Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Selective Multi-head Attention for Abstractive Text Summarization using Coverage and Pointer Generator

Pramod Kumar Amaravarapu[1], Akhil Khare[2]

[1]Assistant Professor, Dept. of CSE, Matrusri Engineering College, Research Scholar, Osmania University.

[2]Professor, Dept. of CSE, MVSR Engineering College, Hyderabad

---

Abstract

Text summarization produces a concise version of text without reusing the phrases from the original text, while retaining the context and key contents. In this work, we present seq2seq architecture for lengthy text summarization. The previous abstractive summarization models have generated summaries, but these models suffer from duplicates and semantic irrelevance. The primary reason for this is that the source text being summarized is longer and typically contains multiple sentences. It also includes a significant amount of repetitive information. We propose selective multi-head attention using coverage and pointer generation for summarization to handle the problems. The selective mechanism helps to improve the encoded representation by making it more accurate. The repetitions are controlled by the coverage mechanism in multi-head attention by tracking the tokens, which have been summarized. The pointer network is also integrated into the multi-head architecture which handle out-of-vocabulary problem. The experimentaion is carried on CNN/DM standard dataset. The suggested model outperforms the baseline extractive and abstractive summarization, according to the finds.

Keywords Multi-head attention; coverage mechanism; pointer-generator; abstractive summarization

1.0 Introduction

The automatic summarization of text, summarizes the source text into condensed text while maintaining the semantics of the original text. This task is achieved by two approaches, one is extractive summarization and another one is abstractive summarization. The extractive summarization extracts essential features from the original input text and uses them to summarize the text. The abstractive text summarization understands the input text and generates a grammatically correct summary. There is a lot of room for improvising abstract summarization, which is our primary direction in the research. In a few years, the success of seq2seq architecture became helpful in speech recognition, machine translation, and question answering tasks. This architecture relies on a context vector generated from the source and target text, which causes the loss of information. Therefore, it is affecting the quality of the summary that is generated. To

overcome the information loss problem in abstractive summarization, many proposed attention-based seq2seq models [1, 2, 3] performed better than the former methods, but they still have problems. For longer text, it loses the crucial information from the input text. At the same time, the decoder generates identical words, which leads to repetition of the phrase.

Recently [4] presented a Transformer architecture using multi-head attention, which outperformed CNN and RNN in various NLP problems, like machine translation [4, 6], sentiment analysis [7], and dialogue systems [6]. The transformer model is non-recurrent, and multi-head attention handles longer text easily. The Transformer model generates the target summary, and it doesn't have the capability of filtering the nonprimary information in the long input. It doesn’t continuously track generated summaries in the decoding. Consequently, when it comes to abstractive summarization, the transformer model suffers from repetition and semantic irrelevance.

To deal with these issues, we offer a selection and coverage mechanism. Abstractive summarization is accomplished by the use of a multi-head transformer. This selection mechanism extracts the salient information from the source and produces two-level representations, which increase the overall semantic quality of the original input. The repetition is reduced as a result of the covering. Using the attention distribution summarization algorithm, the coverage mechanism calculates the coverage vector. Using this coverage vector, the next time step will be able to calculate the new attention distribution. We test the suggested technique on CNN/DM dataset.

2.0 Related work

The section contains the overview of literature work, which includes abstractive summarization of text using seq2seq neural and attention mechanism. With the advancement of deep learning algorithms, neural network-based abstractive summarization has emerged in NLP. Seq2seq architecture is the most often used technique for machine translation. Later it became famous for abstractive summarization. Initially, Rush et al. [8] applied seq2seq neural attention architecture with attention based encoder and feed-forward decoder for abstractive summarization on annotated English Gigaword [9] and produced the best results in their field. Chopra et al.[10] modified the architecture with CNN encoder and RNN decoder, and the model outperformed the previous models. Nallapati et al.[2] proposed RNN based sequence-to-sequence by replacing both encoder and decoder with RNN, which improved the performance. RNN and CNN are used in encoders and decoders [11]. The LSTM (Long Short-Term Memory) [12] and the GRU (Gated Recurrent Unit) [13] are the two unique RNN designs that are most commonly employed in encoders and decoders, respectively. In the generated summary, there are too many repetitions and too many OOV (out-of-vocabulary) words, making it difficult to read for humans.

Attention technique plays an important role in machine translation [4, 6] and abstractive summarization [1, 2, 3]. The attention mechanism highlights the relevant features of the source input dynamically. This mechanism failed to generate OOV and rare words. [14] proposed pointer network to handle rare words and OOV by copying the tokens from the source input using attention weights. They also included a coverage mechanism to handle the repetition of

tokens in the target output. This [15] proposed technique initially which extracts the essential information from original source input. Then the dual-attention seq2seq architecture generates the condition from source text and extracted facts. Still, there are many problems associated with abstractive text summarization. Self-attention was proposed [16] to represent the single text sequence. [4] proposed multi-head attention by adapting self-attention for the task machine translation. Multi-head attention focuses on different location information from different representation subspaces. There is evidence from many researchers that multi-head attention enhances the performance of sequential tasks such as the dialogue system [3], semantic role labeling [17], abstractive text summarization [2], and clinical data analysis [18].

"The extent to which multi-head attention is transparent in abstractive summarization has been investigated" [19]. Also introduced quantitative metrics showing that multi-head attention is partially interpretable. We were able to ablate fewer heads without sacrificing summarization performance by using a sparsemax activation function instead of a softmax activation function. [20] introduced multi-head attention summarization, which uses LSTM with multi-head attention and pointer generator to avoid duplicates. This model improved the performance of the abstractive summarization. [21] introduced coverage and selective "multi-head attention" [4] for abstractive summarization, which uses LSTM with "multi-head attention"[4]. This has improved the representation and removed repetitions.To improve the algorithm's performance, we integrate selective gate and coverage mechanisms into the multi-head attention along with pointer generator.

3.0 Methodology

The following steps in Fig. 1 is are followed in implementing multi-head attention for abstract summarization

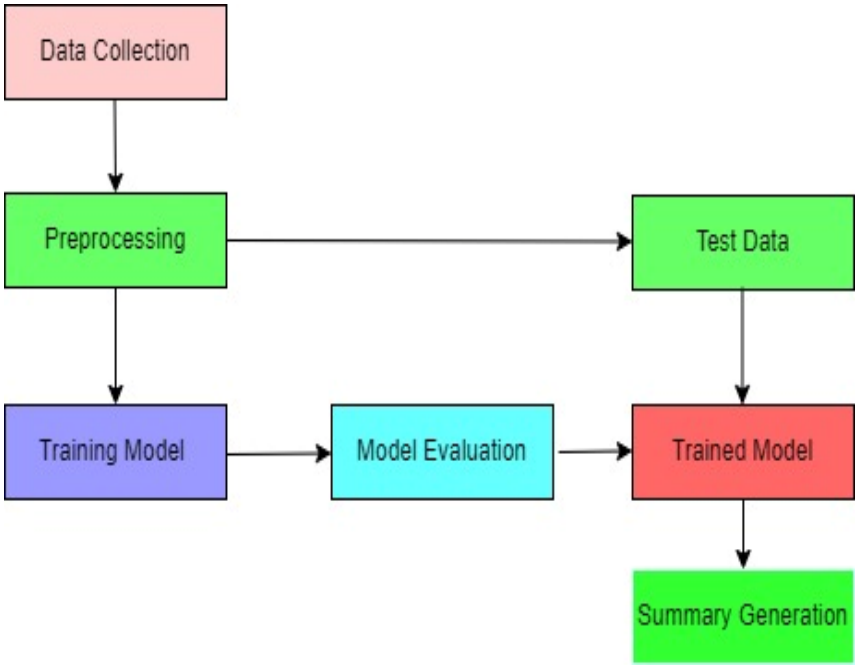

Fig. 1 System Architecture

3.1 Dataset and Preprocessing

We use the CNN/DM dataset for summarization, consisting of 287227 training, 13368 validation, and 11490 testing samples. Each of the articles in the dataset is associated with a handwritten multi-sentenced summary. The average tokens in input text and target summary are 791 and 56, respectively. The details of the dataset is presented in the Table. 1.

Table. 1 CNN/DM Dataset Statistics

Dataset	CNN/DM		
	Train	Valid	Test
Source docs	287227	13368	11490
Target summary	287227	13368	11490
Avg. docs len(sen)	39.8	33.6	34.2
Avg. docs len(word)	790.4	768.9	777.9
Avg. summary len(sen)	3.7	4	3.9
Avg. summary len(word)	55.2	61.5	58.4

3.2 Model Architecture

The attention mechanism is playing major role in machine translation, and abstractive summarization. It is proved that the Transformer-based models outperformed the sequence models. The transformer-based architecture includes encoder and decoder layers to process input and produce output depending on the task at hand. The encoder stack transforms the information into a context vector, and the decoder stack converts the context vector representation into the target output. The original Transformer architecture is presented in Fig. 2. In that both encoder and decoder parts uses only linear layers. The architecture of the modified transformer is presented in the Fig. 3. In this modified architecture we use LSTM layers for processing source and target data.

Encoder and decoder

The proposed Transformer architecture make up of encoder part and decoder part. The proposal adds selective gate and coverage as a component in the decoder part. The encoder part and decoder part of the architecture use LSTM. The proposed model's architecture is presented in Fig. 1. The summarization dataset consists of N data entriess. Each entry (x, y) consists of input x and output y. The encoder processes the source input $x = (x_1, x_2, \dots, x_I)$ and the target $y = (y_1, y_2, \dots, y_T)$ is predicted by the decoder. The target length T is always smaller than the source length I. The Transformer encoder and decoder architecture are presented in Fig. 2. The encoder reads the input $x = (x_1, x_2, \dots, x_I)$ and generates the context vectorh^e. The decoder generates the vector h^d from the context vector h^e and target text.

Multi-Head Attention

"An attention function can be described as mapping query and a set of key-value pairs to an output, where query, keys, values, and output are all vectors. The output is computed as a

weighted sum of the values, where the weight assigned to each value is computed by a compatability function of the query with the corresponding key" [4]. The basic attention is

performed parallelly with different dimensions. The results are concatenated and projected again. We compute the multi-head attention for the given query Q and key-value pair (K, V) as follows

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)$$

Where W_i^Q, W_i^K, W_i^V are learnable parameters, and d_k is the dimension of the key K. For each time step, the attention distribution over the query and keys is computed as

$$a_t = Softmax(\frac{q_t K^T}{\sqrt{d_k}})$$

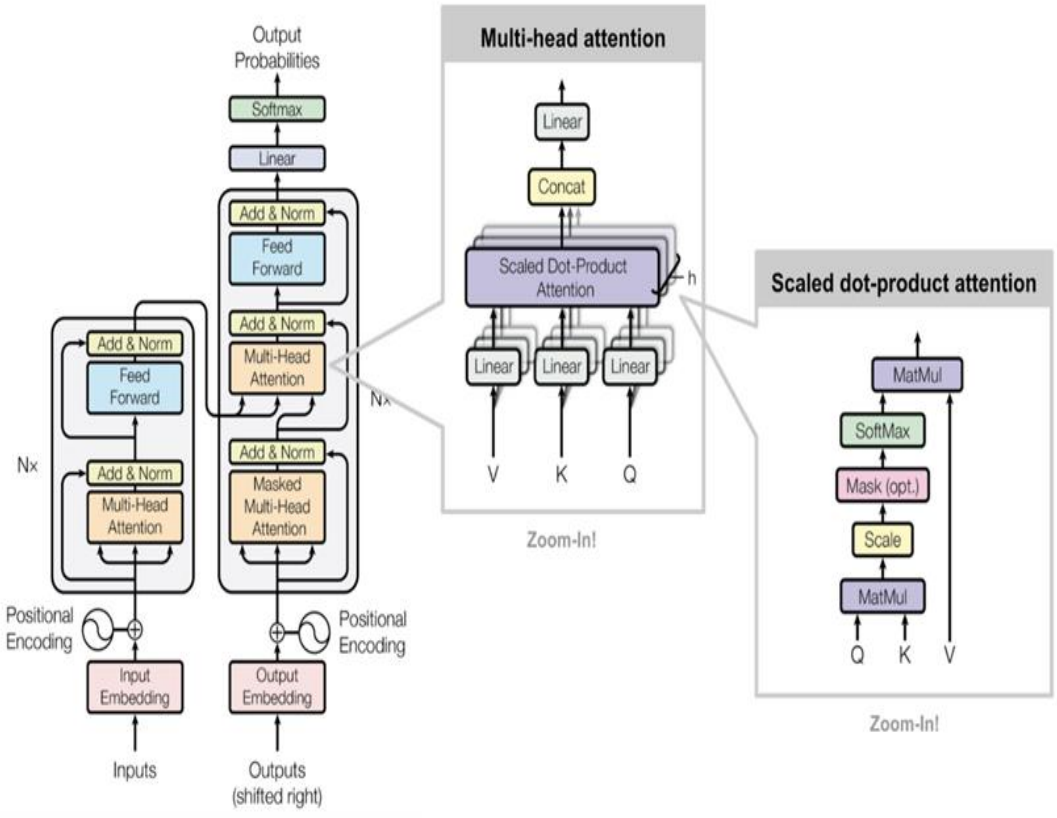

Fig. 2 Transformer Architecture [4]

Selective Mechanism

The most challenging aspect of abstractive summarization is filtering the nonprimary information from the source input while generating the target. Transformer architecture doesn't include any mechanism to achieve this task. In multi-head attention we have the selective gate. The selective gate is included at the decoder's multi-head, which filter's out the nonprimary information from the encoder context h^e. The query input only focuses on primary information in the key-value pairs of the encoder context h^e. The select gate initially uses 1-D convolution and extracts N-gram features and computes $gate_t$ for each time step using sentence representation h_t^e and N-gram feature tailored vector h_t' as

$$C_t = ReLu(\left[h^e_{t-\frac{k}{2}}, \dots, h^e_{t+\frac{k}{2}}\right] W_C + b_C)$$

Where W_C, b_Clearnable parameters and k are is the kernel size.

$$gate_t = sigmoid(h_t^e W_g + C_t U_g + b_g)$$

$$h_t' = h_t^e . gate_t$$

Where . is point wise multiplication and W_g, U_g, b_g learnable parameters. The selective gate highlights the core information. The sigmoid function output 0 or 1. If the output is 1, it highlights the core information otherwise ignores it.

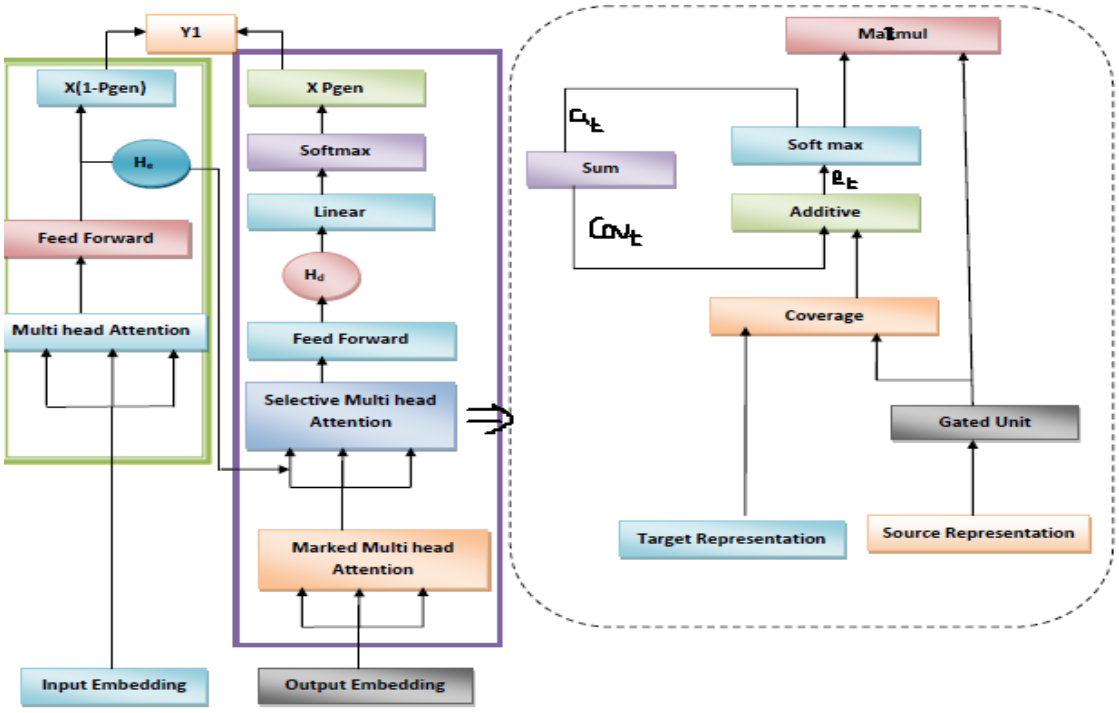

Fig. 3 Modified Transformer Architecture

Coverage Mechanism

The coverage mechanism was introduced in [3] and we include this to handle repetitions in the "multi-head attention" [4]. The coverage vector is computed as

$$cov_t = \sum_{i=0}^{t-1} a_i$$

At each time step t, the attention distribution a_t is computed between the query and key. Also the coverage loss is included in the coverage mechanism to penalize the repeated attentions on the exact words.

Pointer Generator Network

To handle OOV, we add a pointer generator at the decoder of the transformer architecture. It allows copying of the tokens from the actual source content. The pointer network generates the words, either pointing or rendering tokens from the predefined vocabulary. The probability p_{gen} is calculated using decoders state h^d, encoders context vector h^e and decoders input y_t as

$$p_{gen} = sigmoid(h^e W_e + h^d W_d + x_t W_x + b_{gen})$$

The generation probability p_{gen} acts as a soft switch, and it selects a word either from the vocabulary or copying from the input sequence.

4. 0 Experiments

4.1 Experimental Details

In this paper, we considered the size of vocabulary as 50000. We also truncated source and summary tokens to 512 and 128 during training and testing. The word embedding dimension is 256, and it is initialized with GolVe embedding's. The hidden size of the encoder side LSTM and decoder side LSTM is 256. The batch size is 32, and we used Adam optimizer, with β_1= 0.9, $\beta_2 = 0.9$, $\epsilon = 10^{-6}$ and learning rate $\alpha = 10^{-3}$. Eight heads are utilized in multi-head attention. In the training phase, the decoder employs the top generated and ground truth as teacher forcing for target generation. While prediction beam search is used with beam size 5.

4.2 Evaluation Method

The results are evaluated by using ROUGE [18] metric. This is one of the standard metric used in summarization tasks. It is done based on n-gram matching between model predicted summary, and ground truth summary. ROUGE-1, ROUGE-2, and ROUGE-L are computed by using pyrouge package for unigram, bigram and longest common sequence respectively. In the context of ROUGE, recall means how much of the reference summary is recovered or captured by the generated summary.

$$Recall = \frac{number\ of\ overlapping\ words}{total\ words\ in\ reference\ summary}$$

In the context of ROUGE, precision measures how much of generated summary was in fact relevant or needed.

$$Precision = \frac{number\ of\ overlapping\ words}{total\ words\ in\ generated\ summary}$$

4.3 Baseline Models

The performance comparison is done between the proposed model, and baseline models. The evaluation is done on the CNN/DM dataset. For the CNN/DM dataset, RNN with encoder and decoder, pointer-generator, pointer-generator with coverage, intra-attention with RL, intra-attention with RL and ML, Transformer, Transformer with coverage, Transformer with pointer-generator, coverage and selection models.

4.4 Result Analysis

The ROUGE scores of our model and baseline models were computed on CNN/DM dataset. The finding's of proposed model outperformed the baseline models. The results show that the transformer model is adequate for abstractive text summarization. The proposed Transformer with selective coverage and pointer generator outperformed the baseline models. The ROUGE scores for CNN/DM dataset is presented in the Table. 2, Table. 3 and Table 4. The results in Table. 2 show that reinforment learning with intra attention beat the seq2seq attention with pointer generator and coverage model. It also shows that Transformer with selective coverage beat the reinforcement learning with intra-attention model. The Transformer with selective coverage and pointer-generator beat the Transformer with selective coverage model.

Table. 2 ROUGE scores on CNN/DM dataset

Reference	Model	ROUGE-1	ROUGE-2	ROUGE-L
Nallapati et al [2]	words-lvt2k-temp-att	35.46	13.30	32.65
See. A et al [3]	Pointer-generator	36.44	15.66	33.42
	Pointer-generator+coverage	39.53	17.28	36.38
Paulus et al [22]	Intra-attention+RL+ML	39.87	15.82	36.90
Paulus et al [22]	Intra-attention+RL	41.16	15.75	39.08
Zhang rt al. [21]	Trnasformer+selective+coverage	41.35	16.1	39.27
Proposed	Trnasformer+selective+coverage	42.07	16.35	40.1

Our proposed model beats the baseline models as it handles repetitions using coverage mechanism, target generation using a selective mechanism, and out of vocabulary using pointer generator. It also handles the longer documents with multi-head attention. The multiple attention heads attend the parts of the subsequence differently and also take care long term dependencies and short-term dependencies.

Table. 3 ROUGE Precision(%) on CNN/DM dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	38	15.24	36.4
Transformer+coverage	38.7	15.62	36.48
Transformer+selective	38	15.4	36.4
Transformer+selective+coverage	40.2	16.1	39.3
Transformer+selective+coverage+pointer-generator	42	16.3	40.1

The ROUGE Precision and Recall percentage is presented in Table. 3 and Table. 4 respectively. It shows that significant improvement in the results from Transformer model to Transformer with selective, coverage and pointer-generator model. This improvement is due to the elimination of repetations, OOV and better representation of data.

Table. 4 ROUGE Recall(%) on CNN/DM dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	36	14.94	35.04
Transformer+coverage	37.8	15.04	36.5
Transformer+selective	38.1	15.34	36.68
Transformer+selective+coverage	40.4	15.78	38.95
Transformer+selective+coverage+pointer-generator	41.62	16.02	39.56

5.0 Conclusion

The proposed transformer model uses multi-head attention, which uses sequence-to-sequence architecture. Our proposed models outperformed with higher ROUGE scores and also handled the problems of out of vocabulary, effective generation of summary and duplicates. The encoder multi-head attention learns the source text structure and generates the context vector. The decoder multi-head attention improves speed per time step and generates the target by avoiding duplicates. The experiment is conducted on the CNN/Daily Mail dataset. The results show that the proposed multi-head attention-based Transformer outperformed the existing models on summarization of long documents. It has generated a human-like summary.

References

[1]. A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379–389.

[2]. R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang et al., “Abstractive text summarization using sequence-to-sequence rnns and beyond,” in Proceedings of the 20th SIGNAL Conference on Computational Natural Language Learning, 2016, pp. 280–290.

[3]. A. See, P. J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1073–1083.

[4]. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.

[5]. Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., Yan, R.: Get the point of my utterance! Learning towards effective responses with multi-head attention mechanism. In: IJCAI. pp. 4418-4424 (2018)

[6]. Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., Liu, Y.: Improving the transformer translation model with document-level context. In: EMNLP. pp. 533-542 (2018)

[7]. Letarte, G., Paradis, F., Gigu'ere, P., Laviolette, F.: Importance of self-attention for sentiment analysis. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 267–275 (2018)

[8]. A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379–389.

[9]. Napoles, C., Gormley, M., Van Durme, B.: Annotated gigaword. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. pp. 95–100. ACL (2012)

[10]. Chopra, Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the NAACL: Human Language Technologies. pp. 93-98 (2016)

[11]. J. L. Elman, “Finding structure in time,” Cognitive science, vol. 14, no. 2 (1990), pp. 179–211.

[12]. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735– 1780, 1997

[13]. J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555 (2014).

[14]. A. See, P. J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1073–1083.

[15]. B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, “Temporal attention model for neural machine translation,” CoRR, vol. abs/1608.02927 (2016).

[16]. Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang, "Distraction-based neural networks for modeling documents," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial IntelligenceAAAI Press (2016), pp. 2754–2760

[17]. Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, “Deep semantic role labeling with self-attention,” pp. 4929–4936, 2018

[18]. H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, “Attend and diagnose: Clinical time series analysis using attention models,” pp. 4091–4098, 2018.

[19]. C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” Text Summarization Branches Out, 2004.

[20]. J. Baan, M. ter Hoeve, M. van der Wees, A. Schuth, and M. de Rijke, Understanding Multi-Head Attention in Abstractive Summarization. 2019.

[21]. J. Li, C. Zhang, X. Chen, Y. Cao, P. Liao and P. Zhang, "Abstractive Text Summarization with Multi-Head Attention," 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8851885

[22]. X. Zhang and G. Liu, "Selective and Coverage Multi-head Attention for Abstractive Summarization," Journal of Physics: Conference Series, vol. 1453, p. 012004, Jan. 2020, doi: 10.1088/1742-6596/1453/1/012004.

[23]. R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," arXiv preprint arXiv:1705.04304, 2017.