

An Abstractive Text Summarization using Recurrent Neural Network

Dipanwita Debnath¹, Partha Pakray¹ and Alexander Gelbukh²

Department of Computer Science & Engineering
National Institute of Technology Mizoram¹
Aizawl, India

{parthapakray, ddebnath.nita}@gmail.com
Center for Computing Research (CIC),
Instituto Politecnico Nacional (IPN), Mexico City,²
Mexico
{gelbukh@cic.ipn.mx}

Abstract. With the accelerated advance of technology and massive content surging over Internet, it become an arduous task to abstract the efficient information. Text Summarization provides an acceptable means for fast procurement of information in the form of summary through compression and refinement. Abstractive Text Summarization, in particular, builds an internal semantic representation of the text and uses natural language generation techniques to create summary closer to human generated summary. LSTM based Recurrent Neural Network generates comprehensive summaries through exhaustive training using a parallel corpus having significant number of instances in the form of parallel running article and summary sentences. We pre-processed the dataset to eliminated noise and other irrelevant data and to produce structured representation of the text suitable for training and testing with our system. We have trained and tested our system using LSTM based Recurrent Neural Network on various news corpus namely DUC 2003, DUC 2004 and Gigaword corpus. Experiments and analysis of this work is performed on a subset of these whole corpus. Each and every system generated summaries in comparison with the respective target summary have been evaluated using ROUGE evaluation. Experimental result verifies the accuracy and validity of the proposed system.

Keywords: Abstractive Text Summarization, OpenNMT, ROUGE, Long Short Term Memory, Recurrent Neural Network.

1 Introduction

Text Summarization is the process of shortening a text article in order to create a comprehensive summary which captures the key idea of the article. Text Summarization is highly challenging job despite of the long history of research. The main idea of summarization is to find a subset of text which depicts the entire set of text. It gained widespread interest due to overwhelming amount of textual

information availability over the Internet and data analyst needs to go through these huge number of documents every day to abstract important information from them, and a large portion of their time is spent in figuring out the relevant one. By extracting key idea and creating comprehensive summaries, with the help of software, it is possible to quickly assess whether or not a document is worth reading.

Text summarization technique can be broadly grouped into Abstractive Text Summarization (ATS) and Extractive Text Summarization (ETS) [17]. Extractive methods work by selecting most silent featured set of text to generate summary from the source article. In contrast, Abstractive methods build an internal semantic representation and then use natural language generation techniques to create summary closer to what a human might express. It is called Abstractive because it understands the semantics and generates a summary or abstract of the whole article which is not merely a text from the source article but an abstract of the main article which depicts the key concept of main article. For an example, we read a story, understand it and wrote its summary in our own language. These Abstractive summary may or may not contain the original words or sentence from the main article. As a consequence, they require rephrasing and reconstruction of sentences after analyzing the source article. In Abstractive Text Summarization hence, summary may contain new phrases that are not available in the source text.

ATS are further classified into two types: Structure based and Semantic based approach. Structure based approach translates most important information from the document through cognitive schemes such as tree, ontology, lead and body phrase structure. In Semantic based method, semantic based methods uses natural language generation system to generate summaries. Multi modal semantic model, information item based method, semantic graph based method are some of the methods of Semantic Based Approach[1]. Although conventional approaches of summarization have served the purpose for years, their underlying demerits and increased aspiration for perfection in summarization, enforce exploration of competent emerging techniques. Hence, we used Open Neural Machine Translation (OpenNMT)¹, a most successful and proficient approach to neural machine based summarization, incorporates use of exhaustively trained large neural networks in summarization process [15][14].

OpenNMT based text summarization model maps the input sequence of the source article to the output sequence of the target summary to generate model which can further generate similar target summaries from the given test articles. It calls sequence to sequence Recurrent Neural Network because each time it takes a sequence of input and maps the corresponding target sequence to train the system. OpenNMT are successful to a large extent such as in the field of machine translation, summarization and speech recognition etc. because its is a complete platform having pre-processing, training, testing and evaluation section.

¹ <http://opennmt.net>

In neural network based summarization, encoder encodes the source article, one token at a time, uses the LSTM and stores the entire encoding data in its last hidden state[15]. Encoded representation is fed to decoder for summary prediction. Comprehensibility, adequacy and fluency of system generated summary are largely determined by the implementation approach used for decoder. A radical improvement in summarization quality has been observed by use of LSTM based Recurrent Neural Networks, in place of convolution neural networks[12][14] furthers the effectiveness of summarization by allowing decoder to access entire pool of encoder states for summary prediction.

We used ROUGE for automatic evaluation of summaries. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation [8]. It includes measures to automatically determine the quality of system generated summaries by comparing it to other target summaries created by human of the same source article. The measure counts the number of overlapping units such as n-gram, word sequences, and word pairs between the two parallel summaries. We have collected a large amount of data having parallel running source and summary pair and prepared them for training, validation and testing purpose. On these corpus, experiments are carried out with a comparative manner. We comprehensively analyzed performance change of Abstractive Text Summarization using OpenNMT system by varying training data, epochs and other Neural Machine parameters.

The rest of the paper is organized as follows: Overview of text summarization and related work is presented in section 2. In Section 3 detailed system architecture is described. In section 4. We discussed the corpora and experimental setups used. Section 5 describes system results and their comprehensive analysis. Finally section 6 concludes the paper with some insights to future work.

2 Background and Related Work

In the field of text summarization research work started in the year 1950 though a wide range of past work in summarization are of extractive, which consists of identifying key sentences from the article and generating the summary[5] . In the beginning of 1958 frequency based summarization was introduced where frequency is assigned to each words and top scored sentences formed the summary [10]. On the intuition that important sentences are located at certain position in article or in paragraph, such as beginning and end of a paragraph, preference to sentence location along with the frequency based sentence scoring criteria is added by [1]. Headline words similarity and clue word feature is added to this high frequency content words and sentence location feature [4] in the year 1969.

The essence of the extractive text summarization is a selection problem i.e. selecting the best featured text that can represent the main article[13]. The words in the summary generated by extractive summarization is a subset of words of the main article. while abstractive summarization is a novel text generation process which requires a deep semantic and discourse understanding of the text. ATS understands the article and produces the summary using its own vocabulary. Hence the summary words are not a subset of article words. Most

of the conventional summarization systems use extractive approaches based on human-engineered features. The abstraction-based models mostly provide the summary by sentence compression and reformulation which requires complex linguistic processing [3][7][16] and requires more efforts as compared to extractive summary but provides a better summary.

Abstractive Text summarization using neural network gained a widespread interest because of its noticeable performance. Though neural machine translation showed its emergence in 1987 after a number of modifications and research it again gained its popularity in 2013. Long Short Term Memory(LSTM)based Recurrent Neural Networks(RNN) were used in the paper [2] [15] to encode a variable-length source sentence into a fixed-length vector and to decode the vector into a variable-length target sentence. A similar 2-layer LSTM(Long Short Term Memory) based RNN encoder-decoder Neural Machine Translation System facilitating encoding of variable length source sentence into fixed length vector and decoding of fixed length vectors to get target sentence[6] can be used for Abstractive Text summarization as it can read articles of variable length and generate summaries based on learning mechanism of RNN. Recently, a number of papers have proposed the use of neural networks [15] [2] [16] [12] for machine translation and neural network based summarization, motivated us to use the same concept for text summarization. This neural machine translation approach typically consists of input layer, a number of hidden layers and output layer. In every layer it encodes a source sentence, corrects the error and then decodes to a target sentence .

Abstraction based summarization on news corpus and modeled as ABS and ABS+, achieved state of the art since it outperformed all previously published models on summarization [14] based on both abstractive and extractive, neural or non neural systems. Our work is closely related to [14] as we also performed on the same dataset along with some other news corpus to improve the efficiency of the system. We performed preprocessing of the raw data to increase the efficiency in terms of space and time complexity. We selected training, validation and test data randomly from each domain so that neither it under fits nor over fits. Finally, we trained and analyzed the result by varying in training and testing corpus as well as by varying different system parameters during system training.

3 System Description

In our system we have used OpenNMT, which is an open source toolkit for Neural Machine Translation. In this section we have discussed about OpenNMT, Data Preparation, Pre-processing, System Training and System Testing. Fig.3. shows different steps of our system.

3.1 OpenNMT

OpenNMT is an attentional sequence to sequence, Recurrent Neural Network(RNN) based neural machine, mainly consists of Pre-processing, Training and Transla-

tion unit. Neural machine achieved remarkable performance over human evaluation, rule based and Statistical Machine Translation(SMT) systems[18][6] in large-scale translation tasks such as machine translation, speech recognition and text summarization. Neural networks are appealing since it requires minimal domain knowledge and is conceptually simple[11]. It functions like a black box, when we feed in some inputs from one side, it generates outputs from the other side and the decision it makes is mostly based on the current inputs and previous stored output on LSTM(Long Short Term Memory), special kind of RNN.

LSTM (Long Short Term Memory):

LSTMs are the building block of recurrent neural network comprising of a cell, an input gate, an output gate and a forget gate. Each of these gates are neurons and the cells are the memory of LSTM responsible for remembering values over arbitrary time intervals. The idea behind LSTM based RNN is how human brain works, humans dont start their thinking from scratch every second. As we read we understand each word based on our understanding of current word and previous word knowledge. When we want to forget everything and want a fresh start, we simply can not forget by deleting everything and start thinking from scratch again, some of our thoughts are persistent though some we forget.

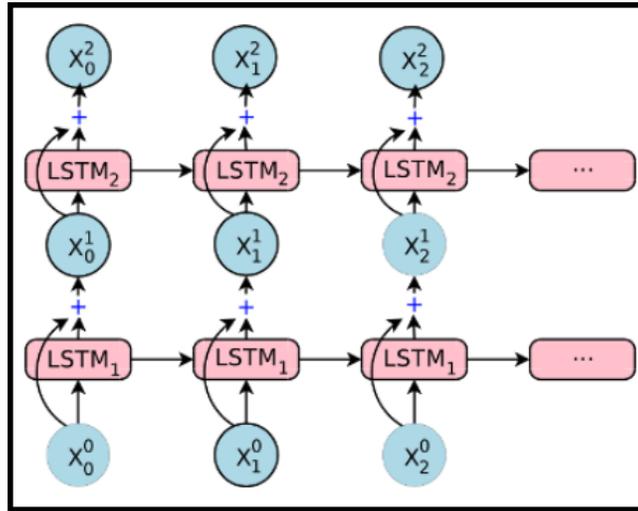


Fig. 1. Block diagram of the system

Same way LSTM works by storing the information in the memory after each and every iteration as well as after each epochs. LSTMs(Long Short Term Memory) are a capable of learning long-term dependencies. They were introduced by Hochreiter and Schmidhuber[6] work tremendously well on a large variety of

problems. Fig.1. shows the LSTM used in OpenNMT where at each iteration and every hidden layer it is used to store data.

Encoder and Decoder of RNN

The encoder consists of unidirectional RNN whereas decoder consists of unidirectional RNN with each hidden state size of same as that of encoder. An encoder in neural network reads and encodes a source sentence into a fixed-length vector until the end of the sentence reached and then starts to emit each target word at a time as fig.2 shows. The decoder computes the next hidden states from the previous states based on word embedding, previous target word, and the conditional input derived from the encoders output. OpenNMT uses sequence to sequence LSTM based RNN. While giving inputs as A, B, C and D in the input layer it generates X, Y and Z as summaries, after a number of hidden layer.

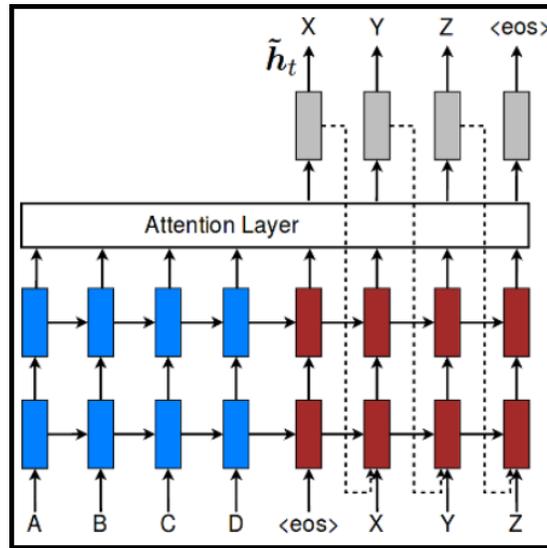


Fig. 2. Neural Machine Translation a stacking recurrent architecture for translating a source article A B C D into a target Summary X Y Z. Here, eos marks the end of a sentence.

Its attention allows to train the neural network by allowing models to learn alignments between different modules. Whole encoder decoder system, which consists of the encoder and the decoder for an input pair, is jointly trained to maximize the probability of a correct summary given a source article. In each layer the encoder reads the input sequence of the source sentence and generated a sequence of states. These source sentence is feed to two unidirectional stack

of LSTM based RNN for the next hidden layer. One stack content is used as it is and other is reversed and these stack layers are concatenated at each linear layer to yield the next layer.

Rare or Unseen Words:

In Abstractive Text Summarization key challenge is to understand the concepts by finding the key entities from the sentence around which the summary revolves. The vocabulary of words are prepared by the system during the training phase from the training articles. The vocabulary is a word vector having only those words of the training article. As a consequence, when LSTM based RNN maps the test article words to the vocabulary it finds some of the words are unknown. This new words or out of vocabulary words are called rare or unseen as they are unseen by the model. Hence RNN puts *< unk >* keyword as unknown token in place of those out of vocabulary words.

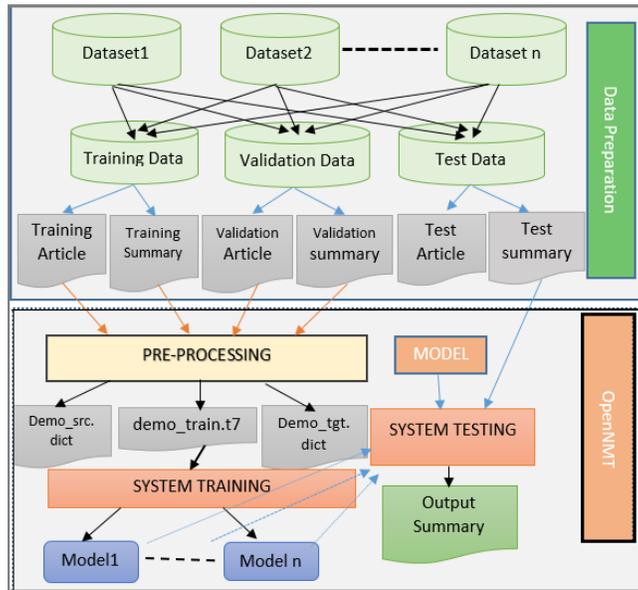


Fig. 3. Block diagram of the system.

3.2 Data Preparation

We performed on DUC 2003, DUC 2004 and Gigaword corpus having news data collected in the raw form. After collecting the raw corpus we converted it to text format for the convenient of our system. We then performed preprocessing of the raw data to increase the efficiency in terms of space and time by removing

noise and other irreverent data. The whole corpus collected was considerably large and hence required large amount of time to train the model. Due to this reason, for faster output generation and evaluation during system training we use a subset of collected corpus each time. To avoid over fitting we selected training and validation data from each domain in the form of article summary pair and the number of instances taken was of ten : one ratio between training and validation dataset. Similarly, for testing we extracted article summary pair. Source article to test the system and the corresponding summary with respect to system generated summary is used further to evaluate the systems accuracy with the help of human evaluator and ROUGE. Careful selection of training and validation dataset is required to avoid over fitting and under fitting. Input for pre-processing phase is hence, selected and prepared from the raw news corpus. Figure 3 shows the collection of datasets and selection of domain specific data for training, validation and testing is done for better results.

3.3 Preprocessing

OpenNMT is a standalone project with commonly used tools like language independent tokenizer which is simple, deterministic in nature and a language independent word embedding system [6]. The pre-processing phase tokenizes the datasets based on tab separation, and ensures article-summary mapping is present. It eliminates all the unmapped data pair because for system accuracy mapped parallel article-summary is essential. So during selection of data it is important to select parallel article- summary pair, i.e. each article with its summary.

The training and testing corpus we used for summarization is pre-processed before training and testing by OpenNMT. The pre-processing module of OpenNMT accepts the four text files for pre-processing. Namely trainArticle.txt, TrainingSummary.txt, validationArticle.txt and validationSummary.txt file which we prepared from the raw corpus during data preparation phase. It generates two human readable file demo.src.dict, demo.tgt.dict and one Torch file namely demo.train.t7 file for system training.

3.4 System Training

Output of pre-processing phase of OpenNMT, Torch file (containing all suitable data for training) is used for training model. Prior to training, data is shuffled and sorted to ensure instances in the training batch uniformly come from different parts of corpus. An epoch in training refers to one forward and one backward pass over all the training instances in batches.

Some of the important parameters are listed in table 1. system is trained for some fixed number of epochs, specified by end-epoch parameter. An epoch comprises of iterations and in each iteration is of one forward and one backward pass are performed over batch size number of training instances. System has been trained for 13 epochs in first experimental setup and 16 and 20 epochs in other

Table 1. Details of system parameters used in training

PARAMETER	MEANING OF THE PARAMETER
Epoch	one forward pass and one backward pass of all the training examples
Batch Size	the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
Iterations	number of passes, each pass using [batch size] number of examples. To be clear, one pass = one forward pass + one backward pass (one forward pass and backward pass as one passe).

setups. A validation score, dynamically computed using validation data, helps checking convergence of training. Different terminologies are discussed below:

3.5 System Testing

For system testing the source articles are pre-processed to generate tokens and word vector. The output of each epoch of system training is a model which is capable of generating summaries. System testing with the OpenNMT summarization system uses these trained model to predict summary for the test articles. Summary generation process makes use of beam search, a heuristic based optimized version of best first search, to search the best or optimal summary words. Fig. 2 shows the Effectiveness of search mechanism is exhibited in its ability to facilitate trade-off between summary generation time and search accuracy which is ensured by tuning beamSize option of beam search to a relatively small value. A comparison is done between test word vectors and the training system vocabulary to predict the output summary based on semantic similarity. The highest probable words are picked based on the prior knowledge as well as present input from the test article to build the summary. As in our corpus source sentences are of approximately of 30 words and expected target summary if of maximum 8 to 12 words. Besides, generator uses 'unk' symbol when it finds rare words. These rare or unseen words are not present in the training models vocabulary. So when mapping is done between the test article to the vocabulary these words are absent and hence system puts these as unknown symbol.

4 Experimental Design

We have conducted extensive analysis to better understand our models in terms of learning, the ability to handle different lengths of articles and choices of attentional architectures. This section contains detailed description about corpora and experimental setup used for training and testing Summarization effectiveness of Neural Network based summarization system.

4.1 Corpora Description:

In this series of experiments we used open source DUC² (Document Understanding Conferences) Corpus namely DUC 2003, DUC 2004 and Gigaword³ news corpora, comprises of parallel running source-target sentence pairs. Pre-processing of the corpus is necessary prior to training as the data contains noise and in xml format. Training data, a subset of news corpus containing 50000 instances, is used for training which is randomly selected from the above mentioned datasets and Validation data is also a subset of training corpus containing 5000 instances, is used for checking convergence of training. Besides, test corpus embodying 1000 article-summary has been used for testing summarization effectiveness of training models. Corpus details like name of corresponding corpus and size or number of instances present are discussed below:

4.2 Corpus Details

The DUC corpus of 2004 consists of 624 article-summary pair and DUC 2003 consists of 500 article-summary pairs. Annotated English Gigaword was developed by Johns Hopkins, Universities Human Language Technology Center of Excellence. Annotated English Gigaword contains the nearly ten million documents (over four billion words) is collected from the original English Gigaword, fifth edition from seven news sources namely Agence France-Presse, English Service , Associated Press Worldstream, English Service, Central News Agency of Taiwan, English Service, Los Angeles Times/Washington Post Newswire Service, Washington Post/Bloomberg Newswire Service, New York Times Newswire Service, Xinhua News Agency, English Service. Gigaword dataset is available in .xml format is required to convert in .txt format for the convenience of our system.

4.3 Experimental Setup

We have used following different experimental setups to train, test and analyze systems performance from different perspectives.

1. Initially, we trained the NMT based summarization system using randomly selected parallel source-target instances comprises of 10000 training and 1000 validation corpora. Training is done for 13 epochs. This Trained models were tested using 1000 sentences selected based on the same domain of training data. Result sets containing system generated summaries and target summaries were evaluated using ROUGE evaluation for each epoch and further analyzed.
2. We have re-trained the system using 55000 corpus selected from various corpus and saved the trained model obtained at 16 different epochs after analyzing the system result of first system. Each of these 16 models have been tested using test corpus and prediction results have been subjected to

² <http://duc.nist.gov/>

³ <https://catalog ldc.upenn.edu/LDC2002T31>

ROUGE evaluation. Such a setup helps us to analyze change in summarization behavior of neural machine based summarization system with increase in number of epochs and training datasets.

3. After successful training of both the model we created a test dataset of 100 relevant sentence for faster output generation. These sample sentences are tested using 13 epochs of two system. This helped us to compare both the systems performance and the output result score using ROUGE are plotted in graph are shown in fig.4.
4. Furthermore, we have created different test sets from the original test data, each test set containing 50 sentences. Average length of sentences in the three test datasets is 8, 15 and 20 respectively. We selected best two model from each system based on previous results and these models are tested using the three test datasets and prediction results have been evaluated using ROUGE evaluation. Such a setup helps us asses relationship between summary performance and average length of sentences in test dataset. All these summaries are analyzed by comparing with target summary as well as source article.

The results of all these experimental setups have been detailed and analyzed in Section 5.

5 Result Analysis

Prediction results of our experiments were self-analyzed and evaluated using ROUGE evaluation[9]. Human evaluators assessed quality of summary with respect to adequacy, length and overall rating. They compared system generated summary and target summary of test datasets.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) includes measures to automatically determine the quality of a summary by comparing it to system generated summaries to target summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the systems generated summary to be evaluated and the ideal summaries created by humans. We used ROUGE 2.0 to evaluate our system, where we learned the ROUGH L, ROUGE 2, and ROUGE 1 score of comparison of summaries in terms of Precision, Recall and F-score.

Precision(P): It the positive predictive value i.e. fraction of relevant instances among the retrieved instances. Precision helps us to predict how many words are correct out of all the system generated summary words. In fig.5 Precision is TP divided by TP and FP. Which is the number of words occurring in both system and target words i.e. intersection of words between both the summaries divided by the number of words in the system summary.

$$Precision(P) = \frac{\text{number of identicle words in both the summary}}{\text{number of words in the system generated summary}}$$

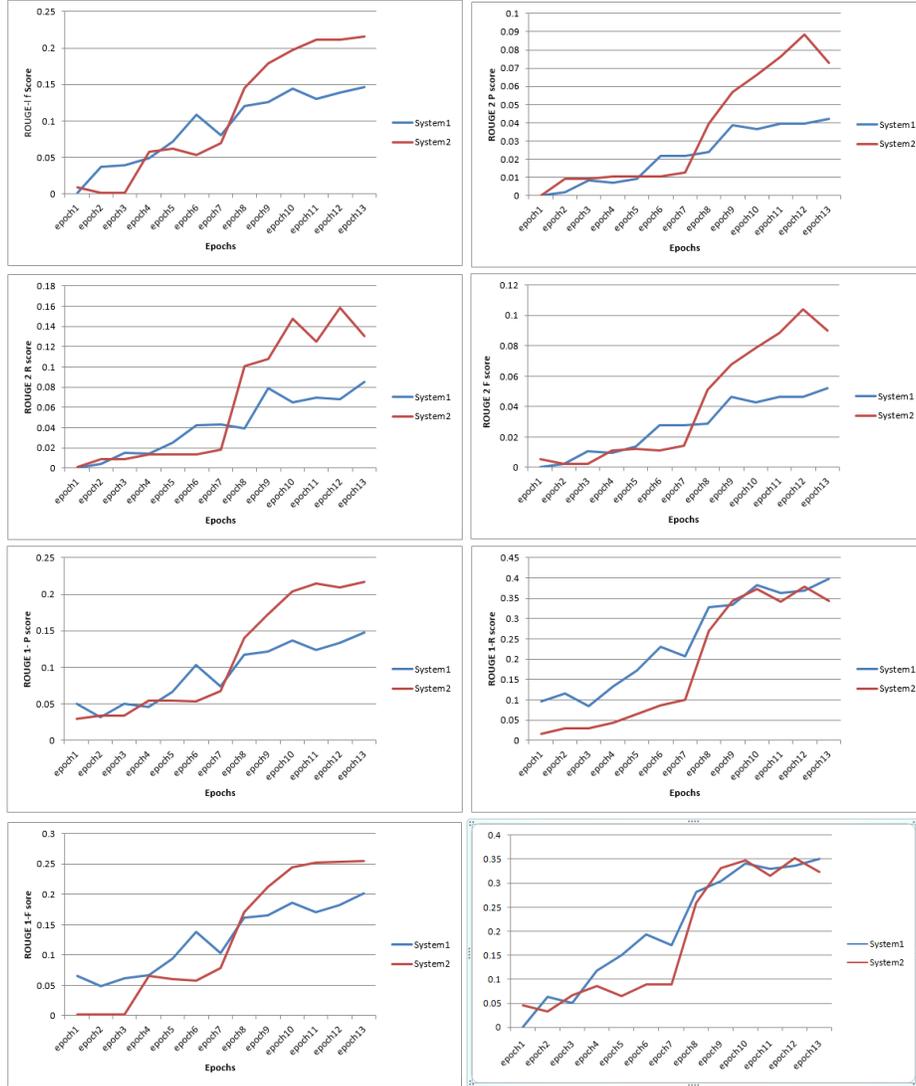


Fig. 4. ROUGE score achieved by system1 and system2 - shown are images of the attention weights learned by various rough score. from the top left ROUGE L-F Score, ROUGE 2-P Score, ROUGE 2-R Score, ROUGE 2-F Score, ROUGE 1-P Score, ROUGE 1-R Score, ROUGE 1-F and ROUGE L-R Score respectively

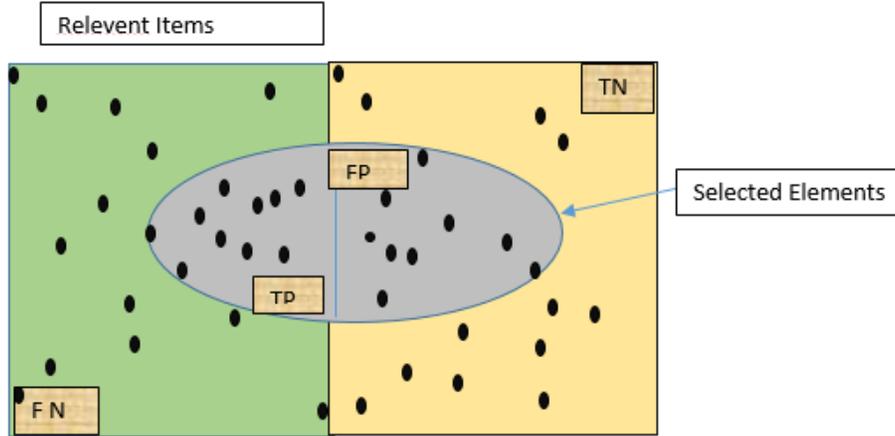


Fig. 5. Data Overview: total data can be divided into four group, TP(True Positive)TN(True Negative), FP(False Positive) and FN(False Negative)

Recall (R): Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Recall helps us to predict how many words are correctly identified out of all the target summary words. In fig.5 Recall is fraction of TP and summation of TP and FN. Which is the number of words occurring in both system and target summaries i.e.intersection of words between both the summaries divided by the number of words in the ideal summary.

$$Recall(R) = \frac{\text{number of identical words in both summary}}{\text{number of words in target summary}}$$

F-score: It is a composite measure that combines precision and recall which depicts how well our system performs. The basic way to compute the F-score is to count harmonic average of precision and recall:

$$F - score = 2 * \frac{P * R}{P + R}$$

ROUGE-N, ROUGE-S and ROUGE-L are the granularity of texts being compared between the system summaries and reference summaries. For example, ROUGE-1 refers to overlap of unigrams between the system generated summary and target summary. ROUGE-2 refers to the overlap of bigrams between the system generated summaries and target summaries. ROUGE-L measures longest matching sequence of words between both the summaries. For example, for this two sentences given below:

System Summary : india vs pakistan

Target Summary: india wins the match against pakistan

Here in the given instances, System Summary has 3 words, Target Summary has 6 words and the number of common words between the System Summary and Target Summary is 2. ROUGE-1 refers to overlap of unigrams between two summary. So for ROUGE-1 corresponding recall, precision and F-score will be:

$$ROUGE - 1recall = \frac{2}{6} = 0.33$$

$$ROUGE - 1precision = \frac{2}{3} = 0.66$$

$$ROUGE - 1F - score = 2 * \frac{(0.33*0.66)}{0.33+0.66} = 0.44$$

If target summary is larger than system summary precision may give good score, and if system generated summary is larger than recall will give good score. finally the score generated at each testing is the average of individual scores. So, we get nine score at every output P,R, and F of ROUGE-L, ROUGE-1 and ROUGE-2 each. Fig.4. shows comparison of ROUGE L-F Score, ROUGE 2-P Score, ROUGE 2-R Score, ROUGE 2-F Score, ROUGE 1-P Score, ROUGE 1-R Score, ROUGE 1-F and ROUGE L-R Score between two best systems 13 epochs.

Table 2. shows ROUGE score achieved by two system. Highest ROUGE score of 39.89 is attained at epoch 13 of System1 and ROUGE score of 37.04 14th epoch of System 2. The ROUGE score curve converges 11th epoch, we also analyzed that for short sentences we got better results of 42.03.

After analyzing all the epochs output with 100 test data, we Further, have created different test sets from the original test data, each test set containing 50 sentences. Average length of sentences in the three test datasets is 8, 15 and 20 words respectively. Table 2 depicts some of the analyzed results. We came to conclusion that small sentences gives batter summary in terms of ROUGE score.

Table 2. Best Experimental results of our system

System Used	Test Data	Best ROUGE score
system 1 13 epoch	100 sentence	39.89
system 2 14 epoch	100 sentence	37.04
system 2 14 epoch	50 sentence	39.89
system 2 14 epoch	50 short sentence	42.03

Table 3 shows some of the system generated summaries of both the system generated from the same source sentences. After analyzing results we came to conclusion that though summaries are not exactly same to the target summary or same in all the systems but summaries generated by the best models(epoch) are semantically correct to a large extent. There result is not accurate because ROUGE score calculates are not that accurate based on number of N gram or sequence matches.

Table 3. Sample Summary Predictions

Prediction by System 1	Prediction by system 2
palestinian minister heads for	palestinian fm says he will not stand down
french prime minister arrives in rome	french prime minister arrives in sarajevo
malaysia 's lavrov calls for international cooperation	malaysia 's defense minister calls for reconciliation
nigerian central banks step up more than ## million dollars	nigerian central african leaders to discuss ceasefire
suicide bomber kills ## in turkey	## killed in iraq chopper crash

6 Conclusion and Future Works

In this paper, we applied the attention based sequence to sequence Recurrent Neural Network for Abstractive text summarization. Attentional Recurrent Neural Network allows to train the neural network by allowing the models to learn alignments between different modulus using LSTMs(Long Short Term Memory). It functions like a black box, when we feed in some inputs from one side, it generates some outputs from the other side and the decision it makes is mostly based on the current inputs and previous stored output based on LSTMs which are a special kind of RNN.

RNN based summarization system relies heavily on size of the training corpus which motivated us to use a significant amount of training corpus. Effectiveness of summarization is largely determined by attention mechanism and the score function used for computing attention of each hidden state and error correction. A practiced and careful selection of values for system parameters such as number of epochs, batches, hidden layers etc. can also significantly improve the summarization quality. We have trained, tested and analyzed the proposed systems for summarization using various news summarization dataset. Predicted summaries have been evaluated using ROUGE evaluation. Human evaluator analyzed the quality of summarization in terms of its adequacy, quality and redundancy and found that the after a certain number of epochs, the trained models are able to produce semantically correct summaries.

In the next work, we will extend our efforts on this corpus and build more robust models compared to our baseline system for more accurate summary generation. The model can be further extended to multi-lingual and multi-document automatic summarization tasks.

References

1. Baxendale, B, P.: Machine-made Index for Technical Literature an Experiment. IBM Journal of Research and Development 2(4), 354–361 (1958)
2. CKyunghyun Cho, Bart van Merriënboer, C.G.D.B.F.B.H.S., Bengio, Y.: Learning Phrase Representations using RNN encoder-decoder for Statistical Machine Translation. arXiv preprint arXiv:1406.1078 (2014)

3. Clarke, J., Lapata, M.: Discourse Constraints for Document Compression. *Association for Computational Linguistics* 36(3), 411–441 (2010)
4. Edmundson, P, H.: New Methods in Automatic Extracting. *Journal of the ACM (JACM)* 16(2), 264–285 (1969)
5. Eduard, H., Chin-Yew, L.: Automated Text Summarization and the SUMMARIST System. In: *Proceedings of a Workshop on held at Baltimore, Maryland: October 13-15, 1998*. pp. 197–214. Association for Computational Linguistics (1998)
6. Guillaume, K., Kim Yoon, Deng Yuntian, S.J., M, R.A.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. arXiv preprint arXiv:1701.02810; This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC- BY-ND (2017)
7. Hal, D.I., Daniel, M.: A Noisy-channel Model for Document Compression. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002*. pp. 449–456. Association for Computational Linguistics (2002)
8. Lin, Chin-Yew: Rouge: A Package for Automatic Evaluation of Summaries. In: *Text summarization Branches Out: Proceedings of the ACL-04 workshop*. vol. 8. Barcelona, Spain (2004)
9. Lin, C.Y.: Improving Summarization Performance by Sentence Compression: a pilot study. pp. 1–8. Association for Computational Linguistics (2003)
10. Luhn, Peter, H.: The Automatic Creation of literature Abstracts. *IBM Journal of Research and Development* 2(2), 159–165 (1958)
11. Minh-Thang, L.H.P., Manning, C.D.: Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing journal, Association for Computational Linguistics* pp. 1412–1421 (2015)
12. Prasad Rajesh, K.U., Jayashree, P.: Machine Learning in Evolving Connectionist Text Summarizer. *Anti-counterfeiting, Security, and Identification in Communication*, 2009. pp. 539 – 543 (2009)
13. Rajesh, P., U.V., K.: Implementation and Evaluation of Evolutionary Connectionist Approaches to Automated Text Summarization pp. 539 – 543 (09 2010)
14. Rush Alexander M, C.S., Jason, W.: A neural Attention Model for Abstractive Sentence Summarization. arXiv preprint arXiv:1509.00685 (2015)
15. Sutskever Ilya, V.O., V, L.Q.: Sequence to Sequence Learning with Neural Networks. pp. 3104–3112. Curran Associates, Inc. (2014)
16. Tara N Sainath, Oriol Vinyals, A.S., Sake, H.: Convolutional Long Short Term Memory, Fully Connected Deep Neural Networks. *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* pp. 4580–4584 (2015)
17. Wang Shuai, Zhao Xiang, L.B.G.B., Daquan, T.: Integrating Extractive and Abstractive Models for Long Text Summarization. pp. 305–312. IEEE (2017)
18. Wu Yonghui, Schuster Mike, C.Z.L.Q.V.N.M.M.W.K.M.C.Y.G.Q.M.K.: Google’s Neural Machine Translation System: Bridging the gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144 (2016)