Grenze International Journal of Engineering and Technology, Jan Issue

Coarser to Finer Level Document Classification through Recurrent Attention Mechanism using RL Agent

Ayesha Mariyam[1], Althaf Hussain Basha SK[2] and Viswanadha Raju S[3]
[1]Computer Science and Engineering, JNTUH, Hyderabad, India
ayesha.mariyam84@gmail.com
[2]Computer Science and Engineering,Krishna Chaitanya Institute of Technology and Sciences,Markapur, India
althafbashacse@gmail.com
[3]Computer Science and Engineering,JNTUH College of Engineering Nachupally, India
svraju.jntu@gmail.com

***Abstract*—Document Classification is a Natural Language Processing task, which generally uses deep neural networks to extract features from full textual information. The extracted features may or may not be relevant for classification of a document. We propose a framework to address the classification of long documents from coarse level to finer level by combining recurrent attention mechanism. It constructs the discriminative features with fewer words. The main idea is to train the recurrent neural network which focuses its attention on distinct parts of the document. It includes reinforcement learning agent at the word level for emitting the next block location to be glimpsed. Convolutional neural network (CNN) is used to extract glimpsed features from focused words. Both sentence and document representation can be obtained from word level and sentence level respectively. Experiments conducted on our collected 5-class arXiv papers dataset, the proposed method surpass the existing methods with less observed words.**

***Index Terms*— Document classification, deep learning, recurrent attention mechanism, deep reinforcement learning.**

I. INTRODUCTION

Document classification is a task of assigning pre- defined labels to text. It is a classic problem of information retrieval. Documents generally consist of structured textual information such as journal papers, books, business reports etc. Document classification is necessary for identifying and preventing the inappropriate content present in the document from distribution among people. With the advent of deep learning, document classification methods have been proposed, such as the convolutional neural networks, recurrent neural networks to comprehend text representations. In the first place, convolutional neural networks were mostly used for deep learning exploration. Hoa T et al [1] shows that deep models give better performance than shallow and wide convolutional neural networks when the input is a sequence of characters. Kim [2] proposed a new model by altering the CNN architecture; it extracts the relevant words of a sentence and converts them into vectors for classification. There are certain limitations exist with CNN. After additional exploration, new model like RNN or LSTM is perceived for remembering information for long period. Zhou et al [3] have combined both CNN and RNN architectures into new model called C-LSTM for document modeling. It outperforms both CNN and

Grenze ID: 01.GIJET.9.1.751

LSTM; achieve excellent results on sentiment classification. Although, these approaches have been effective for short text classification.

Although, the document comprises textual data, but document classification cannot be considered as the summation of the different text classification results. Sentiment classification and topic labeling is mainly intended for short text. But for lengthy textual documents such as novels, academic journals etc., directly employing the deep learning based methods would not be suitable. Because a long document would produce the very large word vector which burdens the memory.

In order to overcome the above limitation, the recurrent attention learning is combined with the deep learning and reinforcement learning is introduced. It requires only few glimpses to extract the meaningful features for classification [4]. It does not focus its attention from coarse level to finer level in a document.

Because the documents have a hierarchical structure, a document representation can be constructed by first creating representations of sentences and then aggregating them. An attention mechanism is included in the document hierarchical structure to pay more or less attention at both words and sentences in order to obtain document representation for classification [5]. This hierarchical attention network does not extract the most relevant parts from a lengthy document. Then in this regard, how to classify the lengthy documents by learning optimal policy through attention mechanism from coarse level to finer level is essential to design an effective model for classification.

In this paper, we propose a new model for document classification which does not make use of complete information present in the document; instead it only uses few glimpses to construct the distinguishing features for classification. For the purpose to determine the best position of glimpses, we confront this issue as reinforcement learning problem. We aim to obtain better classification accuracy by training a smart agent, which regulates how to glimpse the next position within a document and get better document representation. Our objectives are, to design a framework based on hierarchical attention mechanism and reinforcement learning for lengthy document classification, to classify the documents based on the designed framework and to analyze the performance of existing methods with the proposed method.

Our main contribution is the hierarchical recurrent attention learning framework with policy network. We train a smart agent in order to find the most distinguishing blocks of words for classifying the lengthy documents from coarse level to finer level that improves the classification accuracy. We conduct experiments on datasets; the results show that our model outperforms the baseline models.

The paper is arranged as follows: in section 2, we summarize the literature related to our work. Section 3 presents the proposed method framework and details of each of the module. Section 4 presents the classification. In section 5, we discuss about the experiments that is datasets and implementation details. Section 6 presents the results and discussion. Section 7 presents the conclusion and future work.

II. Related Work

Document Classification is essential to assemble documents for extraction, exploration, selection and annotation. Traditional methods used for feature extraction such as bag of words (BOW)[6], n-grams[7], term frequency-inverse document frequency (TF-IDF)[8] which represents the document as feature vectors. The classification algorithms such as Naïve Bayes [9], Support vector machine [10]; logistic regression [11] uses the feature vector as input and predicts the document class label as output. Though the traditional methods are simple and robust, but unable to capture the contextual information or the order of words which effects the classification accuracy, and also lengthy document have the problem of sparsity. The distributed word to vector (Word2Vec) [12] representation captures the semantic information of words.Word2Vec is an embedding model which is trained on the adjacent words over big corpus and employed by neural networks. GloVe [13] is another method of word embedding which is based on global matrix factorization and local context window. Both Word2Vec and GloVe generates word vectors with quiet differences. Recently, deep learning models [14] are developed for numerous text classification problems comprising news categorization, question answering, natural language inference, sentiment analysis, topic classification etc. which automatically extracts the features from input text. Convolutional Neural Networks (CNN) has achieved great success in the field of Computer Vision. CNN has also been applied to the field of natural language processing. For example, Xiang et al. [15] have applied ConvNets on characters, a sequence of characters are converted to some fixed size vectors. Analysis shows that the character level ConvNet is an implicit method for classification. But CNN is only suitable for short text classification. Recurrent Neural Networks (RNN) have been successfully applied for sequential data. RNN based models captures word dependencies. For example, Lai et al [16] have proposed a RNN model which bags the contextual information and constructs the textual representation using CNN. A variant of RNN called LSTM

[14] is used to capture long term dependencies in text. It also tackles the gradient vanishing and exploding problems. For example, Wang et al [17] have proposed a classification model based on CNN and LSTM which connects the extracted local features effectively and improves the classification accuracy. A sequence-to-sequence model comprises of an encoder-decoder architecture [18]. Attention has been successfully applied to NLP tasks. Attention mechanism [19] allows the model to pay attention to only specific parts of the input. Recently, Yang et al [5] proposed a hierarchical attention networks. It has two distinguishing characteristics: (1) a hierarchical structure which reflects the hierarchical structure of documents (2) two levels of attention (at both word and sentence). It performs remarkably better than previous models but cannot be applied to lengthy documents. Seq2seq models [20] experience the problem of exposure bias and instability between train/test estimation. These problems can be addressed by combining the Reinforcement Learning (RL) methods with seq2seq models that remembers long-term memories. Liu et al [21] proposed a new framework based on cognitive technique of reading called JUMPER, which represents the text classification as a sequential decision process. It has the property of making decision whenever the affirmation is sufficient, reduces the reading by 30-40% and achieves the better classification accuracy. Jun He et al [4] proposed a new approach for document classification which needs only the few glimpses of word blocks. It employs the RL agent along with recurrent attention learning. An agent is trained to control the consequent glimpses of locations in a long document to make the correct predictions. However, it is limited to only word representation. Distinct from existing methods, our paper represents document classification as a sequential decision approach. It is similar to the controller in Jun He et al [4] and hierarchical structure in Yang et al [5] which is not applicable to lengthy document. We propose a new decision making model which pays attention from coarse level to finer level in a document and is trained by reinforcement learning.

III. The Proposed Model

The proposed system is influenced by both recurrent attention model (RAM) [22] and hierarchical attention network (HAN) [5]. It employs a smart agent to regulate the attention for next glimpse part related to the context and policy that is learned through experience. It chooses the more appropriate words blocks that constitutes the attributes of the document. The overall framework of proposed model is shown in figure 1. It consists of various modules at both word level and sentence level. A document is divided into subsamples by choosing several word blocks from it.

Consider that a document consists of S sentences s_i, where each sentence consist of W_i words. Then, w_{it} depicts the words in the i^{th} sentence.

A. Glimpse Module

At each time interval, the glimpse module perceives a set of words identified and converted into word vectors through embedding layer at the location loc_t. A glimpse module is basically a convolutional neural network (CNN) which consists of convolutional layer and max pooling layer. The word vectors are mapped to get feature vectoro_g.

B. Word Level

This level consist of two parts: a word encoder and a word attention layer, which are recurrent module and attention module respectively. The word-level attention module consists of a PolicyNet which is basically a reinforcement learning agent. We represent the details of all distinct modules below:

I. Recurrent Module

At time step *t*, the recurrent module takes the feature vector o_g generated by the glimpse module as input, then combines o_g with its hidden state h_{t-1} to induce its next state vector h_t according to LSTM mechanism as shown below:

a) *LSTM based encoder*

Long Short Term Memory has three gates, they are input gate, output gate and forget gate. The forget gate looks at the previous hidden state h_{t-1} and input x_t. In the cell state, for each value it produces number between 0 and 1. f_t closer to 1 denotes retain the information. f_t closer to 0 denotes omit the information.

The input gate has two layers, they are a sigmoid layer and a tanh layer. First, a sigmoid layer decides which values will be updated and then a tanh layer creates a vector for new values, $\tilde{C}_t$ which could be added to state.

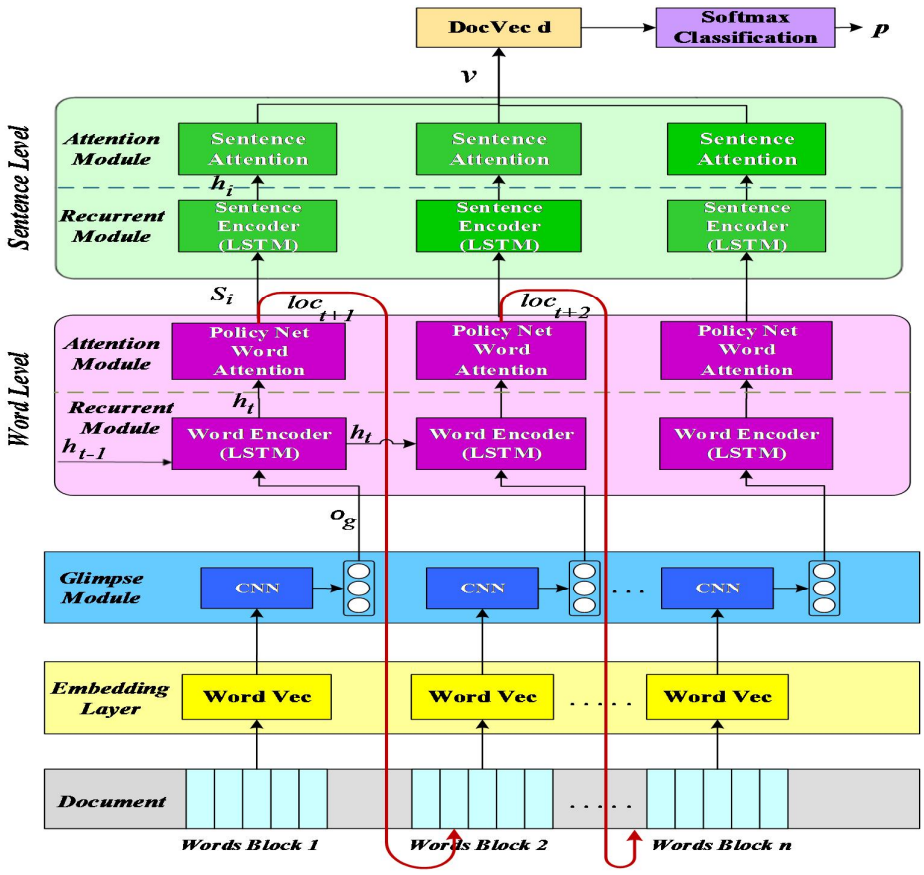

Figure 1: The Proposed Model Framework

Multiply both f_t and old state to forget the values. By adding $i_t \times \widetilde{C}_t$, a new candidate values are obtained, which decides on how much update in state value is required.Finally, the output gate decides what to output based on cell state, ouput is the filtered version. First, the sigmoid is executed. Then, the cell state is passed through tanh and multiplied with the ouput of the sigmoid, which leads to the next hidden state h_t according to LSTM as shown in Equation (1).

$$\begin{cases} f_t = \sigma_o\left(W_f \cdot \left[h_{t-1}, o_g\right] + b_f\right) \\ i_t = \sigma_o\left(W_i \cdot \left[h_{t-1}, o_g\right] + b_i\right) \\ \widetilde{C_t} = \tanh\left(W_c \cdot \left[h_{t-1}, o_g\right] + b_c\right) \\ C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \\ o_t = \sigma_o\left(W_o \cdot \left[h_{t-1}, o_g\right] + b_o\right) \\ h_t = o_g * \tanh(C_t) \end{cases} \quad (1)$$

a) Word Encoder

In word encoder, a BiLSTM is used to get word annotations by summing up information from both forward and backward word directions. Annotation represents the contextual information. The BiLSTM consists of forward LSTM, which scan the sentence s_i from first word to last word and backward LSTM scans from last word to first word as shown in Equation (2).

$$\begin{aligned} \vec{h}_{it} &= \overrightarrow{LSTM}\left(o_{git}\right), t \in [1, T] \\ \overleftarrow{h}_{it} &= \overleftarrow{LSTM}\left(o_{git}\right), t \in [T, 1] \\ h_{it} &= [\vec{h}_{it}, \overleftarrow{h}_{it}] \end{aligned} \quad (2)$$

For a given word w_{it}, an annotation is obtained by combining both the forward and backward hidden states. h_{it} sum up the whole sentence information surrounded by w_{it} .

2) Attention Module:

Here, attention mechanism is introduced to gather the informative words that are important to form a sentence vector.

a) Word Attention

It is basically a policy network called PolicyNet, where a deep reinforcement learning agent is employed to sample the word block of the document to be glimpsed next.

b) PolicyNet

A deep neural network (DNN) acts as a policy network which takes action by emitting the location loc_t after examining present glimpsed feature o_g and its hidden stateh_t. It utilizes the hard attention mechanism to extract important phrases or words in a lengthy document. It is designed as a partially observable markov decision process (POMDP) [23] including reward scheme. It is accompanied by RAM.

At each time interval, a policy network gains an observation s_t *(loc$_t$)* from the environment, which is the glimpsed feature o_g, then the policy network takes an action a_t by emitting the next location using the bellman's equation shown in Equation (3).

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \lambda max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3)$$

Where α is the learning rate, which determines how much previous Q-value retained for the new Q-value for the same state-action pair at a next time interval. λ is the credit assignment attribute which is assigned to states and actions.

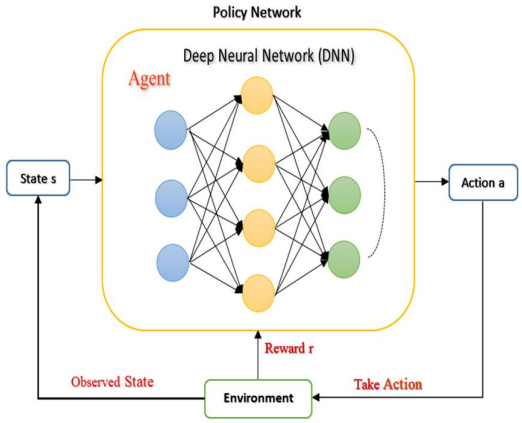

Figure2: Policy Network

Upon taking an action, the policy network acquires a latest observation s_{t+1} and a reward signal r_{t+1}. The intention of the policy network is to maximize the reward [24] defined as shown in Equation (4).

$$R = \sum_{t=1}^{T} r_t \quad (4)$$

At each time step, after obtaining the feature vector from the word encoder, we combine the representation of the factual words to construct a sentence vector. Particularly, as shown in Equation (5).

$$y_{it} = \tanh(W_w h_{it} + b_w)$$
$$\alpha_{it} = \frac{\exp(y_{it}^T y_w)}{\sum_t \exp(y_{it}^T y_w)} \quad (5)$$
$$s_i = \sum_t \alpha_{it} h_{it}$$

y_{it} is the hidden representation obtained by feeding h_{it} to one layer perceptron. The importance of the word is measured as the similarity of y_{it} with the context vector y_w, the context vector is the effective representation of informative words [25] and it is randomly adjusted and mutually learned through training. An importance weight α_{it} is obtained through softmax function. Subsequently, a sentence vector s_i is computed which is a summation of attention weights.

C. Sentence Level

This level consist of two parts: a sentence encoder and a sentence attention layer, which are recurrent module and attention module respectively.

1) Recurrent Module(Sentence Encoder)

After getting the sentence vectors s_i, the document vector is obtained in the same way. Here, sentences are encoded using BiLSTM as shown in Equation (6).

$$\begin{aligned} \vec{h}_i &= \overrightarrow{LSTM}(s_i), t \in [1, S] \\ \overleftarrow{h}_i &= \overleftarrow{LSTM}(s_i), t \in [S, 1] \\ h_i &= [\vec{h}_i, \overleftarrow{h}_i] \end{aligned} \quad (6)$$

The sentence annotation h_i is obtained by concatenating $\vec{h}_i$ and $\overleftarrow{h}_i$. It summarizes the surrounding sentences of sentence *i* but yet put attention on sentence i.

2) Sentence Attention

Sentences are indications of correct classification of a document. We again take advantage of the attention mechanism and initiate a context vector y_s at sentence level, which is used to quantify the significance of the sentences.

$$\begin{aligned} y_i &= \tanh(W_s h_i + b_s) \\ \alpha_i &= \frac{\exp(y_i^T y_s)}{\sum_i \exp(y_i^T y_s)} \\ v &= \sum_i \alpha_i h_i \end{aligned} \quad (7)$$

v is the document vector that represents the summary of the sentences as shown Equation (7). Likewise, the context vector at sentence level y_s is randomly adjusted and mutually learned through training.

IV. ALGORITHM

The pseudo code for the proposed system is given below:

Input: Document D

Output: The probability of predicting category for document D

1. **Initialize the module parameters [$\theta_{gm}, \theta_{rm}, \theta_{am}$] [$\theta_{gm}$ for glimpse module, θ_{rm} for recurrent module, θ_{am} for attention module], no._of_ glimpse G and maximum iterations Max.**
2. **for i=0,1,2,…Max do**
3. **for j=0,1,2,…G do**
    a. **Derive words around the location loc_t and use word embedding to obtain the word vectors.**
    b. **Use the glimpse module $f_{gm}(\cdot|\theta_{gm})$ which is basically CNN, to extract the feature vector o_g.**
    c. **Input o_g to the recurrent module $f_{rm}(\cdot|\theta_{rm})$ to obtain a new hidden state h_t.**
    d. **Predict the next location loc_{t+1} using attention module with reinforcement learning using h_t.**
4. **End for.**
5. **Obtain the representation S_i of glimpse with the input o_g in the recurrent module.**
6. **End for.**
7. **Obtain the representation d of document D with the input S_i.**
8. **Employ the softmax classifier on document representation d.**
9. **If predicted label is correct get reward=1 otherwise get none reward.**

V. CLASSIFICATION

The high level representation called document vector v is obtained after integrating all the sentence representations, which is used as input by softmax classifier for classification as shown in Equation (8).

$$p = softmax(W_d v + b_d) \quad (8)$$

After T time intervals, if the predicted category is correct, a positive reward *r=1* is given, otherwise reward =0 is given. Thus, our intention is to maximize the expected cumulative reward of the policy network by finding the optimal policy.

VI. Experiments

We determine the significance of our approach by conducting experiments on the collected dataset.

A. Data Sets

We gathered different classes of papers from arXiv dataset for long document classification. The papers comprising cs.AI, cs.CV, cs.DS, math.ST and math.GT, and the sum of documents per class is 2995, 2525, 4136, 3025, 3065 respectively as shown in Table I.
All pdf documents are converted to text format, then text files are preprocessed and kept only relevant words useful for classification. The documents were diminished to the first 5000 words.

B. Implementation details

1. Configuration

The propose model is implemented by Tensorflow 2.3 and were trained on NVIDIA Titan X GPU with 32 GB system memory.

2. Embedding

The word embedding used is GloVe. We choose to represent each word with 100-dimension due to memory constraints.

Table I: Data Interpretation: Total Number of Documents in Each Class and an Average Number of Words in Each Document

Class Name	Number of documents	Average words
cs.AI	2995	6212
cs.CV	2525	5630
cs.DS	4136	7439
math.ST	3025	6983
math.GR	3065	6642

3. Hyper parameters

Our proposed model is trained using the Adam's optimization technique. For the convolutional neural network, the kernel sizes ranging from 2 to 4, with 128 convolutional kernels are used. Each BiLSTM cell is set to 256. The learning rate is 0.001.

VII. Results and Discussion

A. Baseline Model

We utilize the proposed HAN model in the literature [5] as first baseline model. Another baseline model used is recurrent attention learning with reinforcement learning [4].We compare our model with these baselines by using the classification accuracy as performance metric.

B. Results Analysis

For each model, it is demonstrated that more words the model is exposed to, more the classification accuracy is. If we define window size as 40, then the total number of words considered for HAN model are 80, 160 and 320. For the second baseline model, the glimpses taken are 1, 2 and 4 glimpses which observe 80,160 and 320 words respectively. Our model also takes 1, 2 and 4 glimpses and observes 80,160 and 320 words respectively.
Table II show the classification accuracy obtained for the baseline models and our proposed model. The window size 20 is applied for all the models, from the below table , we can understand that the total number of words extracted is more, means more word blocks are sub-sampled which is related to number of glimpses taken. The experimental results demonstrate that the more number of words observed means the better classification

TABLE II: COMPARISON OF BASELINE MODELS WITH THE PROPOSED MODEL

Method	Observed Words	Glimpses	Accuracy
Baseline Model (HAN)	80	-	67.33
	160	-	72.52
	320	-	74.86
Baseline Model (Recurrent Attention Learning-RL)	80	1	75.44
	160	2	78.34
	320	4	79.31
Our Model (HAN-RL)	80	1	78.51
	160	2	80.34
	320	4	88.00

accuracy obtained. Because, for a given window size the words have local correlation and the Significance is substantial. Therefore, we can state that for fixed window size and varying glimpses, the observed words varies and gives the better experimental results.

C. Performance Analysis

The comparison of our proposed model (HAN-RL) and the baseline model is shown in the below figure 3.

Figure3: Classification Accuracy on 5-Class arXiv dataset

We can observe that the results of our model are better than the baseline model on the chosen arXiv dataset. By fixing the window size as 40, and varying the number of glimpses as 1, 2 and 4, the observed words varies and thus the classification accuracy differs substantially. It shows that the RAM mechanism applied for hierarchical attention mechanism is well trained and can locate the most significant group of words within a lengthy document.

VIII. CONCLUSION AND FUTURE WORK

This paper introduces a new model for classification of lengthy documents using convolutional neural networks (CNN), recurrent neural network (RNN) and hard attention mechanism. Our model focuses its attention from coarse level to finer level by employing a smart agent that is capable of locating the most discriminative blocks of words. Our model significantly improves the classification accuracy compared to the two baseline models

with less observed words. In our future work, we plan to apply different window sizes, glimpses and improve the policy network using optimization methods [26].

REFERENCES

[1] Hoa T. LE, Christophe Cerisara, AlexanderDenis" Do Convolutional Networks need to be deep for Text Classification", arXiv: 1707.04108v1, Jul 2017.

[2] Y. Kim. (2014). ''Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, 1746–1751.

[3] Chunting Zhou , Chonglin Sun , Zhiyuan Liu , Francis C.M. La ," A C-LSTM Neural Network for Text Classification", arXiv:1511.08630.[Online] http://arxiv.org/abs/1511.08630v2

[4] Jun He , Liqun Wang, Liu Liu, Jiao Feng, Hao Wu, "Long Document Classification From Local Word Glimpses via Recurrent Attention Learning", IEEE Access, Vol. 7(2019),pp. 40707-40718.

[5] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, ''Hierarchical attention networks for document classification,'' in Proc. HLT-NAACL, 2016, pp. 1480–1489.

[6] Hanna M. Wallach, " Topic Modeling: Beyond Bag-of-Words," in Proc. of the 23rd International Conference on Machine Learning, 2006.

[7] W. B. Cavnar and J. M. Trenkle, ''N-gram-based text categorization,''in Proc. 3rd Annu. Symp. Document Anal. Inf. Retr. (SDAIR), vol. 48113,no. 2, 1994, pp. 161–175.

[8] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong,Kui Lam Kwok,"Interpreting TF-IDF term weights as making relevance decisions," ACM Transactions on Information Systems,Volume 26,Issue 3,June 2008,Article No. 13,pp 1–37.

[9] I.Rish,"An empirical study of the naive Bayes classifier," IJCAI 2001 workshop on empirical methods in artificial intelligence , 3, page 41-46.

[10] Wen Zhang, Taketoshi Yoshida, Xijin Tang," Text classification based on multi-word with support vector machine," Knowledge-Based Systems, Volume 21, Issue 8, December, 2008 pp 879–886.

[11] Alexander Genkin, David D. Lewis, David Madigan,"Large-Scale Bayesian Logistic Regression for Text Categorization," Vol. 49, No. 3, Special Issue on Statistics in Information Technology (Aug., 2007), pp. 291-304.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

[13] J. Pennington, R. Socher, and C. Manning, ''Glove: Global vectors for word representation,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.

[14] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad Meysam Chenaghlu, Jianfeng Gao,"Deep Learning Based Text Classification: A Comprehensive Review," arXiv:2004.03705v3. [Online]. Available: https://arxiv.org/pdf/2004.03705

[15] Zhang, X.; Zhao, J.; Lecun, Y. Character-level Convolutional Networks for Text Classification. In Proceedings of the NIPS'15 28th International Conference on Neural Information Processing Systems, Montreal, QC,Canada, 7–12 December 2015; pp. 649–657.

[16] Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification.In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA,25–30 January 2015; Volume 333, pp. 2267–2273.

[17] Ruishuang Wang, Zhao Li, Jian Cao, Tong Chen, Lei Wang ," Convolutional Recurrent Neural Networks for Text Classification," IJCNN 2019. International Joint Conference on Neural Networks.

[18] Dzmitry Bahdanau, KyungHyun Cho Yoshua Bengio,"Neural Machine Translation By Jointly Learning To Align and Translate", ICLR 2015, 19 May 2016,pp.1-15.

[19] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, Varun Mithal,"An Attentive Survey of Attention ,"arXiv:1904.02874v1.[Online].https://arxiv.org/abs/1904.02874.

[20] Yaser Keneshloo ,Tian Shi , Naren Ramakrishnan , and Chandan K. Reddy," Deep Reinforcement Learning for Sequence-to-Sequence Models", 2019 IEEE.

[21] Xianggen Liu, Lili Mou, Haotian Cui, Zhengdong Lu, Sen Song," JUMPER: Learning When to Make Classification Decisions in Reading,"arXiv:1807.02314v1.[online]https://arxiv.org/abs/1807.02314.

[22] V. Mnih, N. Heess, and A. Graves, ''Recurrent models of visual attention,'' in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2204–2212.

[23] Matthijs T.J. Spaan, "Partially Observable Markov Decision Processes," Reinforcement Learning: State of the Art, Springer Verlag, 2012.

[24] Sai Krishna Gottipati, Yashaswi Pathak, Rohan Nuttall, Sahir, Raviteja Chunduru, Ahmed Touati, Sriram Ganapathi Subramanian, Matthew E. Taylor, and Sarath Chandar ,"Maximum Reward Formulation in Reinforcement Learning," arXiv:2010.03744v1. [Online]Available: https://arxiv.org/abs/2010.03744[2020]

[25] Sainbayar Sukhbaatar Arthur Szlam Jason Weston Rob Fergus," End-To-End Memory Networks," arXiv:1503.08895v5[Online]. Available: https://arxiv.org/pdf/1503.08895

[26] Snoek, J.;Larochell, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms.In Proceedings of the Advances in neural information processing systems, Lake Tahoe, NV, USA,3–6 December 2012; pp. 2951–2959.