International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

Research Trends on Deep Transformation Neural Models for Text Analysis in NLP Applications

T.Chellatamilan, B.Valarmathi, K.Santhi

Abstract*: *In the recent few years, text analyses with neural models have become more popular due its versatile usages in different software applications. In order to improve the performance of text analytics, there is a huge collection of methods that have been identified and justified by the researchers. Most of these techniques have been efficiently used for text categorization, text generation, text summarization, query formulation, query answering, sentiment analysis and etc. In this review paper, we consolidate a recent literature along with the technical survey on different neural models such as Neural Language Model (NLM), sequence to sequence model (seq2seq), text generation, Bidirectional Encoder Representations from Transformers (BERT), machine translation model (MT), transformation model, attention model from the perception of applying deep machine learning algorithms for text analysis. Applied extensive experiments were conducted on the deep learning model such as Recurrent Neural Network (RNN) / Long Short-Term Memory (LSTM) / Convolutional Neural Network (CNN) and Attentive Transformation model to examine the efficacy of different neural models with the implementation using tensor flow and keras.

Keywords: BERT, RNN, CNN, Language model, seq2seq, text summarization, text generation, text mining.

I. INTRODUCTION AND PREVIOUS REVIEW STUDIES

Text analysis or Text Mining has been significantly improvised with the state-of-art technology across a different natural language application with the help of advanced machine learning and deep learning techniques. The pre-trained models were built using these techniques along with a large collection of training samples for predicting the word or terms in the text based on the context of those words or terms used in the training corpus .The major challenges and issues faced by the pre-trained models have been summarized as follows:

a) Limitation of representation of the vocabulary of the language
b) Low speed of convergence of the training/testing measures
c) Exposure Bias
d) Remembering the long term memory and preserving the order of sequence of words/terms

In earlier days the statistical language model is constructed with the help of contextual and co-occurrence properties of the words in the text corpus along with their prior probability of the words in the existing corpus (Lewis et al. 1967) [1]. The language model is also playing a major in automatic machine translation, whereas as the sentence written and spoken native language can be translated into the sentence of foreign language. This process of translation requires two passes the first one uses a conventional back-off language model to produce a top list of words in the source language and second pass produces the final translation by producing the top-k corresponding words in the target language [2]. The cross lingual language model is also built to represent the words of the sentence constructed with the combination of words from multiple languages. Usually the social media users are using such cross-lingual comments and post when they convey their views to other peer members of the social media. During such case we can use the cross-lingual language model to construct the original meaning of the sentence automatically [3].
The conventional n-gram model can be converted into a neural network language model to improve the prediction accuracy and reduce the word error rate (WER) [4]. But the traditional neural network language model is not maintaining the sequence order by which the words of the sentence are formed in the source. The succession of the language in the following convictions are different, but the neural network train the net without any deviation in these two sentences ex. “The movie is good not at all a bad”. “The movie is bad, not at all a good”. The RNN solves this problem, it maintains the order of sequence of words and then it construct the language model so that the model can distinguish the difference between these two sentences [5].

II. LANGUAGE MODEL

The Language Model (LM) Contributes crucial role in text analytics. LM is a model of representation of discrete words occurs in the text corpus as in terms of bag-of-words or in terms of n-gram.

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
***** Correspondence Author
T.Chellatamilan, Associate Professor, School of Information Technology and **Engineering** Vellore, India
B.Valarmathi, Associate Professor, School of Information Technology and Engineering Vellore, India
K.Santhi, Associate Professor, School of Computer Science and Engineering Vellore Institute of Technology, Vellore, India

After applying the text pre-processing operations over the text for removing the punctuation and stop words, the word or the term has been taken for the conversion of character sequence into a numerical entity named as document vectors or as word hash. The language model is also helpful in building Topic models that will predict the topic of the given segment of text using multinomial logistic regression based neural networks. The language models are also helpful constructing a query likelihood model determine the best query to retrieve the most relevant information by reformulating the query according to the notion of importance of the query terms using graphical model (Hiemstra, 2002, Hao et al., 2016).

The power of statistics and the art of grammar, writing jointly improving the performance of automatic reply the dialog systems depending on the different point of contextual constraints and dialog (Jonson, 2007). With respect to the conditional probability distribution of words named as probabilistic language model, the next word to be occurred in the sentence sequence is being determined (Mnih et al., 2007). The web documents can be filtered automatically based on the topic of interest based on the n-gram counts [6]. The multi-agent system uses the domain specific language models for gathering the syntactically and semantically matching contents to automatically generate the programming language code through model transformation [7].

A.Bag of Words (BOW):

It is a type of language model which extracts the text features of the text collection. The occurrence of words within the text documents involves two things, one is the vocabulary of known words and other one is the count or the frequency of each word in the text collections. One of the main drawbacks of the back of model is that it is not preserving the order of sequence of words appears in the text collections.

The sample text from the English Tirukkural corpus is shown and its vocabulary is also visualized in Figure 1 along with its frequency count occurrences. The scoring of the word is represented in different ways like binary score, frequency score, hash trick and Term-Frequency and Inverse Document Frequency (TF-IDF).

The probability on any particular sequence of n words: $P(w_1, w_2, \cdots, w_n)$, Mathematically and taking the unary language model approach and break apart this possibility by supposing the word instances are completely autonomous and it is shown in the Eq. (1).

$$P(w_1, w_2, w_3 \ldots . w_n) = \frac{Count(w_1, w_2, w_3 \ldots w_n)}{Count(possible\ sentene)} \quad (1)$$

After removing the stop words like ‘is’, ‘the’, ‘of’ & ‘and’ in the document 1 and document 2. Next, find the term-frequency for the document 1 and document 2 after removing stop words and it is shown in Figure 1.

Term	Doc1	Doc2
A	1	1
first	1	0
alphabet	1	0
God	1	0
primary	1	1
force	1	0
world	1	0
Vigilance	0	1
erudition	0	1
boldness	0	1
three	0	1
never	0	1
abandon	0	1
ruler	0	1

Document 1

A is the first of the alphabet
God is the primary force of the world

Document 2

Vigilance, erudition and boldness:
These three never abandon a primary ruler

Stop Words in document 1 and document 2 are ‘is’, ‘the’, ‘of’ & ‘and’.

Figure 1: Bag of Word Model

B.N-Gram Model:

An N-gram is a two-word sequence of words like “primary force”, “never abandon”, or “primary ruler”, and a 3-gram (three never abandon) is a three-word sequence of words.

$$Unigram\ LM:\ p(w_n) = \prod_{i=1}^{|v|} P(w_i) \quad (2)$$

$$Bigram\ LM: p(w_n) = \prod_{i=1}^{|v|} P(w_i|w_{i-1}) \quad (3)$$

$$Trigram\ LM: p(w_n) = \prod_{i=1}^{|v|} P(w_i|w_{i-1}, w_{i-2}) \quad (4)$$

Whereas |v| is the size of the vocabulary

The Eq. (2) is used for constructing the unigram language model which considers the probability of occurrence of each unique word present in the corpus whereas the Eq. (3) considers the two word sequence combinations and its probability of occurrences of the two words. The Eq. (4) considers the sequence of three words and their probability of occurrences. Text classification is another interesting application of language models. Here the labels of the classification dataset can be represented the topic about the text or sentimental about the text or it could be named entity.

In order to apply text mining in electronic patient records for the purpose of acquiring interesting diagnostic information which could be used in the decision making process of health care system. The details such as radiographic studies, laboratory results, complaints and physical findings relevant to the patients were analyzed [8]. Similarly Text Summarization is done to physically and logically reduce the size of the text corpus without losing any of the important meaning and information about the original text with the help of language models and advanced neural models. The statistical information about the large corpus is further analyzed using big data framework which enables distributed parallel data processing technique [9]. The discriminative features of each sentence had been identified along with the contextual sentence relations, query sentence relations and title sentence relations using a bigram language model [10]. The conditional influence of the sentence in a cluster is found by doing link analysis over the words and sentences in the text corpus which shows that all the sentences has some indistinguishable characteristics [11]. There are more than thirteen different text summarization techniques available, namely TextRank, LexRank, Luhn, LSA, Edmundson, ChunkRank, TGraph, UniRank, NN-ED, NN-SE, FE-SE, SummaRuNNer, and MMR-SE with the constraint of different set of metrics [12].

The Text summarization has been categorized into two different types as abstractive text summarization and extracting text summarization with low inter sentence cohesion [13]. The cohesion and coherence properties of the text sentences are captured by applying the computation of similarities of the distributional representations of the words and sentences during the sentence ordering task [14]. The matching models with the combinations of click model and content model regularize the mapping of large features into the latent features which reduces the time complexity over probabilistic latent structures [15].

III. WORD EMBEDDING MODEL

Word embedding is a method of distributing depiction of the distinct word of the text file in such a way that the semantically similar meaning words have been nearer to each other by indicating these words into the vector space model in terms of real numbers. There are many such pre-trained word embedding models are available namely word2vec by Google, Glove by Stanford and fastest by Facebook.

A Word is a primary unit of a language intern that infers the meaning by its own. An infinite set of concepts were constructed with the help such words and language rules. To enable easy processing of the words by the machine, the words should be translated into a numerical quantity named word vector and then all the words in the vocabulary of the language were represented in a vector space to convey the semantical relationships between each words. The parameter estimation technique estimate the parameter of the sentences using probabilistic computation [16]. The query embedding vectors are assessed based on the individual embedding vectors of the vocabulary terms in the information retrieval system [17]. One of such characteristic parameter is the total number of occurrences of a given pattern with length ‘p’ in a given subsequence of a random text with length ‘s’ [18]. The query terms must be present in the query have been estimated through the probability of occurrence of such query terms in the answer to be retrieved for that query [19]. In earlier days the optimizations were applied in the small size of the database but recently we need to optimize the information retrieval techniques for big data (Blair et al., 1985). The adhoc clinical questions were answered automatically through automatic topic identification and keyword extractions [20]. Identification of Autism Spectrum Disorder (ASD) model solves the demerits of existing approaches and has the advantages of classifying the children with ASD accurately with high precision and recall as well as a new model of LSTM based deep learning neural network has used [61]. Word prediction is useful for disabled persons and IQ of Autism children will be tested based on latent semantic analysis. word embedding, sentence embedding, and sequence-to-sequence modelling, learning and reasoning like Natural Language Processing (NLP) tasks are performed by latest neural network based frameworks [21]. All the words of the language have been mapped into a semantic vector space. For example the animal words like cat, dog and rat are placed closed whereas the color words such as red, green, blue are placed closed but far away from the animal words. From the figure 2, we can infer the semantic similarities between the words man vs women and king versus queen.

King - Man + Woman = Queen

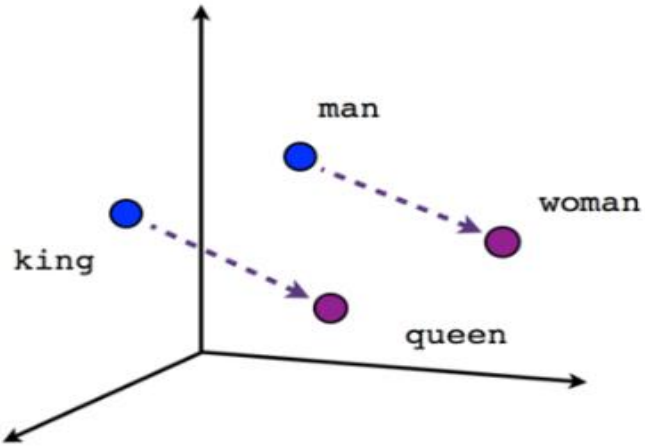
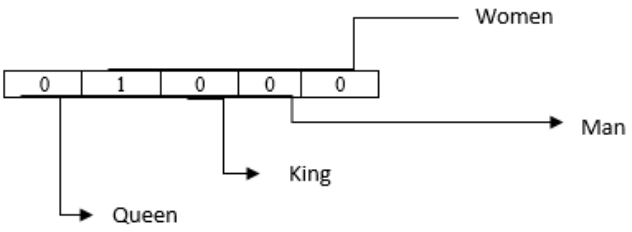


Figure 2: Semantic Vector Space (Bob DuCharme's)

A. One Hot Encoding:

This is the simplest method word embedding technique whereas each unique word present in the vocabulary, it map such unique word in to the unique vector. The size of the vector is equivalent to the size of the vocabulary. The i^{th} word of the vocabulary is mapped into a vector by switching i^{th} position of the vector on (“1”) all other positions are marked with “0”. The figure 3 shows the sample representation of the on hot vector for certain words.



$$Queen = \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} \quad King = \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \quad Man = \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \quad women = \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}$$

Figure 3: One hot Vector

B. Probabilistic Neural Models for Embedding: CBOW and Skip Gram

There are principle mode of generating the word embedding vectors for the given word/words, namely Continuous Bag of Words (CBOW) and Continuous Skip-Gram model. The Figure 4 shows the word2vec that learns the distributed vector representations. The artificial neural network (ANN) is very popular in solving NLP problems with the help of machine learning and deep learning networks. The ANN learns itself from the training data to automatically determine the title of the research paper for the given sentences of an abstract of the paper to be published by the researchers [22]. Deep belief network trains a multi-layer generative model with unlabeled data and the feature vectors are generated with the help of feed forward neural network [23].

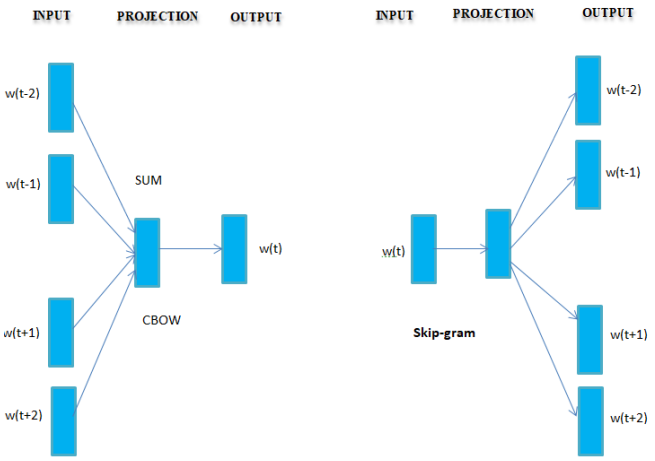


Figure 4: word2vec model

Continuous Bag of Words model is forecasting the current word with the support of given set of context words (surrounding words). The Continuous Skip-gram model is predicting the set of context words with the help of given current words.

C. Recurrent Neural Network (RNN)

The traditional neural network is poor in capturing sequential data from the previous event and pass this knowledge in also poor predicting the future event. The RNN deals with most of the problems related to the sequential data in such a way that it stores lot of past information to predict the future information with the help of hidden states. Similarly the traditional feed forward neural network generates output based on the current only but RNN generates output using the current input as well as by using the previous input.

For example if we feed sequence of binary digits as input to the RNN, the RNN network learns to predict the number of 1's or 0's in the given stream of binary digit. For NLP application like sentiment analysis, if we feed a sentence with sequence of words as input to the RNN, the RNN train itself on the sentiment label and then l generate the polarity of the sentiment of the input either as positive or negative without understanding the thousands of grammar rules present across the language of the comments. The RNN is also very useful in building language models and sequential model in the prediction problems.

The RNN is a special variation deep neural network suitable for sequential data driven problems. The networks maintain the order of sequence of data or evet that appears in the time stamp. One major problem in the case of RNN is the vanishing / exploding problem due to the long sequence convergence. In this case the LSTM, a variation of RNN has been used to retain only the required past data to be remembered in the process to predict the future events and other things. The Bidirectional LSTM keep track of processing the sequence of inputs in two directions one from star to end and another one from end to start in the input sequence. A real time use cases of such Bi-LSTM is the news detection using unstructured news articles [24].

One of the most difficult task in text analysis is to identify features and encode them into discriminative feature vectors .The word level attentive pooling in convolution neural network was used to represent the contextual sentence relations [25] . The CNN is more popular noble approach to extract the high level features that are more relevant for the local translations with substitution and pooling principles [26]. The text features are extracted and transformed into embedding vectors using NgramCNN that incorporate the linguistic knowledge of the native and foreign languages [27]. The variable convolution and pooling convolution neural network have been used as multiple convolutions for text sentiment predictions [28]. The combinations of single layer multi size filters with convolution neural network have jointly used for large multipurpose and multi format dataset with different ML algorithms [29]. The composition of topic aware convolution network and topic aware LSTM process the text and build the topic model for focusing on the semantic composition rather than word understandings [30]. The discrete valued high dimensional vectors and the distributions are estimated by neural autoregressive estimator model [31].

Deep learning fails to address the issue in building the models for text when the text there is any absence of labels. In such situation the transfer learning is being used for domain adaption and parameter update to derive the respective labels of the text collections [32]. The most valuable local information of the text documents are extracted from the large scale scope based convolution neural networks with aggregation optimizations [33]. The following figure 5 and 6 are drawn for depicting the vanilla version and unfold version of the RNN. The Eq. (5) and Eq. (6) have been used for the calculation intermediate output and final output respectively.

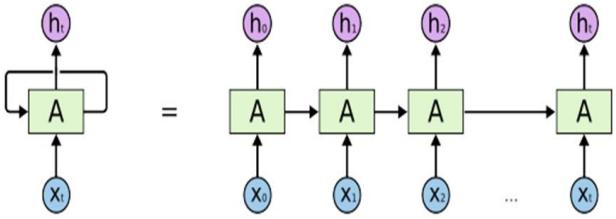


Figure 5: RNN Vennila model

$$h_t = \tanh(W\ x_t + U h_{t-1} + b) \quad (5)$$

y_t = sof tmax(Woh_t + bo) (6)

Tensor shapes are given below:

- x_t is of shape (1, d) for embedding of size d
- h_t is of shape (1, H) for hidden state of size H
- y_t is of shape (1, c) for c labels
- W is of shape (d, H)
- U is of shape (H, H)
- Wo is of shape (c, H)

The Language models have been constructed based on x_t the current word and the model predicts the y_t the next word by estimating the probability of the next word from the current sequence of word through estimate $P(w_i|w_{i-1}, h_{i-1})$. The weight vectors U, V and W have been initialized randomly by real numbers in matrix format. There is bulk of operations such as tanh and softmax functions are applied on the current input and previous input along with weight vectors. The output of the desired behavior is measured with the help loss functions and then the weight vectors are further updated for the next iteration during the stage of training. As the weights are recursively multiplied over the period of training time during the back propagation, the RNN raises two types of problems named as vanishing gradient and exploding gradient problem.

The vanishing gradient problem problems appears in the network when the subsequent weight vector values are getting smaller and then tends to zero. Similarly the exploding gradient problem problems appears in the network when the subsequent weight vector values are getting bigger and then tends to infinity.

D.Long Short Term Memory (LSTM)

It is a variation of RNN used for learning long distance dependence between the input sequences. As the input time sequence increases in RNN, the corresponding possible weights also increased beyond control or to vanish at a particular time sequence. In order to solve this exploding gradient and vanishing grading problems, the LSTM has been proposed to train the long term dependency among length time input sequence with the help of three gates like input gate, output gate and forget gate and the estimate the how long the old information should be hold and when it remembers and forget the old memory with the new input. LSTM can also be used for capturing the long term dependency among the sequence of words in the short text during the process of text classification or sentiment analysis each of the text which could be classified as positive polarity and negative polarity.

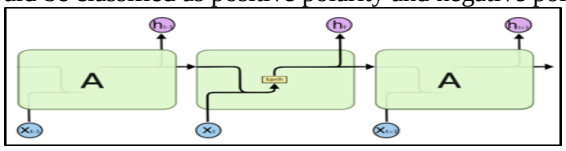


Figure 6.1: RNN with single module layer (Cho)

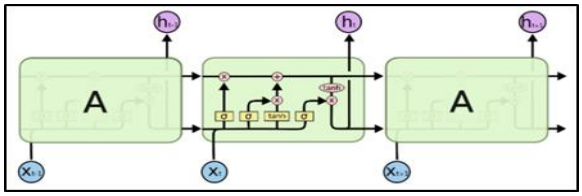


Figure 6.2: LSTM with Four module layer (Cho)

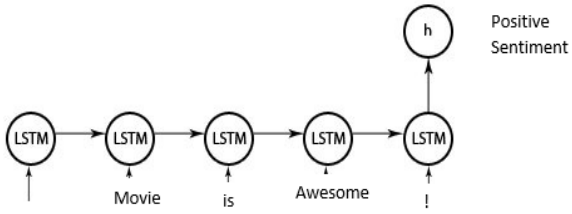


Figure 7: Illustration of our LSTM model for sentiment classification

IV. SENTIMENT ANALYSIS

Sentiment analysis another emerging field of research which requires the help of natural language processing to determine the polarity of the comments or feedbacks using the emotional dictionary constructed from the keywords, word2vec and LSTM [34]. The multi-layer LSTM network takes the commodity comments on e-commerce website and expresses the comment text in terms of word2vec model [35]. The recommendation system is constructed with the help deep learning .It provides the comprehensive review of recent research and devise a taxonomy of deep learning based recommendation system [36]. The user sentiment about the context is extracted from the text with 7 layers CNN, which tag each aspect of opinion with rule based approach during the process of sentiment scoring [37]. The user profiling system extracts the semantical information about the users by mapping the features using deep similarity model and apply the ranking, the users according to their profile of interest (Azzam et al., 2017). The inter model and intra model semantic relationships of heterogeneous sources of language sentences can be used for giving the review about a movie [38]. The SWOT analysis about a firm could be analysed from the raw facts and conditions provided through the question answer system in the form of text sentences [39]. Linking of multiple data sources during the text mining and the process of admission of patients into the hospital marked positives for several diseases were taken into the consideration of examining the effect of disease [40]. The products and its underlying user's sentiments were captured with sentence aware deep recommender system constructed with the help of attention networks [41]. The internal filtering gates were used for input, output and forget purposes with the help of the different components as shown in the figure 6.2. LSTM model for sentiment classification is shown in the figure 7.

A.Sequence to Sequence Model (Seq2Seq: Model

LSTM is a good choice for giving importance to recent information than the old information, but it is poor in performance when we want to go back further into the previous event to resolve the better predictions. To solve this issue in this case the sequence model and attention mechanism are playing major role which in turn gives the importance to a particular portion of the input.Seq2seq model is a general purpose encoder and decoder network, which generate a sequence of output based on the given sequence of inputs. It is widely used in the NLP applications like machine translation, Text summarization, conversation model and image captioning.

The Seq2seq model uses 2RNN, one for encoder and another one for decoder whereas the learning starts from the backpropagation between decoder to encoder. Let us consider the example as shown in the figure 7 to translate the English sentence "The Movie is awesome" to Tamil language target sentence. In this the sequence of words in English is given as input to a encoder and encoder converts these words in to a fixed length context vector that can be used to predict the sequence of output target sentence in tamil as **இந்த திரைப்படம் அற்புதமாக இருக்கிறது!.**

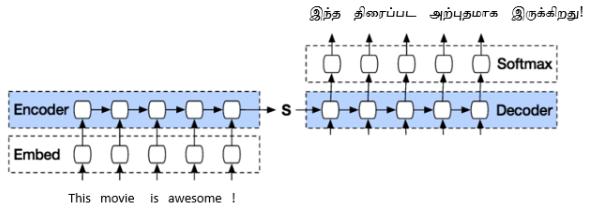


Figure 7.1: Illustration of seq2seq model for Language Translation

$$p(t_1,t_2,t_3,\dots t_T|e_1,e_2,e_3,\dots e_E) = \prod_{i=1}^{T}(t_i|v,t_1,t_2,t_3,\dots t_{i-1}) \quad (7)$$

Where

$e_1,e_2,e_3,\dots e_E$ is the input / source / English sentence sequence $t_1,t_2,t_3,\dots t_T$ is the corresponding output / target / tamil sentence sequence. 'v' is the hidden state of the LSTM

B. Machine Translation

Machine language translation tries to use the deep learning principle to build model line se2seq encoder decoder that accept the input source native language sentences and map them into a fixed length output target foreign sentences and vice versa where the size of the input text and size of the output text to the model differs. On such novel context-aware recurrent neural network based encoder named CAEncoder was widely used for sentence level translation to document level translation [42]. The speech processing and recognition is also an application of natural language processing based on the generative auto encoder modelling with latent variable learning algorithms [43]. The activity state transitions and its representations uses the seq2seq model using deep learning framework to recognize and adapt the situation with respect to the current instantaneous changes of the activity state Zhu et al., 2018). The child language acquisition under miniature language paradigm that allow different language learners able to be trained and analysed under standardized conditions [44]. Massively unlabelled data and its high level representations are extracted using deep generative models using the probability perspective features [45][62]. Educational data mining applies the generative deep learning model to process the longer sequence of learning patterns and then produce a new sequence of underlying sub patterns matched with community of peer learners [46]. The character level sequence model tries to use the POS tagger for Sanskrit sequence labelling with adaptive deep learning algorithms [47]. The inter model and intra model semantic relationships of heterogeneous sources of language sentences can be used for giving the review about a movie [38] . Massively unlabelled data and its high level representations are extracted using deep generative models using the probability perspective features [45] [63]. Educational data mining applies the generative deep learning model to process the longer sequence of learning patterns and then produce a new sequence of underlying sub patterns matched with community of peer learners [46]. The figure 7.1 shows the seq2seq model for translating English to Tamil language sentences and the Eq. (7) has been used for predicting the Tamil words form the sequence of English words.

V. ATTENTION MODEL

The performance of seq2seq model is declining due to the compression of all the information of the very long source input sentence into a fixed length context vector. Bhadanau proposed an attention model to overcome the issue based on the word alignment scheme of the words found in the source and target sentences. The encoder computes an annotation of each word in the input. The mechanism allows the model to focus and place more "Attention" on the relevant parts of the input sequence as needed. The network reads a sentence and stores all the information in its hidden units. It makes predictions one word at a time, and its predictions are fed back in as inputs. t it also receives a context vector c(t) at each time step, which is computed by attending to the inputs. The context vector is computed as a weighted average of the encoder's annotations. Attention component of the network will ensure for each word in the output sentence is map the significant and related words from the input sentence and allot higher weights to these words, boosting the accuracy of the output prediction, This indicates that for each output that the decoder makes, it owns access to the complete input sequence and can selectively pick out specific components from that sequence to produce the output.

$$C^{(i)} = \sum_j a_{ij} h^{(i)} \quad (8)$$

The attention weights are computed as a softmax, where the inputs depend on the annotation and the decoder's state. The Eq. (8), Eq. (9) and Eq. (10) were used for calculating the context vector, normalized weighted output of each unit and final target output-word for the corresponding input sentence respectively.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})} \quad (9)$$

$$e_{ij} = a(s^{(i-1)}.h^{(j)} \quad (10)$$

The attention function depends on the annotation vector, rather than the position in the sentence.

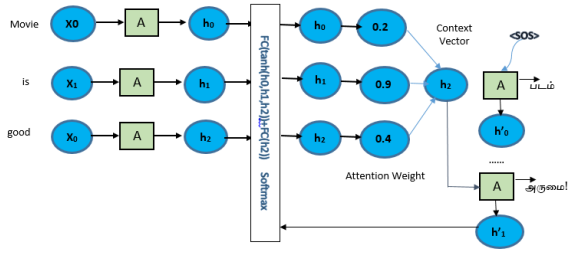


Figure 8: Illustration of Attention model proposed by [Bahdanau, 2014]

Attention Model or mechanism focus the importance of portions of the input sentences rather than giving importance to all the words in the sentences when it is given as input in any of the language processing tasks. For example the sentences like "I like it" and "I dislike it", the importance is given only to the second word of the sentences than any other words present in those two sentences during the period of sentence processing. In information retrieval system, the we generally include the query terms which plays a major contribution as a keyword to fetch the relevant results from the search space [48]. Sequential context knowledge like temporal information, word and character level information are influencing the parts of constructing the potential query to be used in text processing applications [49].

The attention mechanism with LSTM automatically describes the video content with native language using natural language sentences and it explores the salient feature of the video and correlation between multi model representations for generating sentences with rich semantic content [50]. Generation of probability to re rank the candidate answers for question answer system was implemented by obtaining word level and phrase level factoid [51] . The dual attention model uses word2vec tool to build emotional dictionary combining with emotional symbol for microblog sentiment classification [52]. Attention vectors are generated based on the calculation words at different positions of the sentence with the same distance using self-attention variation mechanism [53]. Scene text recognition is capturing the visual cues and generates a language model and linguistic rules from the output of the decoder of the attention model [54]. The bipolar concept model generates the concept vector using the concepts of documents and categorize them into relevant and irrelevant words [55]. The frame level feature information is analysed in combination with LSTM and attention based speech recognition system [56]. The text documents were arranged in an hierarchical order for shortening the length of the sequence of the whole document with over all attention weights and joint embedding space of text and label [57]. The importance of each of the word in a sentence is also predicted based on the position of occurrence of the word in that sentence has been estimated from the attention based weighting process [58]. The multi-head self-attention mechanism enhance the position information of the input text, which can enrich the semantic illustration of the text [59]. MS-Pointer Network that built on the multi-head self-attention mechanism, which a multi-head self-attention mechanism is presented in the basic encoder-decoder model (Guo et al., 2019).

Whenever each time the attention model predicts the output word, it considers only the part of input information rather than considering the entire input sentences. In this model the encoder is not only generating a single context vector rather it generates a collection of context vector which in turn decides which input word or words should be considered for generating the output word. The context vectors are generated by the encoder using the computation of weighted sum of the annotations of the input words for the corresponding output word. An Alignment model has been used for generating such weighted sum of annotations of input words for predicting the output word. An alignment identifies which English word each French word originated from. In a parallel text (or when we translate), we align words in one language with the words in the other d. Mapping an English target word at position i to a German source word at position j with a function a : i → j. A source word may translate into multiple target words. The sample implication of the translation of the individual English word in to the respective tamil words was done through the probability of tamil word for the given English word occurrences in the training corpus as shown in the figure 9.

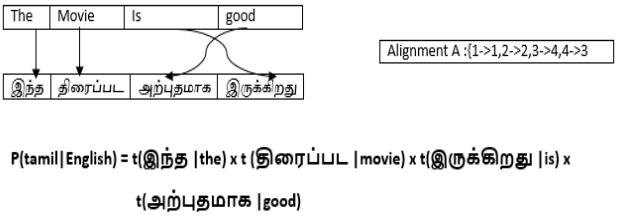


Figure 9: Sample mapping of English to Tamil words

VI. TRANSFORMER NETWORK

It is used to build a fine-tuned network for new task using pre trained model. During the process of machine translation, it is applying positional embedding and then evaluating the relevance score for each of the input terms to output. The relevance score any query term 'i' and its corresponding key terms 'j' is expressed in terms of Relevance Score [i,j] = Query [i] X Key [j] [60].The all to all comparison can be done fully parallel with the help of GPUs. The same word in different sentences may have different meaning with respect to the context at which the word has been used. The positional encoders are used for finding such vectors of the same word when it was used in different positions or contexts.

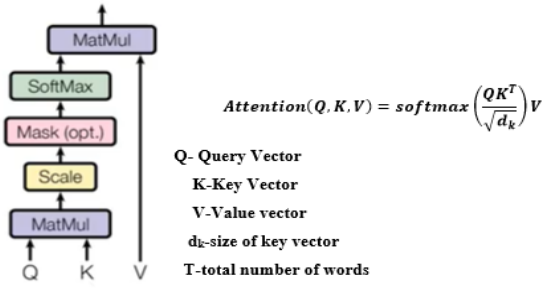


Fig 10: Illustrated Attention Mechanism [60]

The steps for generating the attention vector for the given input is visualized in the figure 10 .First step maps all the inputs word into the respective embedding vector.

Second step calculate the intra word score with respect to the remaining words present on the same sentence using self-attention mechanism. The score helps in determining how much importance should be given to the other parts of the sentence when it is being encoded with position encoder. The vectors Q, K and V have been updated during the training phase of the network to decide how much the key "K" kelps to represent Q with value V, where the value V is the projection of the each word in the sentence. The third and fourth steps are performed to normalize the score and then generate the attention weight vector to be used in the decoder part of the transformer.

VII. CONCLUSION

This paper, a mixed set of approaches of text processing and analytical methods have been studied for different level of natural language processing applications by extracting the essential features of the text corpus.

On the other hand, multi-layer deep learning architectures have been demonstrated which contains an input layer consisting of word embedding feature for each word in the input sentence and different set of layer followed by convolution layer max pooling layers, fully connected layer and dense layer. An attention based concept named transformer networks is also introduced to improve the performance of text feature extraction and applying the machine learning algorithm over the text. The comparisons of all such methods were illustrated with different examples.

REFERENCES

1. P. A. W. Lewis, P. B. Baxendale, and J. L. Bennett, "Statistical Discrimination of the Synonymy/Antonymy Relationship Between Words," *J. ACM*, vol. 14, no. 1, pp. 20–44, 1967.
2. L. H. Son, A. Allauzen, G. Wisniewski, and F. Yvon, "Training continuous space language models: Some practical issues," *EMNLP 2010 - Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, no. October, pp. 778–788, 2010.
3. P. Xu and P. Fung, "Cross-Lingual language modeling with syntactic reordering for low-resource speech recognition," *EMNLP-CoNLL 2012 - 2012 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn. Proc. Conf.*, no. July, pp. 766–776, 2012.
4. E. Arısoy *et al.*, "Deep Neural Network Language Models," *NAACL-HLT 2012 Work. Will We Ever Really Replace N-gram Model. Futur. Lang. Model. HLT*, pp. 20–28, 2012.
5. H. Fang, M. Ostendorf, P. Baumann, and J. Pierrehumbert, "Exponential language modeling using morphological features and multi-task learning," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 23, no. 12, pp. 2410–2421, 2015.
6. I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, and Ö. Çetin, "Web resources for language modeling in conversational speech recognition," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 1, 2007.
7. C. Hahn, "A domain specific modeling language for multiagent systems," *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 1, no. Aamas, pp. 230–237, 2008.
8. S. Tsumoto, T. Kimura, H. Iwata, and S. Hirano, "Mining Text for Disease Diagnosis," *Procedia Comput. Sci.*, vol. 122, pp. 1133–1140, 2017.
9. A. Mackey and I. Cuevas, "Automatic Text Summarization Within Big Data Frameworks," *J. Comput. Sci. Coll.*, vol. 33, no. 5, pp. 26–32, 2018.
10. P. Ren *et al.*, "Sentence relations for extractive summarization with deep neural networks," *ACM Trans. Inf. Syst.*, vol. 36, no. 4, 2018.
11. X. Wan and J. Yang, "Multi-Document Summarization Using," *Sigir*, pp. 299–306, 2008.
12. P. Verma, S. Pal, and H. Om, "A comparative analysis on Hindi and English extractive text summarization," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 18, no. 3, 2019.
13. R. Adelia, S. Suyanto, and U. N. Wisesty, "Indonesian abstractive text summarization using bidirectional gated recurrent unit," *Procedia Comput. Sci.*, vol. 157, pp. 581–588, 2019.
14. B. Cui, Y. Li, Y. Zhang, and Z. Zhang, "Text coherence analysis based on deep neural network," *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. Part F1318, pp. 2027–2030, 2017.
15. W. Wu, Z. Lu, and H. Li, "Learning bilinear model for matching queries and documents," *J. Mach. Learn. Res.*, vol. 14, pp. 2519–2548, 2013.
16. S. Y. Ihm, J. H. Lee, and Y. H. Park, "Skip-gram-KR: Korean word embedding for semantic clustering," *IEEE Access*, vol. 7, pp. 39948–39961, 2019.
17. H. Zamani and W. B. Croft, "Estimating embedding vectors for queries," *ICTIR 2016 - Proc. 2016 ACM Int. Conf. Theory Inf. Retr.*, pp. 123–132, 2016.
18. P. Flajolet, W. Szpankowski, and B. Vallée, "Hidden word statistics," *J. ACM*, vol. 53, no. 1, pp. 147–183, 2006.
19. C. T. Yu and G. Salton, "Effective information retrieval using term accuracy," *Commun. ACM*, vol. 20, no. 3, pp. 135–142, 1977.
20. Y. G. Cao, J. J. Cimino, J. Ely, and H. Yu, "Automatically extracting information needs from complex clinical questions," *J. Biomed. Inform.*, vol. 43, no. 6, pp. 962–971, 2010.
21. M. Zhou, N. Duan, S. Liu, and H. Y. Shum, "Progress in Neural NLP: Modeling, Learning, and Reasoning," *Engineering*, no. xxxx, 2020.
22. U. Khandelwal, "Neural Text Summarization," pp. 1–7, 2016.
23. S. Roukos, "Natural Language Understanding," *Springer Handbooks*, vol. 22, no. 4, pp. 617–626, 2008.
24. P. Bahad, P. Saxena, and R. Kamal, "Fake News Detection using Bi-directional LSTM-Recurrent Neural Network," *Procedia Comput. Sci.*, vol. 165, no. 2019, pp. 74–82, 2019.
25. P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. De Rijke, "Leveraging contextual sentence relations for extractive summarization using a neural attention model," *SIGIR 2017 - Proc. 40th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 95–104, 2017.
26. A. Hassan and A. Mahmood, "Convolutional Recurrent Deep Learning Model for Sentence Classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
27. E. Çano and M. Morisio, "A deep learning architecture for sentiment analysis," *ACM Int. Conf. Proceeding Ser.*, pp. 122–126, 2018.
28. M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, and Y. Cai, "Variable Convolution and Pooling Convolutional Neural Network for Text Sentiment Classification," *IEEE Access*, vol. 8, pp. 16174–16186, 2020.
29. M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network," *IEEE Access*, vol. 8, no. Ml, pp. 42689–42707, 2020.
30. R. Zhao and K. Mao, "Topic-Aware Deep Compositional Models for Sentence Classification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 2, pp. 248–260, 2017.
31. S. Lauly, Y. Zheng, A. Allauzen, and H. Larochelle, "Document neural autoregressive distribution estimation," *J. Mach. Learn. Res.*, vol. 18, pp. 1–24, 2017.
32. B. Wang, W. Liu, Z. Lin, X. Hu, J. Wei, and C. Liu, "Text clustering algorithm based on deep representation learning," *J. Eng.*, vol. 2018, no. 16, pp. 1407–1414, 2018.
33. J. Wang, Y. Li, J. Shan, J. Bao, C. Zong, and L. Zhao, "Large-Scale Text Classification Using Scope-Based Convolutional Neural Network: A Deep Learning Approach," *IEEE Access*, vol. 7, pp. 171548–171558, 2019.
34. H. Yuan, Y. Wang, X. Feng, and S. Sun, "Sentiment analysis based on weighted word2vec and ATT-LSTM," *ACM Int. Conf. Proceeding Ser.*, pp. 420–424, 2018.
35. H. Sun, T. Jiang, and Y. Dai, "Sentiment analysis of commodity reviews based on multilayer LSTM network," *ACM Int. Conf. Proceeding Ser.*, 2019.
36. S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, 2019.
37. P. Ray and A. Chakrabarti, "A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis," *Appl. Comput. Informatics*, 2019.
38. W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang, "Effective deep learning-based multi-modal retrieval," *VLDB J.*, vol. 25, no. 1, pp. 79–101, 2016.

39. H. Thamrin and E. W. Pamungkas, "A Rule Based SWOT Analysis Application: A Case Study for Indonesian Higher Education Institution," *Procedia Comput. Sci.*, vol. 116, pp. 144–150, 2017.
40. S. Kocbek *et al.*, "Text mining electronic hospital records to automatically classify admissions against disease: Measuring the impact of linking data sources," *J. Biomed. Inform.*, vol. 64, pp. 158–167, 2016.
41. A. Da'U and N. Salim, "Sentiment-Aware Deep Recommender System with Neural Attention Networks," *IEEE Access*, vol. 7, pp. 45472–45484, 2019.
42. B. Zhang, D. Xiong, J. Su, and H. Duan, "A Context-Aware Recurrent Encoder for Neural Machine Translation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 12, pp. 2424–2432, 2017.
43. Q. Zhang and J. H. L. Hansen, "Language/Dialect recognition based on unsupervised deep learning," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 5, pp. 873–882, 2018.
44. K. Jack, "A collaborative tool for the computational modelling of child language acquisition," no. March, pp. 10–17, 2009.
45. Y. Zheng, L. Wen, J. Wang, J. Yan, and L. Ji, "Sequence modeling with hierarchical Deep Generative Models with dual memory," *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. Part F1318, pp. 1369–1378, 2017.
46. S. Tang, J. C. Peterson, and Z. A. Pardos, "Deep neural networks and how they apply to sequential education data," *L@S 2016 - Proc. 3rd 2016 ACM Conf. Learn. Scale*, pp. 321–324, 2016.
47. M. Henderson *et al.*, "Efficient Natural Language Response Suggestion for Smart Reply," pp. 1405–1406, 2017.
48. H. J. Song, A. Y. Kim, and S. B. Park, "Translation of natural language query into keyword query using a rnn encoder-decoder," *SIGIR 2017 - Proc. 40th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 965–968, 2017.
49. C. Du, P. Shu, and Y. Li, "CA-LSTM: Search task identification with context attention based LSTM," *41st Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, SIGIR 2018*, pp. 1101–1104, 2018.
50. Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, and H. T. Shen, "Attention-based LSTM with semantic consistency for videos captioning," *MM 2016 - Proc. 2016 ACM Multimed. Conf.*, pp. 357–361, 2016.
51. Y. Peng and B. Liu, "Attention-based neural network for short-text question answering," *ACM Int. Conf. Proceeding Ser.*, pp. 21–26, 2018.
52. M. Shi, "Research on Parallelization of Microblog Emotional Analysis Algorithms Using Deep Learning and Attention Model Based on Spark Platform," *IEEE Access*, vol. 7, pp. 177211–177218, 2019.
53. L. Li, P. Gong, and L. Ji, "A deep attention network for Chinese word segment," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1481, no. 10, pp. 528–532, 2019.
54. S. Fang, H. Xie, Z. J. Zha, N. Sun, J. Tan, and Y. Zhang, "Attention and language ensemble for scene text recognition with convolutional sequence modeling," *MM 2018 - Proc. 2018 ACM Multimed. Conf.*, pp. 248–256, 2018.
55. D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, "ConceptVector: Text Visual Analytics via Interactive Lexicon Building Using Word Embedding," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 361–370, 2018.
56. Y. Xie, R. Liang, Z. Liang, C. Huang, C. Zou, and B. Schuller, "Speech Emotion Classification Using Attention-Based LSTM," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 11, pp. 1675–1685, 2019.
57. C. Gong, K. Shi, and Z. Niu, "Hierarchical text-label integrated attention network for document classification," *ACM Int. Conf. Proceeding Ser.*, pp. 254–260, 2019.
58. T. Belkacem, J. G. Moreno, T. Dkaki, and M. Boughanem, "AMV-LSTM: An attention-based model with multiple positional text matching," *Proc. ACM Symp. Appl. Comput.*, vol. Part F1477, no. 2, pp. 788–795, 2019.
59. F. Long, K. Zhou, and W. Ou, "Sentiment analysis of text based on bidirectional LSTM with multi-head attention," *IEEE Access*, vol. 7, pp. 141960–141969, 2019.
60. A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
61. T. Chellatamilan and B. Valarmathi, "Intelligent multi agent reinforcement Q-learning for the best practice recommendations of E-learning system," *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 2 Special Issue, pp. 2373–2379, 2019.
62. S. Guruvammal, T. Chellatamilan, and L. J. Deborah, "Word based language model using long short term memory for disabilities," *Int. J. Psychosoc. Rehabil.*, vol. 24, no. 6, pp. 6509–6513, 2020.
63. B. Valarmathi, K. Santhi, R. Chandrika, P. Goel, and B. Bagwe, "Performance analysis of genetic algorithm, particle swarm optimization and ant colony optimization for solving the travelling salesman problem," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 4, pp. 91–95, 2019.

AUTHORS PROFILE

Dr. T. Chellatamilan received his Ph.D., degree in Computer Science and Engineering from Anna University,Chennai in the year 2013. He has completed his Masters degree, M.Tech Computer Science and Engineeringfrom Manipal Institute of Technology in the year 2003. He completed his Bachelor of Engineering in Computer Engineering under Madurai Kamarajar University, Madurai in the year 1993. He is working as Associate Professor in the School of Information Technology and Engineering, Vellore Institute of Technology, Vellore. He has published many quality papers in reputed journals. He had travelled to many countries like United States of Malaysia, Singapore to present her research works in reputed conferences. His area of research interest includes Information retrieval techniques, big data analytics, Text analytics, eLearning, Computational intelligence, Deep learning and social media mining. He is an active life member of professional societies including MCSI and MISTE.

B.Valarmathi, PhD is Associate Professor Senior in the School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India. She holds PhD degree in computer science and engineering from Anna University, India. She has more than 27 years of experience in teaching and research. Her research interest includes Big-data analysis, Sentiment Mining, Soft Computing, Pattern Recognition and Machine learning. She has coauthored a text book on total quality management.

K Santhi, PhD is Associate Professor in the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. She holds PhD degree in computer science and Engineering from Pondicherry University, Puducherry, India. She has more than 20 years of experience in teaching and research. Her research interest includes Big Data Analytics, Machine learning and Algorithm Analysis.