install the methods with additional hardware requirement to develop the models.

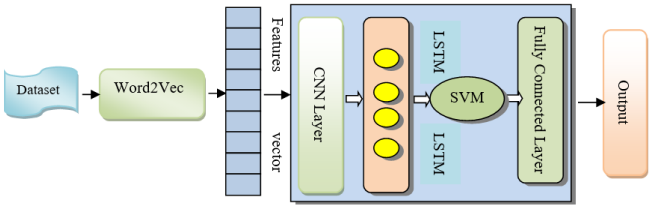

Fig. 9. Hybrid deep learning approach

Table 6. Application areas of hybrid deep learning approach

Application area	Architecture	References
Sentiment classification, Question classification, Document modelling, Language modelling	Word embedding method used: Word2Vec, GloVe, FastText, Classifier Used: LSTM, CNN, Bi-lstm etc.	Tang at el. (2015); Zhou et al. (2016); Kim et al. (2016); Luštrek et al. (2016)

3.6 Transformer Based Network

Computational bottlenecks endured by different variants of RNNs are sequentially processing text. In comparison to RNN, the processing of text data is not sequential in CNN. However, the computational cost for RNN and CNN to learn the relationship between the words increases as sentence length increases. Fig. 10 illustrates the transformer model's block diagram.

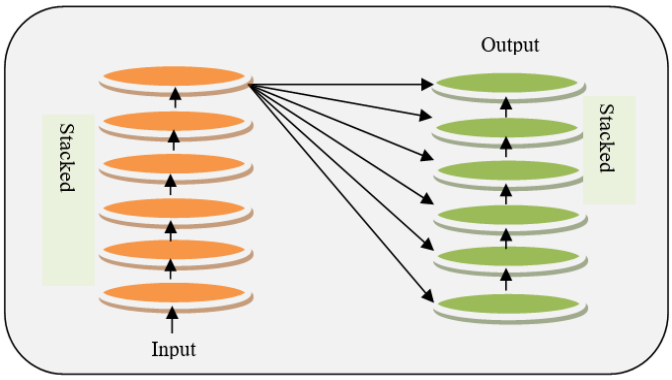

Fig. 10. Block diagram of transformer model

To conquer the constraint of RNN and CNN; transformer architecture based self-attention mechanism is introduced; which provides parallel computation for each word of a sentence.

Transformer Network makes it possible to pre-train big language models with the use of GPU and provides more uniformity than CNN or any variant of RNN- based models to achieve high accuracy. Attention mechanism based transformer model with encoding and decoding architecture of RNN to perform sequence-modeling tasks. Like RNN

transformers do not process data sequentially which allows transformers more parallelization and hence reduces the training time. The architecture of the transformer consists of two components: A stack of encoder and Stack of a decoder.

The main advantage of the transformer is that, it pays equal attention during the processing of all the elements in the sequence. Transformer processes plenty of data very fast in very less time. Transformer restricts complete utilization of sequential input. Where, the representation of the input in the given layer may only access its representation from its lower layer, instead of its higher level. Application regions of the Transformer network are shown in Table 7.

Table 7. Application areas of Transformer network

Application area	Architecture	References
Machine translation, Question answering, Summarization, Reading comprehension	Transformer-based PLMs (Pre-trained dialect model) uses stacked encoder and decoder	Vaswani et al. (2017); Radford et al. (2019)

4. APPLICATION AREAS OF TEXT CLASSIFICATION

Many natural language tasks has been effectively classified by applying Deep learning based text classifier that takes input as pair of text to process and predict output.

- Sentiment Analysis: To analyze the opinion of a user in written text (e.g. Tweet, blogs, reviews and feedback) by extracting their feelings and viewpoint can be in the form of binary class or multi-class problems.
- Topic Classification: Aims to predict the subject or content of large text data into defined categories with accuracy for strong relevant information retrieval. (e.g. Product review and customer support system, Product use fullness)
- News Classification: The content of news is a very useful source of information. A news categorization system assists the end user to obtain information of its interest in good time e.g. identifying news of concern in real-time, flash news based on user interest.
- Question-Answering: Question-Answering deep learning model can frame answers by some given context or without context. It can exactly phrase answers from paragraphs.

5. RESULT WITH DISCUSSION

Under mentioned segment comprises of discussion about the performance metrics used for appraising the deep learning model for text classification and present the performance analysis of deep learning-based text classification models.

Accuracy / Error Rate – It is the fundamental matrix to assess the quality of the classification model. Consider TPR, FPR, TNR and FNR indicate a true positive result, a false positive result, a true negative result, and a false negative result, respectively. The accuracy of the model and error rate is given as Equations (1) and (2).

$$\text{Accuracy} = \frac{\text{TPR}+\text{TNR}}{N(\text{Total Sample})} \quad (1)$$

$$\text{Error Rate} = \frac{\text{FPR}+\text{FNR}}{N(\text{Total Sample})} \quad (2)$$

Precision, Recall and F1-Score - This matrix is very useful and used more than the accuracy-error rate matrix for test classification these days to access the models. For binary classification task, precision and recall defined by the Equations (3) and (4). The F1 score is calculated by taking the Harmonic Mean (HM) of Precision (Pre) and Recall (Rec) represented in the Equation (5).

$$\text{Precision (Pre)} = \frac{\text{TPR}}{\text{TPR}+\text{FPR}} \quad (3)$$

$$\text{Recall (Rec)} = \frac{\text{TPR}}{\text{TPR}+\text{FNR}} \quad (4)$$

$$\text{F1 -Score} = \frac{2*\text{Pre}*\text{Rec}}{\text{Pre}+\text{Rec}} \quad (5)$$

For classification of multi - class label, precision and recall are calculated for each class label individually or by taking average of all class together, final precision and recall values can obtained. The Table 8, given below provides the accuracy comparison of different deep learning based text classification models.

Several experiments for the text classification task on various datasets were executed to obtain the performance of the model in terms of accuracy. It has observed that, for text classification most of the models have performed sound on a different dataset, mainly deep - CNN for sentiment prediction as well as for news classification. In addition, the hybrid architecture based on a variant of RNNs with CNN has given adequate performance. Meanwhile, it was also noticeable that the uses of transformer architecture, Bidirectional Encoder Representations from Transformers (BERT) in different text classification datasets produce significant performance in comparison to CNN and RNN.

6. CONCLUSION

Continuous increasing use of text messages for communication over the internet results in the need for text classification. In this paper, the standard steps of text classification task like the collection of data, pre-processing of text data and different text classification architecture with application area was discussed. Consequently, deep learning approaches are used to do this. It was also observed that multiple deep learning techniques are encapsulated to discover new architecture called a hybrid deep learning based model for better analysis and accuracy by utilizing the concept of NLP.

https://doi.org/10.6703/IJASE.202309_20(3).0028

Table 8. Accuracy comparison of text classification models

Architecture	Dataset	Accuracy	References
Character-extent CNN	Amazon-2 (SA)	94.49	Zhang et al. (2015)
	Yelp-2 (SA)	95.12	
Deep-CNN	Amazon-2 (SA)	96.68	Johnson et al. (2017)
	Yelp-2 (SA)	97.36	
BLSTM-CNN	SST2 (SA)	89.50	Zhou et al. (2016)
BERT (base)	Amazon-2 (SA)	96.04	Devlin et al. (2018); Sun et al. (2019)
	Yelp-2 (SA)	98.08	
	IMDB (SA)	95.63	
	SST2 (SA)	93.50	
FastText	AG News (NC)	92.50	Joulin et al. (2016)
	SogouNews (NC)	96.80	
Deep - CNN	AG News (NC)	93.13	Johnson et al. (2017)
	SogouNews (NC)	98.16	
BERT (large)	Dbpedia (NC)	99.32	Xie et al. (2020)
LSTM-Self Attention-ELMO	SQuAD1.1 (QA)	85.83	Sarzynska et al. (2021)
BERT-Nested-CNN	SQuAD2.0 (QA)	86.76	Liu et al. (2017)

Neural networks of the future are not only “deep”, but also conversely, they can process knowledge in an accelerated way. An interesting example is repeated attention-grabbing visual patterns. Among the various available models for the Text Classification task, the Transformer model is best suited method in the application of the text classification area with more percent of accuracy, whereas other art of state models are limited and restricted with less percent of accuracy than the transformer model. So later, more attention was given to developing a deep learning based model by using Transformer architecture like BERT for textual data classification and related NLP tasks.

REFERENCES

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146.

Chen, J., Huang, H., Tian, S., Qu, Y. 2009. Feature selection for text classification with Naïve Bayes. Expert Systems with Applications, 36, 5432–5435.

Chen, Y. 2015. Convolutional neural network for sentence classification (Master's thesis, University of Waterloo).

Cheng, J., Dong, L., Lapata, M. 2016. Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601-06733.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. 2011. Natural language processing from scratch. Journal of Machine Learning Research, 12, 2493–2537.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.arXiv preprint arXiv:1810–04805.

Dhuliawala, S., Kanojia, D., Bhattacharyya, P. 2016. Slangnet: A wordnet like resource for english slang. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, 4329–4332.

Fahad, S.A., Yahya, A.E. 2018. Inflectional review of deep learning on natural language processing. International Conference on Smart Computing and Electronic Enterprise IEEE., 1–4.

Gupta, G., Malhotra, S. 2015. Text document tokenization for word frequency count using rapid miner. International Journal of Computer Applications, 975, 8887.

Gupta, V., Lehal, G.S. 2009. A survey of text mining techniques and applications. Journal of Emerging Technologies in Web Intelligence, 1, 60–76.

Hinton, G.E., Osindero, S., Teh, Y.W. 2006. A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

Joachims, T. 2005. Text categorization with support vector machines: Learning with many relevant features. In Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings, 137–142. Berlin, Heidelberg: Springer Berlin Heidelberg.

Johnson, R., Zhang, T. 2016. Supervised and semi-supervised text categorization using LSTM for region embeddings. In International Conference on Machine Learning PMLR, 526–534.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T. 2016. Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612–03651.

Kaggle, 2021a, https://www.kaggle.com/datasets/yelpdataset.

Kaggle, 2021b, https://www.kaggle.com/lakshmi25npathi/imdbdataset-of-50k-movie-reviews

Kalchbrenner, N., Grefenstette, E., Blunsom, P. 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404–2188.

Karhunen, J., Raiko, T., Cho, K. ,2015. Unsupervised deep

learning: A short review. Advances in independent component analysis and learning machines, 125–142.

Kim, Y., Jernite, Y., Sontag, D., Rush, A.M. 2016. Character-aware neural language models. In Thirtieth AAAI Conference on Artificial Intelligence.

Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E. 2017. Hdltex: Hierarchical deep learning for text classification. In 16thIEEE International Conference on Machine Learning and Applications IEEE. 364–371.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Bizer, C. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6, 167–195.

Liu, P., Qiu, X., Huang, X. 2016. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605–05101.

Liu, X., Shen, Y., Duh, K., Gao, J. 2017. Stochastic answer networks for machine reading comprehension.arXiv preprint arXiv:1712–03556.

Liu, Y., Sun, C., Lin, L., Wang, X. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv preprint arXiv:1605–09090.

LozaMencía, E., Fürnkranz, J. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases 50-65.

Luštrek, M., Gams, M., Martinčić-Ipšić, S. 2016. What makes classification trees comprehensible. Expert Systems with Applications, 62, 333–346.

Bijaksana, M.A., Li, Y., Algarni, A. 2013. A pattern based two-stage text classifier. In Machine Learning and Data Mining in Pattern Recognition: 9th International Conference, MLDM, Proceedings 9, 169–182. Springer Berlin Heidelberg.

Haddoud, M., Mokhtari, A., Lecroq, T., Abdeddaïm, S. 2016. Combining supervised term-weighting metrics for SVM text classification with extended term representation. Knowledge and Information Systems, 49, 909–931.

Mahto, D., Yadav, S.C. 2022. Hierarchical Bi-LSTM based emotion analysis of textual data.Bulletin of the Polish Academy of Sciences. Technical Sciences,70.

Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52ndAnnual Meeting of the Association for Computational Linguistics: System Demonstrations, 55–60.

Mawardi, V.C., Susanto, N., Naga, D.S. 2018. Spelling correction for text documents in Bahasa Indonesia using finite state automata and Levinshtein distance method. In MATEC Web of Conferences, 164, 01047. EDP Sciences.

Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Hodjat, B. 2019. Evolving deep neural networks. InArtificial intelligence in the age of neural networks and brain computing, Academic Press, 293-312.

Mikolov, T., Chen, K., Corrado, G., Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301–3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. 2013. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26. http://qwone.com/~jason/20Newsgroups/

Pahwa, B., Taruna, S., Kasliwal, N. 2018. Sentiment analysis-strategy for text pre-processing. International Journal of Computer Applications, 180, 15–18.

Pennington, J., Socher, R., Manning, C.D. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on empirical methods in Natural Language Processing (EMNLP), 1532–1543.

Prusa, J.D., Khoshgoftaar, T.M. 2016. Designing a better data representation for deep neural networks and text classification. In EEE 17thInternational Conference on Information Reuse and Integration, 411–416.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1, 9.

Rajpurkar, P., Jia, R., Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822.

Saif, H., Fernández, M., He, Y., Alani, H. 2014. On stopwords, filtering and data sparsity for sentiment analysis of twitter.

Sampson, G. 2005. The'LanguageInstinct'Debate: Revised Edition. A&C Black.

Santos, C.D., Tan, M., Xiang, B., Zhou, B.2016. Attentive pooling networks. arXiv preprint arXiv:1602–03609.

Sarzynska-Wawer, J., Wawer, A., Pawlak, A., Szymanowska, J., Stefaniak, I., Jarkiewicz, M., Okruszek, L. 2021. Detecting formal thought disorder by deep contextualized word representations.Psychiatry Research,304, 114135.

Sun, C., Qiu, X., Xu, Y., Huang, X. 2019. How to fine-tune bert for text classification? InChina national conference on Chinese computational linguistics Springer, Cham, 194–206.

Tai, K.S., Socher, R., Manning, C.D. 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503–00075.

Tang, D., Qin, B., Liu, T. 2015. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1422–1432.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Polosukhin, I. 2017. Attention is all you need. Advances in Neural Information Processing Systems, 30.

Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang,

X., Carin, L. 2018. Joint embedding of words and labels for text classification. arXiv preprint arXiv:1805–04174.

Wang, H., Raj, B.2017. On the origin of deep learning.arXiv preprint arXiv:1702–07800.

Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q. 2020. Unsupervised data augmentation for consistency training.Advances in Neural Information Processing Systems,33, 6256–6268.

Yang, Y., Pedersen, J.O. 1997. A comparative study on feature selection in text categorization. In Icml, 97, 35.

Yang, Y., Yih, W.T., Meek, C. 2015. Wikiqa: A challenge dataset for open-domain question answering. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013–2018.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. 2016. Hierarchical attention networks for document classification. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1480–1489.

Zhang, X., Zhao, J., LeCun, Y. 2015. Character-level convolutional networks for text classification. Advances in Neural Information Processing Systems, 28.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., Xu, B. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv:1611–06639.

Zhou, X., Wan, X., Xiao, J. 2016. Attention-based LSTM network for cross-lingual sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 247–256.

https://doi.org/10.6703/IJASE.202309_20(3).00211