Journal on Big Data
DOI:10.32604/jbd.2021.015954
Retraction

Tech Science Press

RETRACTED: Recent Approaches for Text Summarization Using Machine Learning & LSTM0

Neeraj Kumar Sirohi[1,*], Mamta Bansal[1] and S. N. Rajan[2]

[1]Shobhit Institute of Engineering & Technology, Meerut, 0091121, India
[2]IMS Engineering College, Ghaziabad, 0091120, India
[*]Corresponding Author: Neeraj Kumar Sirohi. Email: neerajsiroh@gmail.com
Received: 03 September 2020; Accepted: 20 December 2020

Abstract: Nowadays, data is very rapidly increasing in every domain such as social media, news, education, banking, etc. Most of the data and information is in the form of text. Most of the text contains little invaluable information and knowledge with lots of unwanted contents. To fetch this valuable information out of the huge text document, we need summarizer which is capable to extract data automatically and at the same time capable to summarize the document, particularly textual text in novel document, without losing its any vital information. The summarization could be in the form of extractive and abstractive summarization. The extractive summarization includes picking sentences of high rank from the text constructed by using sentence and word features and then putting them together to produced summary. An abstractive summarization is based on understanding the key ideas in the given text and then expressing those ideas in pure natural language. The abstractive summarization is the latest problem area for NLP (natural language processing), ML (Machine Learning) and NN (Neural Network) In this paper, the foremost techniques for automatic text summarization processes are defined. The different existing methods have been reviewed. Their effectiveness and limitations are described. Further the novel approach based on Neural Network and LSTM has been discussed. In Machine Learning approach the architecture of the underlying concept is called Encoder-Decoder.

Keywords: Text summarization; extractive summary; abstractive summary; NLP; LSTM

1 Introduction

Text summarization is a method for extracting the utmost features of a text, compile and assemble them into a brief summary of the main document [1]. According to Mani et al. [2], text summarization is the procedure of extracting the utmost vital information from a document to generate a reduced form for a specific document for user. Another scholar [3] describe summary as "Summaries are typically roundabout 17% of the original document and also hold everything that the critical information or key idea of document is also preserve". Summarization is an effective and powerful method to produce an abstract of the entire data. Mainly there are two categories of summaries namely, extractive summary and abstractive summary. The abstractive summary is a recent concept under great research; but unfortunately, still, no algorithm has been attained to get good result. These summaries are derived from text after learning what was spoken in the object and then altering it into a form articulated by the machine. It is done just like human who generate summaries after reading the article. On the other hand, extractive summary is generated after picking the important words and sentences from the original text and arranging them before presenting it to the reader.

We get summary by selecting important keywords which describe the text. The process of picking the phrase and words from the document that can describe the core idea (sentiments) of the source(document) without any human intervention depending on the model is known as Automatic keyword extraction [4].

In this paper, we have explained the machine learning approach that uses ANN (Artificial Neural Networks) to produce summaries of random length document. Precisely, the Encoder-Decoder RNN (Recurrent Neural Network) architecture established for machine version has found out to produce auspicious results when it is used for the problem of text summarization. The model involves two neural networks working in simultaneously parallel–the encoder, that accepts the input order and generate a vector output then the decoder that takes the prior vector output produced by encoder as its input and produces the concluding output sequence. Initially the paper reviews an overall outline of the approaches for text summarization; and then details on the Encoder-Decoder model of the machine learning method along with its execution using TensorFlow in Keras.

2 Related Work

Although the study on the Automatic Text summarization has been started long back the initial effort was made by [4] in 1950's at IBM Laboratories. This approach picks important sentences from the article and concatenates them together. Here term frequency is used to measure the sentence. Sentences are involved in the summary if the term frequency of that particular sentence is high. Later [5] proposed a graph based ranking model for text processing which produced improved and extra influential results.

2.1 Text Summarization Process

Primarily the approaches of text summarization can be categorized into different categories, mainly known as statistical based, machine learning based, coherent based, graph based, and algebraic based (Fig. 1).

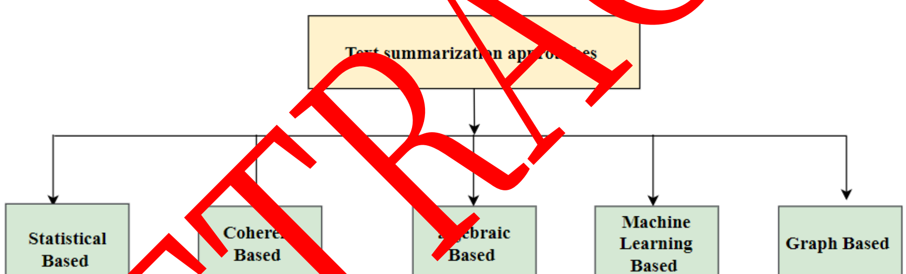

Figure 1: Text summarization methods

Under each category, various sub techniques are there. Our focus here is on the technique of Machine learning along with proposed architecture of Encoder-Decoder based LSTM neural network.

Machine learning approach is based on feature dependent and for that we need annotated dataset to train the models and then test the model. There are numerous good machine learning approaches namely Decision Tree, SVM (Support Vector Machine), Bayesian Model of Summaries, Hidden Markov Model (HMM), methods based on Fuzzy Logic and Neural Networks Based Summarization.

2.1.1 Decision Tree

One of the most common and usually used inductive learning approach is known and Decision tree algorithms [6–7]. the C4.5 algorithm [7] is selected for summarizer training. By searching and selecting the features that has produced the utmost information decision tree is generated then a node of tree is created by using a set of rules equivalent to the feature. Until there is no further addition in the gained information this process is repeat for other sentences as well various times. In testing, a pattern is repeatedly compared with a node of a decision tree starting from the root and following appropriate branches based on the condition and feature value until a terminal node is reached. C4.5 has been recognized an extremely quick and skilled algorithm with decent simplification capability.

In this category SUMMARIST is a method to produce a durable computerized text summarization scheme. It follows the 'equation':

summarization = topic understanding + topic identification + generation

Here the 3 phases are:

Topic Identification: Recognize the utmost significant (central) areas of the document [8]. SUMMARIST uses location reputation [8–9], term occurrence and cue idioms [9–11]. Later on, the Reputation constructed on discourse structure will be added [12]. This is the extremely advanced phase of SUMMARIST.

Topic Understanding: To mixed ideas like as menu, food and water into unique universal thought restaurant, our necessity is further to the modest word, in conventional data salvage the aggregation is used. Here the approach of concept counting [13] and topic signatures [14] has been used to deal with the fusion issue.

Summary Creation: SUMMARIST is capable to produce summaries from numerous arrangements such as keywords (significant noun idioms), featch (vital sentences in novel document), pattern based summaries [15] (collected from pre-specified patterns), and polished summaries (created by a sentence plotter and identifiers) [16–17].

2.1.2 Support Vector Machines Approach

SVMs is a supervised learning system for dual class difficulties. Fig. 2 display the abstract construction of SVM. Training data is inputted by (a1, b1), …, (az, az), $a_i \in$ Rn, bj $\in \{+1, -1\}$. Here, a_i denotes characteristic vector of the i-th example, b_i is its class ticket, positive (+1) or negative (−1).

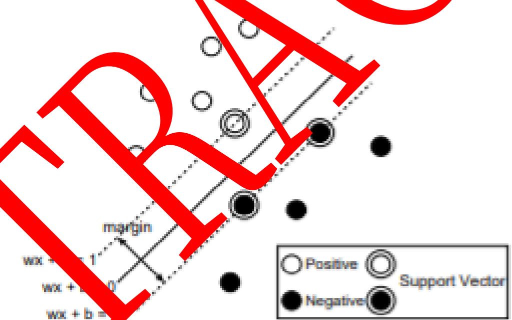

Figure 2: SVM approach for text summarization [18]

SVM is based on fature selection that are associated with sentence Si some important features are

Sentences Position: Location of sentence play a vital role for selecting the sentence as summary sentence. Selection of sentence is very important as because we know that Sentences in the opening express the main idea of text whereas the final sentence is concluded or summary sentence.

The sentence is assessed through its position in the document. Position of sentence provides the weight of sentence. If the position is in the first 5, the feature score is evaluated by [19–20]. Considering first 5 sentences in the paragraph, the score would be:

Score = 5/5 for 1st, 4/5 for 2nd, 3/5 for 3rd, 2/5 for 4th, 1/5 for 5th, 0/5 for other sentences.

After some time [21] introduced the different and modest method used to evaluate position of the sentence.

Score = 0, if sentence position is in the central of passages in the document,

Score = 1, if sentence position at the starting at the document,

Score = 1, if sentence position is the end in the text.

Sentence Length: second important feature is length of sentence. Usually, too long and too small sentences are not fit for summary. Very long will have redundant information and too short sentence does not provide ample information about the text.

Weight of Sentence: weighting of sentence is achieved through two steps. First step is to clean the text by eliminating the stop words then a weight is assigned to individually term. The weight is measure as follows:

The weight,

wh = occurrence of the term/Entire no. of terms in the text

After fixing the weight of individually term, then in second step according to their weight a rank is assigned. After that by summing up the weights of every term of sentence, and divided this sum by entire no of term in the sentence then weight of every sentence is determine, i.e.,

whs = (wh$_i$)n$_i$ = 1/n

where whs = sentence weight.

wh1, wh2, wh3, ... whn = weights of distinct terms.

n = complete amount of terms in that sentence.

Sentence Similarity to Title: According to this property characteristic of sentence that holds the word which present in the heading is assign more weighted and have greater chances to be consider as summary sentence. These sentences are selected through using the title of the text as a "query" against all the sentences of the text; then by using cosine similarity [22] measure the likeness of the title and every sentence from the text.

Sentence-to-Sentence Cohesion: This characteristic is selected as follows: for individual sentence s we initially calculate the likeness between every new sentence and ***s*** from the text; then we sum up all equal values which is obtained from the verdant (raw) significance of individual attribute for s; the procedure is recurring for each and every phrase. The controlled significance (in the range [0, 1]) of this characteristic for a phrase s is gained by calculating the relation of the raw characteristic significance for ***s*** over the prime verdant feature phrase between all phrase in the text. importance nearer to 1.0 represent sentences with Great cohesion.

2.1.3 Bayesian Model of Summaries

In extractive summarization, the consideration of ranking sentences is based on how vital/significant they are as components of summary. Bayesian believe in an exclusive ranking method which is consider sentence likelihood and under a given DOV (distribution of votes) it is an element of summary, i.e.,

$$P(s|y) \tag{1}$$

where s denotes a given sentence, and $y = (y_1, \ldots, yn)$ represents distribution of votes, an array of experiential sum of the votes for sentences in the document; y_i mentions the amounts of votes for a sentence at the opening location of text, y_n to that for a sentence occurring at the second place, etc.

Either with BIC (Bayesian Information Criterion) or with MC (Monte Carlo integration method (MacKay, 1998), generating a summarizer on it is an equally simple concern. Specified text **t** and a summarization rate r, summarizer basically assign a rank to each sentence from **t** based on $P(s|y_i)$ and select an **r** section of uppermost status sentences.

providing a training set of texts with manually selected document, prepare a groupage function that guesses the likelihood of a specified text which is a part of a summary. Then novel summary is produced by providing the weight to the sentences based on this likelihood and then highest scoring sentences are selecting as summary sentence. For individual sentence '***s***' calculate the likelihood and comprised with a summary S specified the **n** characteristic y$_x$; x = 1…n, which can be explained through Bayes' law as follows [23]:

$$\frac{P(s\in S|y_1,y_2,\ldots,y_k)=P(y_1,y_2,\ldots,y_k|s\in S)P(s\in S)}{P(y_1,y_2,\ldots,y_k)} \tag{2}$$

Assuming arithmetical individuality of the features:

$$\frac{P(s\in S|y_1,y_2,\ldots,y_k)=\prod_{x=1}^{k} P[y_j|s\in S]\;P(s\in S)}{\prod_{x=1}^{k} y(f_j)} \tag{3}$$

2.1.4 Hidden Markov Model

Hidden Markov Model (HMM) [24] is an alternative approach to select a sentence from the text. HMM have fewer assumptions for selecting a sentence in compression with Bayesian method. Actually, the HMM (Hidden Markov Model) do not imagine that the chances of selection of sentence ‘I’ as the part of summary, is basically independent from i-1 sentence in the summary. Three features were used In Hidden Markov Model for selecting a sentence:

- Position of the Sentence in the text,
- Total terms in the sentence,
- Similitude between given text terms and the sentence terms.

The Hidden Markov Model has 2s + 1 states, alternating with summary ‘s’ and non-summary ***s+1***. Below, the Fig. 3, is an example of HMM demonstrations with 7 nodes, agreeing to s = 3.

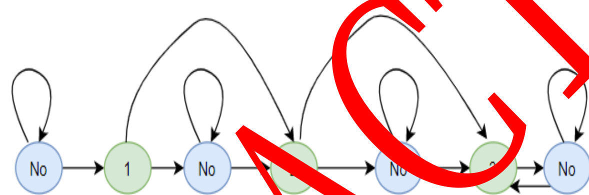

Figure 3: HMM model for summarization [24]

This chain is intended to select up to s-1 summary and randomly no of supporting statements. Each and every path through the chain sight each first s-1 summary states. The first two states in the chain allow arbitrary no of non-summary and summary sentences. This Markov chain has 2’s total permitted constraints which helps in defining the probability of different evolutions between pairs of state. These parameters are assessed on the basis of training data. For example, the evaluation of likelihood between summary states 2j and 2j+2 summary state is the no of times the summary sentence j+1 directly followed by summary sentence J, in the training data. And the probability of changes between summary state 2J and non-summary state 2J+1 is defined to be one less this likelihood.

Then calculate the maximum probability for each and individual by using this computation, and then create a transaction matrix M for our Markov chain, where [i, j] values in matrix denotes the predictable probability of transition state i to j.

In the same way we also calculate p(i) the maximum likelihood evaluation of the primary distribution for the chain by using the following equation

P(i) = pr (the initial sentence corresponding to state i)

where p(i) = o for i > 2 since the initial sentence is also the first summary sentence (state 2) or a state that leads the initial summary sentence (state 1). After performing little improvement in the chain that permits us to mining an accurately S summary sentence. This improved chain shown in below Fig. 4, which is differ from above chain of Fig. 3. This improved chain removes the cycle which exists between last summary and non-summary states. This chain is utmost suitable for processing fixed length summary. It has 2s free constraints to be projected from training state i as an output function

$$B_i(o) = pr(0|state\ i) \tag{4}$$

Here o is the experimental vector of features related to a sentence.

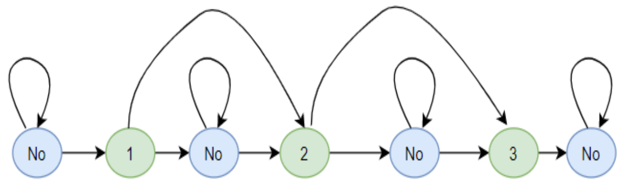

Figure 4: HMM model for summarization-1 [24]

2.1.5 Neural Networks Based Text Summarization

ANN (Artificial Neural Networks) is unmatched and the utmost common and influential category of machine learning systems. Artificial Neural Networks is used to produce summaries of random size articles. Generally, an article database is used to trained neural network. Then improved the neural network, by combination a generate summary with furthermost graded sentences of the text. The network fixes the weight of numerous characteristics used to selecting the summary-impotence of separate sentence by using feature fusion [25]. ANN has two stages that is training stages and testing stages. In training stage, the neural network studies the patterns and characteristics of sentences that are the part of summary and those that should not be consider as summary sentence. In a classical model of Neural Network structure, it has 7 input layer neurons and three feed-forward layer, single output layer neurons and 6 hidden layer neurons. Every sentence is denoted with the help of a vector [f1, f2, f3, …, f7] which contains of seven characteristics (Fig. 5). The features are carefully chosen considering to location of text or location of the sentence.

f1 = Title (Paragraph Position) followed by Paragraph.

f2 = Position of Paragraph in document

f3 = Position of Sentence in paragraph

f4 = Paragraph's initial Sentence

f5 = Dimension of the Sentence

f6 = Count of signification arguments in sentence

f7 = Sentence Title word

There are basically three phases in text Summarization process using neural network mainly these steps are: Training step, feature fusion step and sentence selection phase. The neural network is trained in initial phase and capable to identify the sentence type that essentially be a part of summary. Then try to minimize the neural network and crash down the secreted layer element activations into distinct significance with occurrences. After that finally sentence selection is done through this trained neural network and filtering of document also picking the highly graded sentences is perform in this third phase [25].

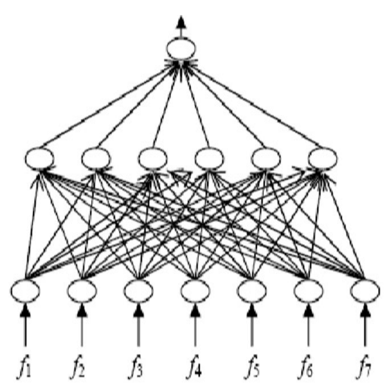

Figure 5: The neural network after [26]

By using any appropriate clustering method, a cluster is created in which every children layer neurons and activation value of secrete layer is consider. Every cluster is recognized by its centroid and occurrence. The activation value of individually hidden layer neuron is exchanged by the centroid of the cluster. The grouping of these two phases links to simplifying the properties of characteristics, as complete, the grouping of these two phases links to simplifying the properties of characteristics, as complete, and generate effective argument for sentence grade. One more approach described by [27] used Neural Network for document summarizing used "Mathematical Information Feature" for input list of features so the network usages input of eight neurons. Subsequently discovery advanced graded essence (summary) sentences by neural network pass these sentences to linguistic construction to discovery the recitation, construction from that, and discover linguistic relation in sentences which might assistance in the process of discover improved summary sentences, which supplementary might be further used to generate improved summary.

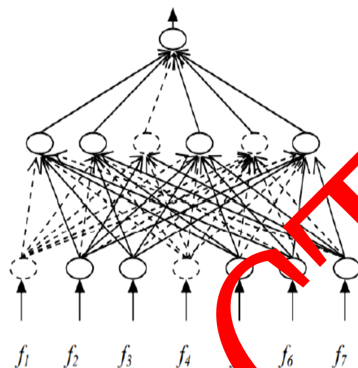

Figure 6: Neural network after pruning [26]

2.1.6 Fuzzy Logic Based Text Summarization

This approach used Fuzzy Logic instruction and fuzzy logic set. It is to identify the vital sentences constructed from on their characteristics. Fuzzy logic methods deliver expert systems and decision-support with strong intellectual abilities. Fuzzy logic theory projected by Zadeh [28] it is a scientific instrument which is efficiently deals with ambiguity, inaccuracy and uncertainty. A limited research was complete in field of text summarization using fuzzy. Witte et al. [29] projected a fuzzy-theory method based on coreference proposal and its submission to text summarization. Automatic Fixation of coreference among noun idioms is replete with ambiguity. Patil et al. [30] also worked on Fuzzy Logic to grade a sentence after collection of features and pre-processing stage. They usage 8 characteristics for text summarization and these are: length of sentence, sentence to sentence similarity, title word, sentence position, thematic words, numerical data, Proper Nouns and term weight. The system comprises in steps describe below:

• Examined the basis collection into the scheme,

• In preprocessing stage, the scheme picks the specific sentences from the unique text. Afterward, distinct the input text into distinct words. Following stop words removal. Word stemming is performed at the last in preprocessing step.

• Every characteristic is linked with vector of 8 characteristics, declared above, whose significance are gained from the contented of the sentence;

• An element with uppermost value sentences is picked as text of summary depends on the compression frequency.

Fuzzy Logic Scheme has mainly four elements: Inference Engine, Fuzzifier, Fuzzy Knowledge Base and Defuzzifier. In the fuzzifier, hard ideas are interpreted into verbal significance using a relationship function. Afterward fuzzification, the implication engine mentions to the instruction dependent holding fuzzy IF-THEN instructions to discover the verbal data.

The final stage, the output verbal variables from the interpretation are transformed to the concluding important values by the defuzzifier using association function for representative the concluding sentence score [31]. In order to implement fuzzy logic-based text summarization, each individual sentence is connected with 8 feature paths. By considering all these eight feature values, the value for individual sentence is calculated via fuzzy logic scheme. The fuzzy rules and triangular association function used by fuzzy logic technique. The triangular association function fuzzifies individual value is between one of the three values that are HIGH, MEDIUM & LOW. The fuzzy rules are described in the form of IF-THEN. Then after enforced fuzzy rules we know whether the sentence is insignificant, regular or significant. This is also called defuzzification. For example, if (X1 is H) and (X2 is H) and (X3 is M) and (X4 is M) and (X5 is H) and (X6 is M) and (X7 is M) and (X8 is H) THEN (sentence is vital). Based on their score all the sentence are graded in a descending order in sentence selection segment. The most n sentences based on score are selected as text summary depends on looseness rate. Lastly, the sentences in instant are organized in the direction in which they arise in the provided document.

3 Summarization Methods Observations

The produced works claims the following foremost opinions.

- The primary job before finding extractive summarization is to discover significant evidence which will be included in the summary.
- Sometime summary holds useless information because of the selected sentences are extensive then regular sentence and contains unnecessary information.
- Important evidence is kept by autonomous segments of the text; sometimes extractive summary may not able to discover altogether useful material across the text.
- Duplicate information may also present in the summary.
- Extraction dependent synonyms are sometime unpleasant to speak.
- Sometime flow of information may be absent in summary as because selected sentence picked from various data fragments divert the topic unexpectedly.
- Creation of summary through abstractive methods is a foremost problem.
- Semantic connection among key term of the text may lose sometimes in abstractive summary.
- For constructing comprehensive summary NLG (Natural Language Generation) instructions are extremely desired.
- No coherent relation found many times in abstractive summaries.
- For good abstractive summary the semantic understanding of text is required.
- Abstractive summaries quality is depending on the deep verbal knowledge.

4 Proposed Approach for Abstractive Summarization

In the previous works we study the techniques that has been used for both extractive and abstractive summarization. In that all machine learning methods were used to create extractive summary whereas neural network and fuzzy logic methods were used to generate abstractive summary. They all had many drawbacks like NN, and Fuzzy approaches faced the problem of long sentence, they forget the sentence if it is long therefore could not generate precise results as anticipated. Therefore, summarization of text with maximum precision can be accomplished through LSTM model where we used LSTM cells, instead of basic RNN cell, along with the attention mechanism to generate precise guess of the summary. Instead of Word2Vec for word embeddings we are going to use the GloVe [32]. which is same as Word2Vec but reasonably better than that. For classification we will be using 1D convolution layer followed by max pooling layer, LSTM layer and then finally used fully connected layer.

This model could produce more accurate and identical summary which is closer to human generated.

An Encoder-Decoder LSTM Architecture [33]

This style for summarization is a technique of forming recurrent neural networks for classification forecast difficulties, which have several numbers of inputs, outputs, or even both. This model has two major components which are: an encoder and a decoder. Encoder essentially accepts the complete input arrangement and encrypts it into an inner symbol, sometimes in a fixed-length vector, known as context vector. The decoder accepts the encrypted sequence provided by the encoder and produces the output sequence. Both the sub models are trained in parallel simultaneously.

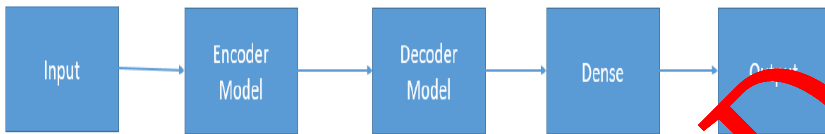

Figure 7: Encoder-decoder LSTM model architecture

B Text Summarization Encoders [33,34]

The encoder is known brain of the system where all extreme complexity and calculation reside. Primary function performed by an encoder in which encoder accepts the original text as input and produce intermediate solution as the output. This output will become input of decoder and then decoder produced the final result. There are many types of the encoders that are used in text summarization application; some are very common like the one which is very popular is LSTM. In training stage each time step 't', we send sentence to encoder through words one by one. For example, if we have a sentence "Geeta is a decent girl", then the word *Geeta* is given to encoder at time step t = 1, the word '*is*' is passed at time step t = 2, and so on.

If there is a sequence *'s'* containing the words w1, w2, w3, w4 then the training stage of encoder looks similar to below (Fig. 8).

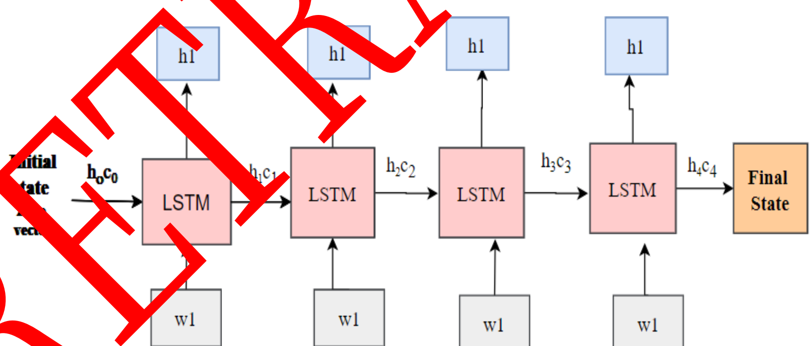

Figure 8: Encoder Architecture training phase

B Text Summarization Decoders

Creation of summary words in output sequence is the responsibility of decoder. The references of decoder are:

a) Context Vector: Is a result produced by encoder and vector representation of source document.

b) Generated Sequence: Is a sequence of the word already a summary word.

Final state of encoder is treated as initial state of decoder. Means decoder is already trained enough to generate output as order depends on the data provided (encoded) by the encoder.

Decoder used <start> and <end> two distinct symbols are appended to the goal sequence before sending it to decoder. Because the target sentence is not knowing when to decode the test sequence, we start forecasting the goal sequence by passing the foremost word to the decoder which is always be a <Start> sign and in conclusion the <End> sign indicates the end of sequence.

LSTM unit of decoder provide the output in the form of s^1, s^2, s^3,.., s^k (Fig. 9), where

s^1 = output generated at time t1

s^2 = output generated at time t2 and so on

K = length of output sentence

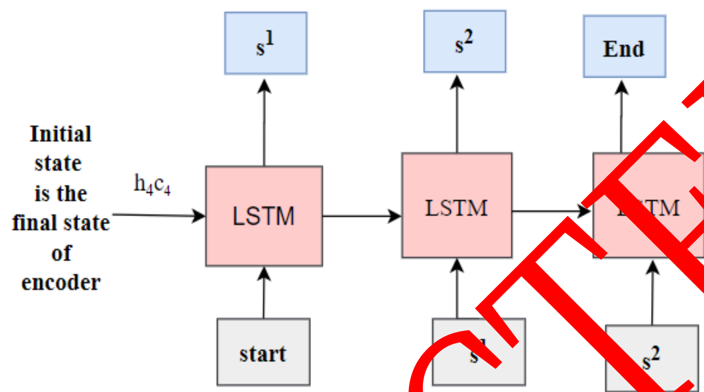

Figure 9: Decoder architecture

But in the testing stage we have a problem that how we decode the test order. We follow the steps mention below:

1. Encode the complete input order and pass to the decoder with inner situations of the encoder;

2. Permit <**start**> symbol as an input to the decoder;

3. With innermost positions run the decoder for one time step;

4. We find the probability for the subsequent word as output. The word with the extreme likelihood will be picked;

5. Permit the experimented word at the next timestep as an input to the decoder and update the inner positions with the existing time step;

6. Until we do not produce <**end**> symbol or not process extreme length of the goal order we repeat Steps 3–5.

4.1 System Architecture

The workflow of our projected model is explained in this section.

Before generating the output, the input text undergoes through different stages. Initially the text will go for text preprocessing step where the text will be produced which is free from noise and undesirable data. This ensures that the data is clean and prepared for the next step. In preprocessing we perform various tasks like 1) Noise Removal, 2) Tokenization, and 3) Normalization. ***Noise Removal*** is the initial step in the preprocessing process in which it involves the elimination of markup data like XML, file headers, HTML, etc., and even mining the vital data from the document, and formats like as, CSV, JSON of XLS files. The second step of preprocessing would be ***Tokenization*** where the larger sentences is broken down of into smaller phrases and then into words. Foe that we use word.tokenize() function of NLTK. The next step of preprocessing is ***Normalization***. It involves changing the entire text into lower case or in upper case, punctuation removal, translation of the numerals to their word equals etc. It converts the text into a similar level which makes the processing of words easy. Tokenized words then go to word embedding segment which, after the preprocessing, identifies the word sense and the attributes that individually word can convey. The text which we get after word embedding to the tokenized words, as output, is passed to our module in where the actual summarization takes place.

As discussed, prior the processed text which we get as output after preprocessing, passed to the word embedding phase which then recognizes the word meaning and the attributes that each word can carry.

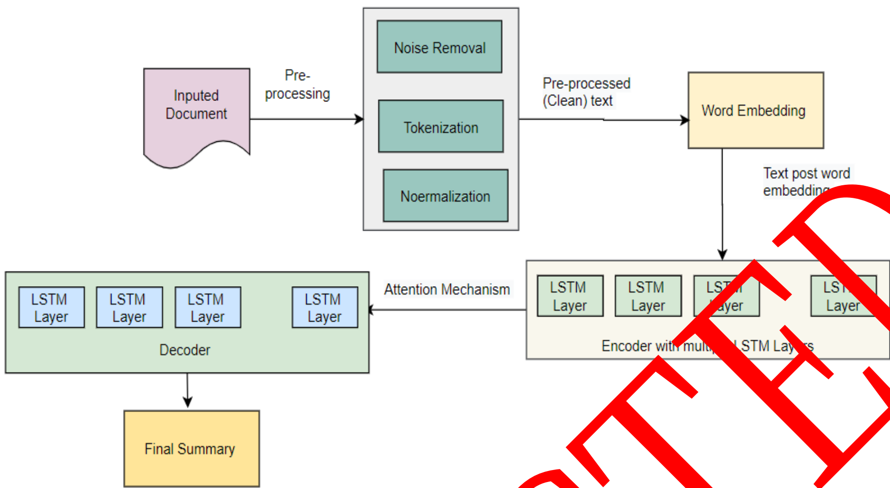

Figure 10: Proposed system architecture

The text is passed through the Encoder which has Multiple LSTM layers. Each LSTM layer has use functions like tanh and sigmoid activation functions which will convert the values of each words in range of {–1 to 1} and {0 to 1} respectively. We have used gates like forget gate, reset gate, output gate in LSTM. These gates taken care the task which information passed to the next LSTM layer and which data is discard (forgotten). Apart from that the gates also recall the relationship among words as we are passing the prior time step output (i.e., previous word) to next LSTM cell and so on. In this method the association will be extracted. The encoder will generate a fixed length representation of the data passed as input. These representations (words) are then feed to the Attention Mechanism. It is actually a tough task to summarize a huge data into some typical words which should depict the similar meaning as that of the entire text. This might result in the loss of some vital information. This is addressed by viewing if any specific words have different sense in the local and global situation. These words need special attention for that specific situation, and that's why ***attention mechanism*** is needed. This helps in choosing the particular words with extra accuracy. The attention mechanism assigns distinct value to those words which play the major role of the summary. Suppose we want to predict summary for Chicken Burger. Then the words like spicy, good, delicious and tasty should have a higher priority on the words like 'the, is, it'. This is taken care by attention mechanism layer. The output generated by attention mechanism layer is passed to the decoder module which again contains various LSTM layers whose job is to forecast the words which are significant. This decoder also consists the LSTM layers which taken care the representation of an embedding of the word that is last produced by encoder and then uses these as inputs for generating the final words in the summary of the text. The LSTM cells have the same functionality as that of in the encoder but as the input changes in the decoder module the output will be a summary of the text without losing the meaning of the text.

5 Conclusion

Because information overload is a common problem nowadays due to rapid evolution of technology and www (world wide web), the problem can be resolved if we have any exist robust text summarizers which is capable of providing a good summary of document to user. Automatic Text Summarization is a common study domain from last decades which get consideration from many science disciplines. This paper discussed various types of summarization techniques which could be applied in a technique to generate summary. We studied various Extractive summarization methods based on machine learning approach like Decision tree, SVM, Bayesian Classifier, Hidden Markov Model etc. We also studied a range of novel models using neural networks for abstractive summarization which needs substantial technology from natural language processing, includes superior knowledge of language rules (Grammar) and dictionaries for analyzing and creation of summary. Exclusively, this paper focus on extractive summarization approaches and proposed a novel approach using Neural network model uses LSTM with encoder and decoder.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. Hovy and C. Y. Lin, "Automated text summarization and the summarist system," in *Proc. of the TIPSTER Text Program*, pp. 197–214, 1998.

[2] I. Mani and M. T. Maybury, *Advances In automatic Text Summarization*. United States: MIT Press, vol. 293, 1999.

[3] H. P. Luhn, "The automatic creation of literature abstracts", *IBM Journal of Research and Development*, vol. 2, pp. 159–165, 1958.

[4] C. Zhang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[5] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," *ACM SIGIR Forum.* [Online]. Available: https://doi.org/10.1145/3130348.3130369, 2017.

[6] R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[7] R. Quinlan, *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, Publishers, Inc., 1993.

[8] C-Y. Lin and E. H. Hovy. "Identifying topics by position," in *Proc. of the Applied Natural Language Proc. Conf.*, pp. 283–290, 1997.

[9] H. P. Edmundson, "New methods in automatic extraction," *Journal of the ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[10] C. D. Paice, "Constructing literature abstracts by computer: Techniques and prospects," *Information Processing and Management*, vol. 26, no. 1, pp. 171–186, 1990.

[11] C. Buckley, C. Cardie, "SMART summarization system," in *Proceedings of the TIPSTER Text Phase III Workshop*, Washington, USA: ACM, pp. 21–28, 1997.

[12] D. Marcu, "The rhetorical parsing, summarization, and generation of natural language texts," Ph.D. dissertation, University of Toronto, Canada, 1997.

[13] C. Y. Lin, "Topic identification by concept generalization," in *Proc. of the Thirty-third Conf. of the Association of Computational Linguistics (ACL-95)*, pp. 308–310. 1995.

[14] C. Y. Lin, "Robust automated topic identification," Ph.D. dissertation, University of Southern California, Canada, 1997.

[15] K. R. McKeown and D. R. Radev. "Generating summaries of multiple news articles," in *Proc. of the 18th Annual Int. ACM Conf. on Research*, pp. 74–82, 1995.

[16] E. H. Hovy and L. Wanner, "Managing sentence planning requirements," in *Proc. of the Workshop on Gaps and Bridges in NL Planning and Generation, ECAI Conf.*, pp. 53–58, 1996.

[17] I. Langkilde and K. Knight, "Generation that exploits corpus knowledge," in *Proc. of the COLING/ACL Conference*, pp. 704–709, 1998.

[18] T. Hirao, Y. Sasaki, H. Isozaki and E. Maeda, "NTT's text summarization system for DUC-2002," in *Proc. of the Document Understanding Conf.*, pp. 104–107, 2002.

[19] S. A. Babar and P. D. Patil, "Improving performance of text summarization," *International Conference on Information and Communication Technologies*, vol. 46, pp. 354–363, 2014.

[20] L. Suanmali, N. Salim and M. S. Binwahlan, "Fuzzy logic based method for improving text summarization", *International Journal of Computer Science and Information Security*, vol. 2, no. 1, 2009.

[21] G. PadmaPriya and K Duraiswamy, "An approach for text summarization using deep learning algorithm", *Journal of Computer Science*, vol. 10, no. 1, pp. 1–9, 2014.

[22] F. Kyoomarsi, H. Khosravi, E. Eslami and P. H. Dehkordy, "Optimizing text summarization based on fuzzy logic", in *Proc. of Seventh IEEE/ACIS Int. Conf. on Computer and Information Science*, pp. 347–352, 2008.

[23] J. Kupiec, J. Pedersen and F. Chen, "A trainable document summarizer," in *Proc. of the 18th ACMSIGIR Conf.*, pp. 68–73, 1995.

[24] J. Conroy and D. P. O'leary, "Text summarization via hidden markov models and pivoted qr matrix decomposition", in *Proc. of the 24th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2001.

[25] K. Kaikhah, "Automatic text summarization with neural networks," in *Proc. of Second Int. Conf. on intelligent systems*, vol. 1, pp. 40–44, 2004.

[26] K. Kaikhah, "Text summarization using neural networks," *Department of Faculty Publications Computer Science*, 2004.

[27] A. T. Sarda and A. R. Kulkarni, "Text summarization using neural networks and rhetorical structure theory," *International Journal of Advanced Research in Computer and Communication Engineering*, 2015.

[28] L. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.

[29] R. Witte and S. Bergler, "Fuzzy coreference resolution for summarization," in *Proc. of 2003 Int. Sym. on Reference Resolution and Its Applications to Question Answering and Summarization*, 2003.

[30] P. D. Patil and N. J. Kalkurni, "Text summarization using fuzzy logic," *International Journal of Innovative Research in Advanced Engineering*, vol. 1, no. 3, 2014.

[31] L. Suanmali, N. Salim and M. S. Binwahlan, "Fuzzy logic based methodfor improving text summarization," *International Journal of Computer Science and Information Security*, vol. 2, no. 1, 2009.

[32] Nlp.stanford.edu, 2019. [Online]. Available: https://nlp.stanford.edu/pubs/emnlp15_attn.pdf.

[33] A. M. Rush, S. Chora and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," 2015. [Online]. Available: https://arxiv.org/abs/1509.00685. 2015.

[34] K. Lopyrev, "Generating news headlines with recurrent neural networks," [Online]. Available: https://arxiv.org/abs/1512.01712, 2015.