Medical Abridgement for Enhancing Physicians' Accuracy

Insiya Abdulsalam[1], Anju K.S[2], Surekha Mariam Varghese[3]

[1]Computer Science and Engineering Department, Mar Athanasius College of Engineering, Kerala, India, inusaam7@gmail.com.
[2]Computer Science and Engineering Department, Mar Athanasius College of Engineering, Kerala, India, anjusanthosh999@gmail.com
[3]Computer Science and Engineering Department, Mar Athanasius College of Engineering, Kerala, India, surekha@mace.ac.in

---

ABSTRACT: Managing huge amount of medical documents and obtaining relevant information from them within a reasonable time is a challenging task. Abstractive summarization using sequence to sequence encoder-decoder for single document summarization is proposed here for solving these issues. It is implemented by using LSTM recurrent neural network. The LSTM cell remembers values over arbitrary time intervals and the gates manage the flow of information into and out of the cell. An encoder LSTM network receives document in vector form and encodes it into a single vector. This encoded vector is fed to the decoder LSTM network. In each time step, the output of the decoder is fed to the Softmax layer. Softmax layer predicts the probability of each word to be generated at the current time step. The word with the highest probability will get generated. These generated words together form the summary of the document. The proposed abstractive summarization system using deep learning attained a ROUGE score of 0.7 and improved the legibility of the generated summary.

Key words: Abstractive summarization · LSTM · RNN · Softmax layer

1. INTRODUCTION

Summarization techniques have become increasingly important in biomedical research, over the last few years due to information overload. Medical Literature such as medical news, research articles, and clinical trial reports on the web serves as an important source of information which help clinicians in patient treatment. Initially, clinicians go through author-written abstracts or summaries available with the medical articles to decide whether articles are relevant to them for in-depth study. Since all types of medical articles do not come with author written abstracts or summaries, automatic summarization of medical articles will help clinicians or medical students to find the relevant information on the web rapidly. Moreover, to keep track of infectious disease outbreaks or other biological threats calls for rapid information gatherings and summarization. Text summarization plays a pivotal role in alleviating the problem of accessing accurate and up-to-date information relevant to biomedical researchers and physician's needs.

Biomedical literature and medical information such as health records are now accessible from various resources. The clinical researchers, physicians, and information seekers find it exceedingly difficult to gain the relevant and required information for giving a proper patient care and for conducting experiments. To provide most appropriate care for patient, clinicians need to efficiently and effectively retrieve, interpret, and integrate relevant information from multiple sources [1]. Summarization helps to reduce this complexity.

Extractive summarization techniques extracts the important sentences from the document based on various criteria and forms the summary of the document so that the clinicians can easily understand the major concepts of the document [2]. Extraction may be inappropriate because there is a possibility of anaphoric link to be extracted without previous context and also it may produce summaries which are overly verbose or biased towards some sources [3].

Here, an automated abstractive summarization system for medical reports has been proposed. Advantage of abstractive summarization is that it generates less redundant and grammatically correct sentences and also it gives combined meaning of sentences [4].

Various methods like tree-based, graph-based are available to achieve abstractive summarization. The proposed system introduces a novel abstractive summarization system to effectively and efficiently generate the summary of the document using deep learning concepts. Aim of this work is to implement an automatic abstractive summarization system to generate an efficient and accurate summary of the medical documents.

2. LITERATURE SURVEY

Text summarization reduce the size of a document while preserving the main concepts conveyed by the document. Extractive summarization technique selects the exact sentences from the original document to generate summary [5]. Moen et.al, 2016 presented extractive text summarization methods for summarizing documents. These methods are based on word space models built using distributional semantic modelling [6].

Most of the works in the field of abstractive summarization focus on the components like parsing, Coreference resolution, construction and merging of semantic graphs, natural language generation, lexical chains and distributional semantics for generating the final summary out of the selected sentences [7].

Barzilay et.al, 2005 used text-to-text generation to create informative summaries [8]. Sentences were represented using dependency trees and common information among sentences was determined by processing the trees. They computed the fusion lattice by finding the intersection of sub-trees and then used tree traversals on them to produce the final sentence. One of the limitation of this approach is that it is unable to recognize the relation between the sentences without identifying the intersected phrase between the sentences. Graph based techniques are very frequently used in sentence compression but in most cases they resulted in redundancy [9].

Mozhgan et.al., 2018 proposed a method in which biomedical summary is generated using a graph based method [11]. Major concepts in the document are extracted using UMLS and a semantic graph is built which represent the relation between various concepts in the document. This helps in identifying medically relevant concepts. Anju et.al., 2019 used methods like Locality Sensitive Hashing and Word2Vec to identify the semantic similarity between the chunks [12]. These methods enhanced the performance by preventing duplication in the data.

Deep learning has now emerged as a new technique to model the abstractive summarization problem which can capture both the structural and semantic information of the text [7]. It has been successfully applied to text summarization [13]. Nallapati et.al., 2016 used encoder-decoder Recurrent Neural Network (RNN) along with Gated Recurrent Unit (GRU) for effective summarization [14]. The bidirectional GRU can solve the problem of vanishing gradient. Vanishing gradient problem happens during the training of neural network which prevents the network from further training. Word embedding's are given as input for the neural networks for training purpose and attention mechanism is used for creating the context vector at each time step. Disadvantage with GRU is slow convergence rate and low learning efficiency, which results in long training time [17].

RNNs are very successfully implemented for processing sequential data [15]. RNN models help predict complex relations which simple structured or semantic type models cannot do [16]. By introducing a gate into cell, Long-Short Term Memory cell (LSTM) can handle problem of long term dependencies. Though there are several neural network cell variants like RNN and GRU, LSTM outperforms all others [17]. The advantage of deep learning models is inclusive semantics, because these models learn the collocation between words, and will reproduce a sequence of words based on the collocation between words after training [18].

RNN does not allow memory to persist for a longer time in the cells, effective text summarization can be achieved by using convolutional gated units along with the global encoding at the encoder side and unidirectional LSTM at the decoder side [19]. Sequence-to-sequence models were used by Lin et.al., 2018 along with attention mechanism to solve the problem of repetitions. The major advantage of attention is that it gives the user ability to interpret and visualize what the model is doing [20].

Medical domain has got some additional challenges as compared to other domains. The uniqueness of medical document is due to their heterogeneity, volume, and also due to the fact that they are most rewarding documents to analyze especially those that contain human medical information due to the expected social benefits [21]. Unavailability of medical dataset for summarization due to privacy concerns is also a challenge in carrying out experiments in medical domain [22].

3. PROPOSED METHODOLOGY

Medical summary generation can be modeled as a sequence-to-sequence converter. LSTM improves memory persistence. Persistent memory, helps the neural network to update its state dynamically in accordance with the similarity between the encoded input and input slots, resulting in a stronger capacity in assimilating sequences with multiple patterns [23]. LSTM is a variant of RNN where flow and period of persistence of information can be regulated using several gates. LSTM was specially designed for maintaining long term dependencies.

Since it is the summarization of medical data, more importance has to be given to medical terms. UMLS Metathesaurus is used to identify all the medically relevant terms. A tool called Metamap assigns a semantic label to each word in the document [24]. Words are send to the encoder along with its semantic label, so that through efficient training the system can generate more medical terms in the summary.

A sequence of input words (source document) are fed into the system and it outputs another sequence of words (summary).

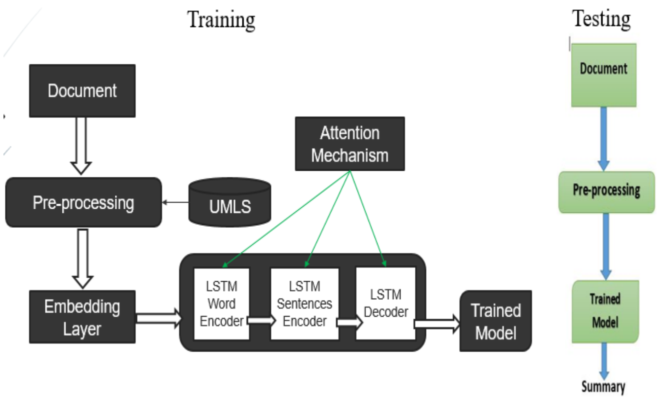

Figure 1: System Architecture

System consists of two main parts, encoder and decoder. Two LSTM networks are used in the system. Bidirectional LSTM network is used to implement encoder and unidirectional LSTM is used for implementing decoder. There are two phases
in the system implementation, training phase and testing phase. During training, the system maximizes the probability of generating the correct summary of the source document.

Major stages of proposed system are vector representation, encoding, single vector representation, decoding and summary generation.

3.1 Vector Representation

Machine learning algorithms cannot work with categorical data directly [25]. The data must be initially converted to numbers. This is required for both input and output that are textual. An integer can be encoded directly, rescaled where needed. For this reason text is converted to vectors. Integer encoding and one hot encoding are the two prominent encoding techniques commonly used. Integer encodings are used when there is a natural ordinal relationship between the categories, otherwise one hot encoding is used. It allows the representation to be more expressive [26]. After tokenization, tokens are converted into vectors by using one hot vector method. Each words in the source document is converted to vector form. For example, in the document X, all the words are converted into sequence of vectors, $X = x_1, x_2, ..., x_n$. A vocabulary is created from the training dataset and maintained. In order to convert words in the documents into vector, numpy function in tensorflow is used.

3.2 Encoding

A Seq2seq model maps two sequences to each other that are not necessarily in the same size, in two steps: Compressing the first sequence, and then inferring the output from it. This architecture has two sides named encoder and decoder that are both LSTM layers. The encoder receives the input data step by step. At each time step the state of the hidden layer is looped back and combined with the input data. At the end of this procedure, the hidden layer of last time step holds a state which has been affected by all the elements in the sequence or has retained the memory of the whole sequence in a single layer. The name of Encoder originates here, because it encodes a long sequence to the state of a hidden layer which is a vector.

Tokens are fed to the encoder network one at a time which is terminated by an <eos> tag. As input is fed to the encoder in vector form the hidden values get continuously updated until the <eos> tag. At the end encoder will generate a single vector which represents the semantic meaning of the entire document. The encoded version of the input sequence is a representation of its information and its principal patterns. It is like a compressed memory of the entire elements of the input. Output of an encoder is calculated as equation 1, where hj is the output of current encoder unit, hj−1 is the output of previous encoder unit and xj is the current input.

$$h_j = \mathrm{LSTM}(h_{j-1}, x_j) \qquad (1)$$

Regardless of how long the input is, hidden node layer at the end of input sequence or embedded vector contain all the information about the input. This means that this embedded vector is overloaded with information and also information at the final layer may get diluted. In fact, some of the input units are directly linked to some output units [27]. Attention mechanism is used to bring out this connection as depicted in figure 2. Here, a weighed combination of all hidden input units are fed to each output unit. Weights are function of current output state which vary by output time. Weights at ith hidden representation is a function of ith hidden representation and hidden output state at time t-1, it is represented in the equation 2 where $w_i(t)$ is the weight associated with the ith hidden representation at time t, hi is the ith hidden representation and s_{t-1} is hidden output state at time t-1. Input to each decoder is give in equation 3.

$$w_i(t) = a(h_i\, , s_{t-1}) \qquad (2)$$

$$\text{Input to each decoder unit} = \Sigma_i w_i(t) h_i \qquad (3)$$

3.3 Decoding

To make the decoder guess which summary word matches with source sentence, put the hidden state from encoder to the first time step of the decoder. This way decoder can be trained with the presence of the encoder's content. Internal state of decoder is calculated using equation 4, where di is the internal state of current decoder unit, d_{i-1} is the internal state of the previous decoder unit and y_i is the output generated by the previous decoder unit.

$$d_i = \mathrm{LSTM}(d_{i-1}, y_i) \qquad (4)$$

Finally the decoder will acquire a set of weight that will generate a correct summary with the presence of hidden state of

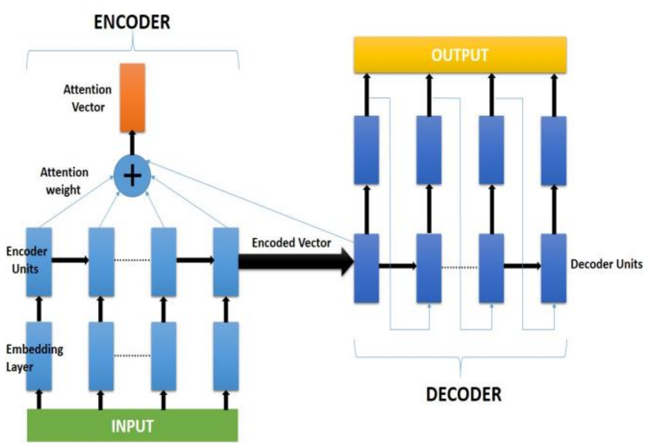

Figure 2: Sequence-to-sequence model with attention

the encoder (the memory of source sentence) in its layers. The decoder receives the encoded vector which represents the semantic meaning of the document and decodes it. The output is received from the output layer of the network and give it to the softmax layer. This layer assigns a probability of occurrence for each word generated and the word with highest probability is drawn as the output. The attention distribution tells the network which of the encoder unit contribute more for the prediction of a particular output word.

3.4 Summary Generation

Softmax layer assigns probability to every words in the dictionary to get selected next or to be generated next as a part of the summary. Softmax function is a type of squashing function. Squashing functions makes the output of the function converge to the range 0 to 1. This allows the output to be interpreted directly as a probability. The word with highest probability among others will get generated. This supplementary restriction helps training converge more quickly than it otherwise would.

4. RESULT AND PERFORMANCE ANALYSIS

Medical dataset was prepared by collecting articles from New England Journal of Medicine. During training, article was given as the source document and their abstract was fed as summary.

Result of the model is evaluated with the standard ROUGE Score. It works by comparing an automatically produced summary or translation against the human generated summary. Some of the major evaluation techniques available are cosine similarity, unit overlap, LSA-based measures, Longest Common Subsequence etc. If abstracts are generated ROUGE1 score is the best option [32], so ROUGE1 is considered here.

Figure 3 shows the medical article fed to the summarizer and figure 4 shows the abstractive summary generated by the system.

Out of hospital cardiac arrest is a leading cause of death in europe and the united states. despite advances in the field of resuscitation and intensive care management, the outcome in patients after cardiac arrest remains poor. a recent study reported mortality of approximately 40% among patients who had been successfully resuscitated after out-of-hospital cardiac arrest associated with ventricular fibrillation or pulseless ventricular tachycardia. 1 recommended postresuscitation care includes targeted temperature management, vital-organ support, and treatment of the underlying cause of the arrest. however, the cause of arrest is often unclear immediately after the event, and the lack of a definitive diagnosis can lead to uncertainty regarding the appropriate treatment. the most frequent cause of cardiac arrest is ischemic heart disease, and coronary artery disease has been reported in up to 70% of patients who have been resuscitated and are referred for immediate coronary angiography. 2 if myocardial infarction is the cause of the arrest, immediate percutaneous coronary intervention might salvage myocardium, improve circulatory function, and prevent the recurrence of life-threatening arrhythmias. current european and american guidelines recommend immediate coronary angiography with pci in patients who present with st-segment elevation myocardial infarction and cardiac arrest. in patients with cardiac arrest who do not have st-segment elevation on electrocardiography, the role of immediate coronary angiography is still a matter of debate. data from randomized trials are lacking, and observational studies have shown conflicting results regarding the effect of immediate coronary angiography and pci on outcomes in this patient group. at present, international guidelines on cardiopulmonary resuscitation recommend emergency coronary angiography in selected patients after out-of-hospital cardiac arrest, even in the absence of st-segment elevation. it has been advocated, however,

Figure 3: Medical article given to the summarizer

despite advances in the field of resuscitation and intensive care management the outcome in patients after cardiac arrest remains poor. if myocardial infarction is the cause of the arrest immediate percutaneous coronary intervention might salvage myocardium, improve circulatory function, and prevent the recurrence of life-threatening arrhythmias. current european and american guidelines recommend immediate coronary angiography with pci in patients who present with st-segment elevation myocardial infarction and cardiac arrest. in patients with cardiac arrest who do not have st-segment elevation on electrocardiography the role of immediate coronary angiography is still a matter of debate. data from randomized trials are lacking, and observational studies have shown conflicting results regarding the effect of immediate coronary angiography and pci on outcomes in this patient group.

Figure 4: Generated summary

4.1 Precision and Recall in the Context of ROUGE

Since ROUGE score is used, overlapping words are counted. Overlapping words are those words which occur both in generated summary and reference summary [28].

Recall is measured by counting the number of common words in the system generated summary and reference summary and divide it by the total number of words in the reference summary. Equation 5 is used for the calculation of recall.

Precision is measured by counting the number of common words in the system generated summary and reference summary and divide it by the total number of words in the system summary. Equation 6 shows the equation of the precision.

$$ROUGE1_{Recall} = \frac{\text{Number of overlapping words}}{\text{Total words in reference summary}} \quad (5)$$

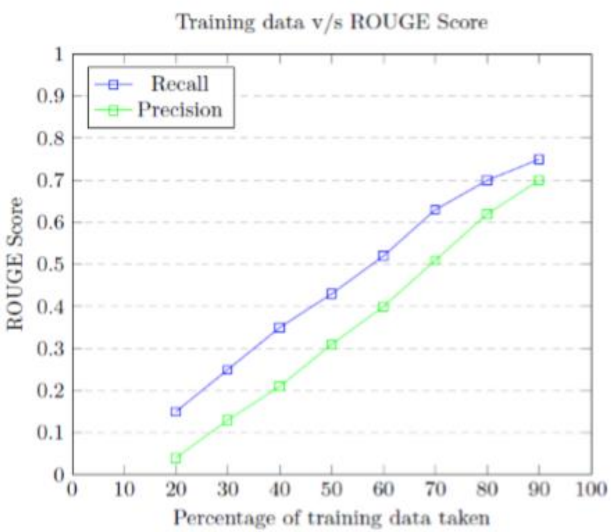

Figure 5: Precision and Recall When the System is Implemented on Medical Dataset

$$ROUGE1_{Precision} = \frac{\text{Number of overlapping words}}{\text{Total words in system summary}} \quad (6)$$

Higher the ROUGE Score, higher will be the accuracy of the generated summary. Testing shows that the ROUGE Score (Precision and Recall) directly depends on the amount of training data. The experiments have been conducted with different random samples of data and Precision and Recall of the system when it is implemented on medical dataset has been calculated.

Figure 5 shows the performance of the system. $ROUGE_{Recall}$ score is 0.75 when the system is trained with 90 percent of the data and the system attained $ROUGE_{Recall}$ score of 0.7 when it is trained with 80 percent of the data. $ROUGE_{Precision}$ score is 0.7 when the system is trained with 90 percent of the data.

It is found that the accuracy of the system increases with the amount of training data. Larger the training dataset, higher will be the accuracy (ROUGE Score). A good summarizer has ROUGE score 0.7 and above. The proposed system attained this ROUGE score when it was trained with 70 percent of the news article dataset and 80 percent of the medical dataset. The

5. CONCLUSION

The system was implemented on medical dataset with limited number of article-summary pair. In this work, the proposed system architecture obtained ROUGE1 scores of 0.85 and 0.75 on the news article dataset and medical dataset respectively. Based on the results obtained using the proposed model, it can be concluded that the accuracy of the generated summary increases with percentage of training data used. A good summarizer has ROUGE score above 0.7. The proposed system attained ROUGE score of 0.7 at 80 percentage of training data. Performance can be improved by using a larger dataset.

REFERENCES

1. R. Mishra, J. Bian, M. Fiszman, C.R. Weir, S. Jonnalagadda, J. Mostafa, et al. **Text summarization in the biomedical domain: A systematic review of recent research,** J. Biomed.Inform. 52 (2014) 457–467.
2. Afantenos, V. Karkaletsis, P. Stamatopoulos **Summarization from medical documents: a survey,** Artif. Intell. Med. 33 (2) (2005) 157–177.
3. Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. **Information fusion in the context of multi-document summarization,** In Proc. 37th ACL, 550–557.
4. N. R. Kasture, Neha Yargal, Neha Nityanand Singh, Neha Kulkarni, Vijay Mathur, **A Survey on Methods of Abstractive Text Summarization,** International Journal for Research in Emerging Science and Technology, Volume-1, Issue-6, November-2014.
5. Y. Ko, J. Park, J. Seo **Automatic text categorization using the importance of sentences**, Proceedings of the 19th international conference on Computational linguistics – Volume1, Association for Computational Linguistics, 2002, pp. 1–7.
6. Hans Moena, Laura-Maria Peltonen, Juho Heimonen, Antti Airola, Tapio Pahikkala, Tapio Salakoski, Sanna Salanter¨a, **Comparison of automatic summarisation methods for clinical free text notes,** Artificial Intelligence in Medicine 67 (2016)
7. Som Gupta, S.K Gupta, **Abstractive Summarization: An Overview of the State of the Art,** Expert Systems With Applications, (2018), doi: https://doi.org/10.1016/j.eswa.2018.12.011
8. Regina Barzilay, Kathleen R. McKeown, **Sentence Fusion for Multidocument News Summarization,**

2005 Association for Computational Linguistics-Volume 31, Number 3.
9. Katja,C. **Multi-sentence compression: finding shortest paths in word graphs,**COLING'10 Proceedings of the 23rd International Conference on Computational Linguistics(pp.322–330).
10. Khushboo S. Thakkar, Dr. R. V. Dharaskar, M. B. Chandak, **Graph-Based Algorithms for Text Summarization,** Third International Conference on Emerging Trends in Engineering and Technology.
11. Mozhgan Nasr Azadani, Nasser Ghadiri **Graph-Based Biomedical Text Summarization: An Itemset Mining And Sentence Clustering Approach,** Journal of Biomedical Informatics 84 (2018) 42–58
12. Anju K S, Sadhik M S, Surekha Mariam Varghese. **Semantic Deduplication in Databases,** International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6S, April 2019
13. JiweiLi, Minh-ThangLuong, DanJurafsky **A Hierarchical Neural Auto encoder for Paragraphs and Documents,** arXiv:1506.01057v2 [cs.CL] 6 Jun 2015
14. Nallapati, R., Zhou, B., dos Santos, C., Gulehre, C., Lapata, M. (2016). **Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond**, In The SIGNLL Conference on Computational Natural Language Learning
15. Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang, **A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,** Neural Computation 31, 1235–1270 (2019)
16. Jobson, E., Gutirrez, A. **Abstractive text summarization using attentive sequenceto-sequence rnns,** Stanford Reports.
17. Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang, **A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures**, Neural Computation 31, 1235–1270 (2019) c 2019 Massachusetts Institute of Technology doi:10.1162/necoa01199
18. Shengli Song, Haitao Huang Tongxiao Ruan, **Abstractive text summarization using LSTM-CNN based deep learning,** Springer Science+Business Media, LLC, part of Springer Nature 2018
19. Lin,J.,Sun,X.,Ma,S.,Su,Q.(2018). **Global encoding for abstractive summarization,** In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (pp.163–169).
20. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jone, Aidan N.Gomez, Lukasz Kaiser, **Attention Is All You Need**, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
21. Stergos Afantenosa, Vangelis Karkaletsis, Panagiotis Stamatopoulos, **Summarization from medical documents: a survey**, Artificial Intelligence in Medicine (2005) 33, 157—177.
22. Krzysztof J. Cios, G. William Moore, **Uniqueness of medical data mining**, Artificial Intelligence in Medicine 26 (2002) 1–24.
23. Kui Zhao, Yuechuan Li, Chi Zhang, Cheng Yang, Shenghuo Zhu**, PRNN: Recurrent Neural Network with Persistent Memory,** arXiv:1801.08094v1 [cs.LG] 24 Jan 2018
24. Alan R Aronson, Fran¸cois-Michel Lang**, An overview of MetaMap: historical perspective and recent advances**, J Am Med Inform Assoc 2010;17:229e236. doi:10.1136/jamia.2009.002733
25. Kyunghyun Cho, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio**, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine,** Translation, arXiv:1406.1078v3 [cs.CL] 3 Sep 2014.
26. Kedar Potdar, Taher S. Pardawala , Chinmay D. Pai, **A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,** International Journal of Computer Applications (0975 – 8887) Volume 175 – No.4, October 2017
27. SumitChopra, Michael Auli, Alexander M. Rush, **Abstractive Sentence Summarization with Attentive Recurrent Neural Networks,** Proceedings of NAACL-HLT 2016, pages 93–98, San Diego, California, June 12-17, 2016. c 2016 Association for Computational Linguistics
28. Lin, Ch.—Hovy, E.: **Automatic Evaluation of Summaries Using n-Gram CoOccurrence Statistics**, In Proceedings of HLT-NAACL, Edmonton, Canada, 2003.
29. J. Han, J. Pei, Y. Yin **Mining frequent patterns without candidate generation,** SIGMOD Rec. 29 (2) (2000) 1–12.
30. L.H. Reeve, H. Han, A.D. Brooks **The use of domain-specific concepts in biomedical text Summarization,** Inf. Process. Manage. 43 (6) (2007) 1765–1776.
31. R. Agrawal, T. Imielinski **Mining association rules between sets of items in large databases,** SIGMOD Rec. 22 (2) (1993) 207–216.
32. Josef Steinberger, Karel Jezek, **Evaluation Measures for Text Summarization, Computing and Informatics,** Vol. 28, 2009, 1001–1026, V 2009-Mar-2