Deep Learning in Text Summarization - A Survey

Athira S[*],SruthyManmadhan[**]
[*] PG Scholar, Department of CSE, NSS College of Engineering, Palakkad
[**] Assistant Professor, Department of CSE, NSS College of Engineering, Palakkad

Abstract- In this modern world dealing with enormous amount of data, text summarization plays a crucial role in extracting meaningful content and presenting precise, understandable information from large text. Many approaches to summarize text have been introduced over the years. Conventional methods create summary from text directly by extracting words that leads to redundancy and neglect document summary relationship. Deep learning techniques are proved to be effective in generating summaries. The paper focuses on deep learning based techniques for text summarization introduced over the years.

Index Terms- Text Summarization, Deep Learning, Auto Encoder, RNN, LSTM, GRU.

I. INTRODUCTION

The textual information available has been flooding with the increasing number of articles and links over the past few years making it difficult to search over the data to collect valuable information and present the information in a concise and clear way. Increase in data increases importance for semantic density thus there arise the need to recognize the most important things in the shortest amount of time. The generated summary helps to decide whether the textual content condensed in the article is relevant or not.

The idea of text summarization is finding a subset of the information to represent the entire document. Text summarization is taken as a task for condensing some textual information to a shorter version of itself which may contain all relevant and important information related to that document. It can be considered as a form of compression and hence suffer from information. Loss. Text summarization is effectively used in generating medical record summaries, weather data summary, news summaries etc.

Text summarization is broadly classified into the following categories:

1. Extractive text summarization: In this text summarization task objects are extracted from the documents without modification.
2. Abstractive Summarization: Different from extractive summarization as a word are modified, perform rephrasing or uses word that are not in the original document to create the summary hence making it more complex.

1.1. Deep Learning

Deep learning[1] is considered to be a type of representation learning method that uses cascade of multiple nonlinear processing units for performing transformations and feature extractions in such a way that output of one layer if feed as an input to next layer. Deep learning algorithms are capable of learning from the inputs in a supervised or unsupervised manner through multiple levels called as feature layers. The features layers are not described and designed by humans but are automatically learned from generalized learning process.

Conventional methods for text summarization includes directly extracting words from the textual content to represent the summary, Text summarization include removing stop words, identifying noun groups, lemmatization etc. The major disadvantage of conventional methods is that the summary generated may contain redundant words. As there are no record of the words that are already been selected it is possible that words may repeat itself in the summary as it does in the main text. Also, in conventional methods the relation between the summary that are generated and the document is very low. Thus, making it difficult for the users to have clear understanding of the document from the summarized content. Thus to overcome the disadvantages Deep learning techniques are employed to text summarization.

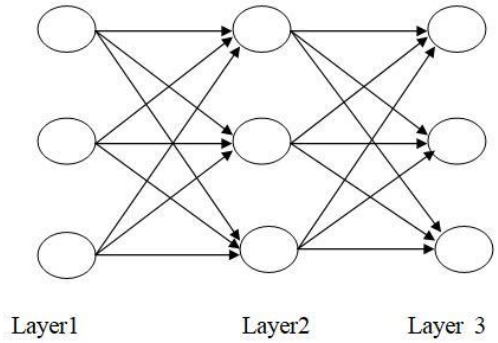

Figure 3: Three layered neural network with one input layer,one output layer and one hidden layer

The paper is organized as follows: Section II explains the various models of deep learning that are employed in text summarization and Section III compares various text summarization techniques using Deep learning models. Section IV discusses evaluation techniques. Section V pro-vides the conclusion and VI the references.

II. MODELS IN TEXT SUMMARIZATION

Various deep learning models are used in text summarization some of them are explained below:

2.1Auto Encoder

Auto encoder is an unsupervised learning algorithm that is used to learn data coding efficiently. Auto encoder aims at learning the representation of a set of data in order to reduce its dimensionality as well as complexity. The auto encoder consists of three layers that are the input layer, the encoding layer and the decoding layer (output layer). The input and the output layer are basically the same in such a way that the algorithm learns to compress the input data into an encoded format in the encoding layer which is later decoded to its original self by decoding layer. Auto encoders perform dimensionality reduction by compressing the input data by removal of noises.

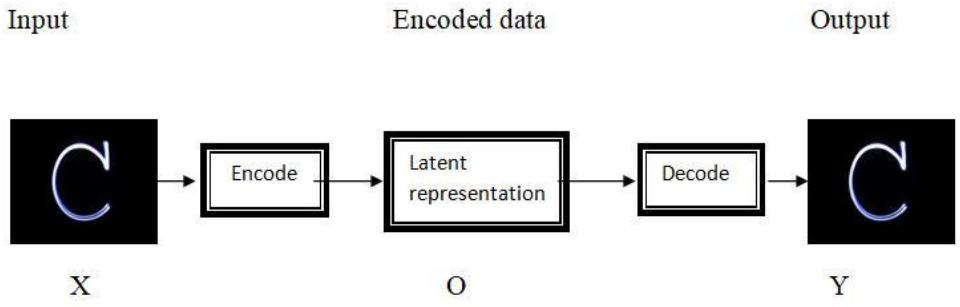

Figure 4: Basic Architecture of Auto Encoder [2]

For each input X given at the input layer, the input is encoded into an latent representation O such that O = f(X) and at the output layer the encoded representation is reconstructed as Y=g(O).

2.2. **Recurrent Neural Network**

Recurrent neural network [11] belongs to the class of artificial neural network that are represented using graphical models. Nodes belong to the part of directed graph along a sequence that allows the exhibition of temporal dynamic behavior. In traditional neural network input and output are considered to be independent of each other thus it does not take into account the previous information. In feed forward neural network [12] the connections between nodes never make up a cycle, information moves in one direction from input states through hidden states to output state. Also, the outputs are independent of each other such that output at time step t is not dependent on the out of time step t-1. In a scenario such as predicting the next word in a sentence feed forward network are observed to be inefficient. In RNN, the internal states can be used to process a sequence of input through back propagation. RNN are found to be effective where the input and outputs are dependent on each other.

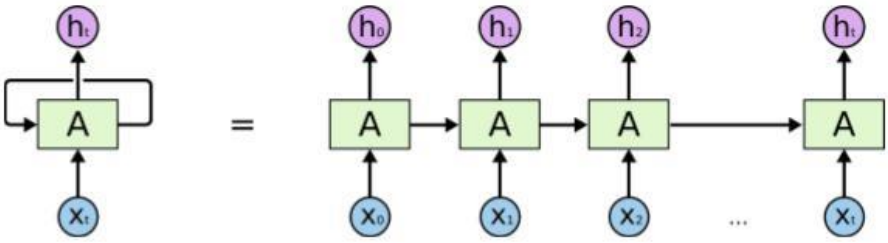

Figure 5: Recurrent Neural Network Unit [13]

Here in each time step t, the RNN unit takes the input vector x_t and the hidden state vector h_{t-1} to form the output of the hidden state h_t the process is thus repeated until all the inputs are processed.RNN can be formulated as a function given as:

$$h_t = f(x_t, h_{t-1}) \quad (1)$$

2.2.1. Limitation in RNN

The major drawback of RNN are termed as vanishing gradient problem[14] and exploding gradient problem[15].While performing back propagation in RNN it tends to calculate the error that is square of the difference between the actual output and the output observed from a model. With the calculated value for error the weight function for the next time step is calculated as:

$$W = n\frac{\partial e}{\partial w} \quad (2)$$

where w is the change in weight, $\frac{\partial e}{\partial w}$ is the rate of change of error with respect to weight that is called the gradient and n is the learning rate. Thus from this the new weight is calculated by adding w to the old weight. Now, if the value of the gradient becomes very small than 1 then Δw becomes negligible resulting in no greater difference in the weight calculation for the next time step. This is referred to as the Vanishing gradient problem .Similarly, If the gradient value becomes too large and there are long term dependencies then in each time step the weight value increases drastically. This is referred to as Exploding gradient problem and can be solved by using truncated BTT[16] or by clipping gradient at threshold. As for the former, the solution is to use LSTM and GRU.

2.3. Long Short Term Memory (LSTM)

LSTM's [24] are a special form of RNN that are capable of learning long term dependencies. In some scenarios, only recent information are required to perform a given task such as language models trying to predict the last word in a sentence .In situations where the gap between relevant information and the place where it is needed is small RNN learns to use the past information without the occurrence of problems discussed earlier But there are cases where more context are needed where the gap between relevant information and where it is needed are large. In such cases LSTM network are proven to be efficient.

LSTM has a chain like structure where the repeating module has four interacting modules that are the cell state, output gate, update gate and the forget gate. The cell state is represented using the horizontal running layer in the LSTM repeating module and act as a convener belt with minimal interactions. The basic structure of an LSTM module is given as follows:

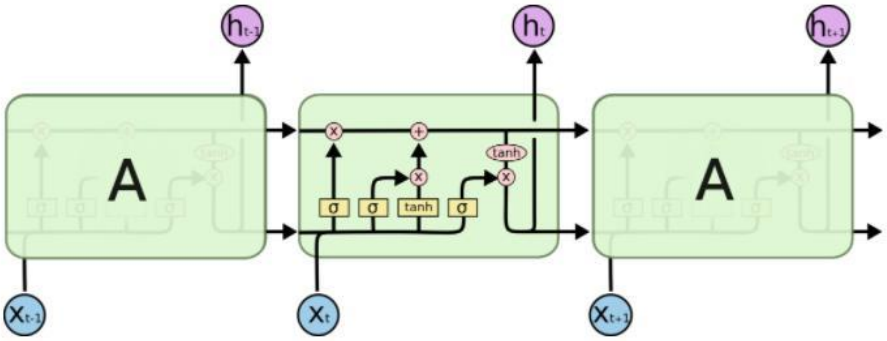

Figure 6: Basic Architecture of LSTM [25]

2.3.1 Forget Gate

The forget gate identifies the information that need to be eliminated from the cell state so that only relevant information are taken forward. The gate take into account the output from the previous time stamp h_{t-1} the new input $x_t$and produces an output between 0 and 1 for each cell state describing whether to keep the state or not. The following equation defines the sigmoid function carried out by the forget gate.

$$f_t = \sigma\ (w_f\ [h_{t-1}, x_t];\ b_f\ ) \qquad (3)$$

2.3.2. Update Gate

Update gate decides on what information to be stored in the memory cell. The gate works in two layers, one of which is the input sigmoid layer that decides upon the value to be updated and an tanh layer that creates the vector of the new candidate value $\widetilde{C_{t-1}}$ that can be added to the states later.

At first the sigmoid function take into account the input coming from the precious time stamp $h_{t\ 1}$ and the new input x_t to calculate the value of i_t as:

$$i_t = \sigma\ (w_i\ [h_{t-1}, x_t];\ b_i\ ) \qquad (4)$$

and also the inputs are passed through the tanh layer to create $\widetilde{c_t}$ as follows:

$$\widetilde{C_{t-1}} = tanh\ (w_c\ [h_{t-1}, x_t];\ b_c\ ) \qquad (5)$$

now the values of i_t , f_t and are used to update the value of new cell state c_t as :

$$c_t = f_t * c_{t-1} + i_t * \widetilde{C_t} \qquad (6)$$

2.3.3. Output Gate

Output gate contains sigmoid layer that decides what parts of the cell state is given to the output. The cell state is made to pass through a tanh activation function to push its values between -1 and 1. After activation, the cell state is multiplied with the output of sigmoid layer to decide the output. The mathematical representation is given by the two equations:

$$o_t = \sigma\ (w_o\ [h_{t-1}, x_t], b_o\ ) \quad (7)$$

$$h_t = o_t * \tanh(c_t) \quad (8)$$

The major disadvantage with an LSTM network is that there are lots of operations performed within a single repetitive unit. Which when considered for a big network, the training process consumes greater amount of time. In order to overcome this limitation GRU were introduced.

2.4. Gated Recurrent Unit

Unlike LSTM, GRU does not account to any cell state it uses two different gates namely reset gate and an update gate. The update gate decides how much of the past information needs to passed on to the future states. Where, Reset gate decides in how much past information to forget. These two gates account to the output produced by the repetitive unit in each time step.

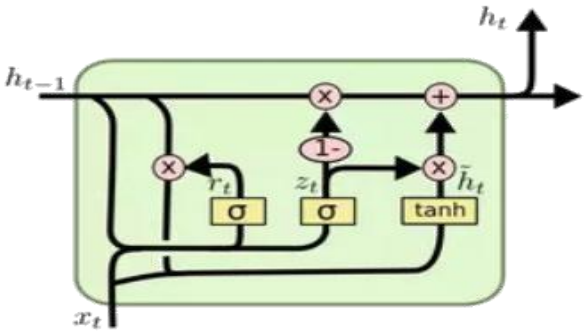

Figure 4: Basic Architecture of GRU [25]

III. COMPARISON

Table 1: Comparison of Text Summarization Techniques Using Auto Encoder

Paper	Pros	Cons
[3]	Local representation reduces sparsity and multiple runs improve efficiency.	Only works well with small vocabulary and mostly support query based search.
[4]	More descriptive feature space and improve recall on average.	Under performed system, need improvement in accuracy.
[5]	Account to summarization and	Computational needs are high. Relation

	reconstruction of text and aims at efficient semantic representation of variable size text.	between summary and document are poor.
[6]	Uses attention model to reduce redundancy.	Needs improvement in performance.
[7]	Improves summary quality and outperforms state-of-art models.	Not proven to be efficient.
[8]	Document–Summary pair training and more interpreted summaries	Sometimes fails to arrange word in correct order
[9]	Achieves state-of-art performance in benchmark datasets and better internal representation	Requires improvement in terms of sentence segmentation representation
[10]	Splits sentences based on position and is a hierarchical model	Fails to outperform baseline attention decoder

Table 2: Comparison of Text Summarization Techniques Using RNN

Paper	Pros	Cons
[17]	Structurally simple and performs end to end training.	Efficient alignment and consistency in generation are challenges.
[18]	Provides state-of-art performance and promising results.	Factual data incorrectly reproduced and replace uncommon words with alternatives.
[19]	Adress the modeling issue of preserving meaning and key content.	Doesnot account to previous information
[20]	Trained on human generated reference summaries.	Less Rouge value.
[21]	Reduces inaccuracy and repetition and out performs state-of-art model	Performance and high level abstraction needs to be achieved
[22]	Captures notations of salience and repetition .Easily interpretable.	Structured summarization problem.
[23]	Semi supervised technique outperforms standard seq2seq.	Less accurate

Table 3: Comparison of Text Summarization Techniques Using LSTM &GRU

Paper	Pros	Cons
[26]	Capture compositionality better without complex Architecture.	Timescale constant can be optimized further.
[27]	Efficient document summary scoring	Requires large scale training corpus.
[28]	Computational efficiency is prominent and has good Accuracy	Redundant information.
[29]	Generate natural sentences	Training is time consuming and also determining semantic similarity between phrases is difficult

IV. EVALUATION METHOD

Recall-Oriented Understudy for Gisting (ROUGE) [30] is a evaluation method for text summarization it automatically determines a summary quality by comparing it with ideal summaries created by humans called the gold standards. The measure determines the count of overlapping unit between the generated and the ideal summaries. There are four different Rouge measures described as follows:

1. ROUGE-N :N-Gram co-occurrence statistics It is a n-gram recall calculated between the generated candidate summary and the set of referenced summaries.
2. ROUGE-L: Longest Common Subsequence Given two sequence P and Q , the longest common subsequence of P an Q are the common subsequence with the maximum length.
3. ROUGE-W: Weighted Longest Common Subsequence Improves ROUGE-L values by remembering the length of the consecutive matches that are encountered so far.
4. ROUGE-S: Skip-Bigram Co-Occurrence Statistics Calculates the overlapping Skip-Bigram between the generated summaries and the set of referenced summaries

V. CONCLUSION

The paper summarizes the various model of deep learning that are employed in text summarization process and also the techniques that are developed over the years. It is observed Auto encoders, RNN and GRU are three main models that are widely employed in text summarization process and are proven to be more efficient than conventional text summarization methods.

ACKNOWLEDGMENT

This research was supported by NSS College of Engineering,Palakkad. We are thankful to our colleagues who provided expertise that greatly assisted the research, although they may not agree with all of the interpretations provided in this paper.

REFERENCES

[1] G. E. Hinton, S. Osindero, Y.-W. Teh, *A fast learning algorithm for deep belief nets*, Neural computation 18 (2006) 1527-1554.

[2] F. Li, H. Qiao, B. Zhang, *Discriminatively boosted image clustering with fully convolutional auto-encoders*, Pattern Recognition 83 (2018) 161-173.

[3] M. Y. Azar, K. Sirts, L. Hamey, D. M. Aliod, *Query-based single document summarization using an ensemble noisy autoencoder*, in: Proceedings of the Australasian Language Technology Association Workshop 2015, pp. 2-10.

[4] M. Youse -Azar, L. Hamey, *Text summarization using unsupervised deep learning*, Expert Systems with Applications 68 (2017) 93-105.

[5] B. Oshri, N. Khandwala, *There and back again: Autoencoders for textual reconstruction*, 2016.

[6] P. Nema, M. Khapra, A. Laha, B. Ravindran, *Diversity driven attention model for query-based abstractive summarization*, arXiv preprint arXiv:1704.08300 (2017).

[7] P. Li, W. Lam, L. Bing, Z. Wang, *Deep recurrent generative decoder for abstractive text summarization*, arXiv preprint arXiv:1708.00625 (2017).

[8] Y.-S. Wang, H.-Y. Lee, *Learning to encode text as humanreadable summaries using generative adversarial networks* (2018).

[9] S. Ma, X. Sun, J. Lin, H. Wang, *Autoencoder as assistant super-visor: Improving text representation for chinese social media text summarization*, arXiv preprint arXiv:1805.04869 (2018).

[10] Y. Zhang, Y. Wang, J. Liao, W. Xiao, *A hierarchical attention seq2seq model with copynet for text summarization*, in: 2018 International Conference on Robots & Intelligent System (ICRIS), IEEE, pp. 316-320.

[11] S. Grossberg, *Recurrent neural networks*, Scholarpedia 8 (2013) 1888.

[12] G. Bebis, M. Georgiopoulos, *Feed-forward neural networks*, IEEE Potentials 13 (1994) 27-31.

[13] M. T. Nayeem, et al., *Methods of sentence extraction, abstraction and ordering for automatic text summarization*, Ph.D. thesis, Lethbridge, Alta.: Universtiy of Lethbridge, Department of Mathematics and Computer Science, 2017.

[14] S. Hochreiter, *The vanishing gradient problem during learning recurrent neural nets and problem solutions*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6 (1998) 107-116.

[15] R. Pascanu, T. Mikolov, Y. Bengio, Understanding *the exploding gradient problem*, CoRR, abs/1211.5063 (2012).

[16] R. J. Williams, J. Peng, *An efficient gradient-based algorithm for on-line training of recurrent network trajectories*, Neural computation 2 (1990) 490-501.

[17] A. M. Rush, S. Chopra, J. Weston,*A neural attention model for abstractive sentence summarization* ,arXiv preprint arXiv:1509.00685 (2015).

[18] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., *Abstractive text summarization using sequence-to-sequence rnns and beyond*, arXiv preprint arXiv:1602.06023 (2016).

[19] G. Rossiello, *Neural abstractive text summarization*., in: DC@ AI* IA, pp. 70-75.

[20] R. Nallapati, F. Zhai, B. Zhou, *Summarunner: A recurrent neural network based sequence model for extractive summarization of documents*., in: AAAI, pp. 3075-3081.

[21] A. See, P. J. Liu, C. D. Manning, *Get to the point: Summarization with pointer-generator networks*, arXiv preprint arXiv:1704.04368 (2017).

[22] R. Nallapati, B. Zhou, M. Ma, *Classify or select: Neural architectures for extractive document summarization*, arXiv preprint arXiv:1611.04244 (2016).

[23] C. Khatri, G. Singh, N. Parikh, *Abstractive and extractive text summarization using document context vector and recurrent neural networks*, arXiv preprint arXiv:1807.08000 (2018).

[24] H. Sak, A. Senior, F. Beaufays, *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*, in: Fifteenth annual conference of the international speech communication association.

[25] C. Olah, *Understanding lstm networks*, GITHUB blog, posted on August 27 (2015) 2015.

[26] M. Kim, M. D. Singh, M. Lee, *Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization*, arXiv preprint arXiv:1607.00718 (2016).

[27] Y. Hou, Y. Xiang, B. Tang, Q. Chen, X. Wang, F. Zhu, *Identifying high quality document-summary pairs through text matching*, Information 8 (2017) 64.

[28] Y. Zhang, J. Liao, J. Tang, W. Xiao, Y. Wang, *Extractive document summarization based on hierarchical gru*, in: 2018 International Conference on Robots & Intelligent System (ICRIS), IEEE, pp. 341-346.

[29] S. Song, H. Huang, T. Ruan, *Abstractive text summarization using lstm-cnn based deep learning*, Multimedia Tools and Applications (2018) 1-19.

[30] C.-Y. Lin, *Rouge: A package for automatic evaluation of summaries*, Text Summarization Branches Out (2004).