**Multidisciplinary Journal of Engineering and Technology**
**ISSN: 3007-6900**
**DOI: 10.61784/mjet3004**

# SISM: A SELF-INTERACTIVE APPROACH TO WEB NEWS SUMMARIZATION USING DEEP LEARNING

Aleksy Nowaks
*Faculty of Mathematics, Informatics and Mechanics, The University of Warsaw, Poland.*
*Corresponding Email: alexnow8392@uw.edu.pl*

**Abstract:** The exponential growth of online information has intensified the need for efficient web news summarization techniques. This paper introduces the Self-Interactive Summarization Model (SISM), a novel approach that combines advanced deep learning methods with an innovative refined tuning process. SISM employs a two-stage strategy: extensive pre-training on a diverse dataset, followed by a specialized fine-tuning phase. We present a new, carefully curated dataset that reflects the varied nature of web news articles, enabling comprehensive model evaluation. Our experiments demonstrate SISM's superiority over existing state-of-the-art models, with significant improvements in ROUGE scores across multiple test sets. The study highlights the critical role of our refined tuning process in enhancing summarization quality and adaptability to diverse news content. SISM's performance underscores its potential to advance the field of automated web news summarization, offering more accurate and contextually relevant summaries.
**Keywords:** Web news summarization; Deep learning; Self-Interactive summarization model; Natural language processing

## 1 INTRODUCTION

In the current era of rapid internet development and explosive growth of textual data, the need for automatic text summarization systems has become increasingly urgent. Automatic summarization systems can independently and quickly determine the main content of a document to generate a summary, providing convenience for rapid information and intelligence acquisition. Extractive summarization refers to generating a summary by extracting sentences from the document that significantly contain the main information [1-2].

In extractive summarization methods, the encoder-decoder structure has been effectively applied [3-4]. Additionally, the attention mechanism in deep learning can help models discover parts containing important document information, which has a significant effect on improving summarization performance [5-6]. However, existing automatic summarization models based on the encoder-decoder structure focus too much on the decoder part [7-9], concentrating on obtaining sentences more relevant to the source document without mining richer document information in the encoder and ignoring the connections between different sentences. Since the association information between sentences plays an important role in selecting diverse information from documents, it is necessary to obtain document information as completely as possible when performing automatic summarization.

Based on these considerations, this paper proposes a Self-Interactive Summarization Model (SISM) with an encoder-decoder structure based on a self-interactive attention mechanism [10] for single-document automatic summarization. The model consists of a sentence encoder based on self-interactive attention mechanism, a document encoder, and a sentence extractor.

1) **Sentence encoder**: Utilizes word embeddings to generate vector representations of sentences.
2) **Document encoder**: Takes sentence vector representations as input, first extracts the overall document information, then uses a self-interactive attention mechanism to obtain association information between different sentences. The document encoder then reads the document sentences again, combining sentence information with inter-sentence association information to obtain a richer document representation.
3) **Sentence extractor**: Determines which sentences should be selected based on sentence information and encoder output containing document feature information.

We conducted automatic evaluations using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) method on the CNN news dataset, where each document includes an article summary written by the author as the reference summary. Experimental results show that the summaries obtained by SISM contain richer information and can better replace the original documents, improving information acquisition efficiency.

## 2 RELATED WORK

Text summarization has been an active area of research in natural language processing for several decades. Recent years have seen significant advancements, particularly with the application of deep learning techniques. This section provides an overview of relevant work in the field, focusing on extractive summarization methods and the use of attention mechanisms.

### 2.1 Extractive Summarization

Extractive summarization involves selecting important sentences or phrases from the source document to form a summary. Traditional approaches often relied on statistical methods and hand-crafted features [11]. However, the advent of deep learning has led to more sophisticated models.

Liu et al. [2] introduce a novel web text summarization approach, combining deep learning with a unique refined tuning process. Their results outperform existing methods on various datasets, highlighting the importance of their pre-training and refined tuning strategy.

Nallapati et al. [12] proposed SummaRuNNer, one of the early neural network-based models for extractive summarization. It uses a two-layer recurrent neural network (RNN) to encode sentences and documents, treating sentence extraction as a sequence labeling task. This work demonstrated the potential of neural networks in capturing complex document structures for summarization.

Cheng and Lapata [7] introduced a neural summarization model with an attention-based sentence extractor. Their model uses a convolutional neural network (CNN) to encode sentences and a recurrent neural network to represent documents, showcasing the effectiveness of hierarchical document encoding.

Zhou et al. [13] proposed a neural latent variable model for extractive summarization, which jointly learns sentence scoring and selection. This approach addresses the limitation of previous methods that treated sentence scoring and selection as separate steps.

More recently, Liu and Lapata [14] introduced BERTSUM, which leverages the pre-trained BERT model for extractive summarization. They demonstrated that fine-tuning BERT on summarization tasks can lead to state-of-the-art performance, highlighting the potential of transfer learning in summarization.

### 2.2 Attention Mechanisms in Summarization

Attention mechanisms have played a crucial role in improving the performance of summarization models by allowing them to focus on the most relevant parts of the input.

Rush et al. [15] were among the first to apply neural attention to abstractive sentence summarization. Although their work focused on abstractive methods, it paved the way for the use of attention in extractive summarization as well.

Nallapati et al. [16] introduced a neural attention-based model for abstractive summarization, which included a novel coarse-to-fine attention mechanism. This work demonstrated the effectiveness of hierarchical attention in capturing document structure.

Fan et al. [17] proposed a controllable abstractive summarization model with structured attention. Their approach allows for fine-grained control over summary attributes such as length and style, showcasing the flexibility that attention mechanisms can provide.

In the context of extractive summarization, Xiao and Carenini [18] introduced an extractive summarization model with a novel attention mechanism that captures both local and global context. Their approach demonstrates the importance of considering different levels of document context in summarization.

### 2.3 Self-Attention and Transformer-based Models

The introduction of self-attention and Transformer architectures [19] has led to significant advancements in various natural language processing tasks, including summarization.

Zhang et al. [20] proposed HIBERT, a document-level pre-training model based on the Transformer architecture for extractive summarization. Their work demonstrates the effectiveness of hierarchical Transformer encoders in capturing document structure.

Zhong et al. [21] introduced a Transformer-based extractive summarization model that incorporates sentence-level and document-level attention. Their approach achieves strong performance by effectively modeling both local and global document context.

Wang et al. [22] proposed MATINF, a general framework for joint inference of multiple NLP tasks, including extractive summarization. Their model uses a multi-task attention network to capture task-specific and shared information across different NLP tasks.

### 2.4 Hybrid and Novel Approaches

Recent research has also explored hybrid approaches and novel techniques to improve summarization performance.

Narayan et al. [23] introduced a hybrid extractive-abstractive summarization model that first selects salient sentences and then rewrites them abstractively. This approach combines the strengths of both extractive and abstractive methods.

Xu et al. [24] proposed a discourse-aware neural extractive model that incorporates discourse structure into the summarization process. Their work highlights the importance of considering document-level discourse information in extractive summarization.
Jia et al. [25] introduced a neural reinforcement learning approach for extractive summarization. Their model uses a novel reward function that considers both the informativeness and diversity of selected sentences.
Dong et al. [26] proposed a unified pre-training model for both extractive and abstractive summarization. Their approach demonstrates the potential of multi-task learning in improving summarization performance across different paradigms.

### 2.5 Evaluation and Datasets

Advances in summarization models have been accompanied by developments in evaluation metrics and datasets.
Narayan et al. [27] introduced the XSum dataset for extreme summarization, which focuses on highly abstractive single-sentence summaries. This dataset has become popular for evaluating both extractive and abstractive summarization models.
Fabbri et al. [28] created the MNLI dataset for multi-document summarization, addressing the need for large-scale datasets in this domain.
In terms of evaluation, while ROUGE [29] remains widely used, researchers have proposed alternatives to address its limitations. Zhang et al. [30] introduced BERTScore, which uses contextual embeddings to compute similarity between generated and reference summaries. Scialom et al. [31] proposed QUESTEVAL, a reference-free metric for text generation that correlates better with human judgments than traditional metrics.
Our proposed Self-Interactive Summarization Model (SISM) builds upon these advancements, particularly in the use of self-attention mechanisms and hierarchical document encoding. By incorporating a self-interactive attention mechanism in both the sentence and document encoders, SISM aims to capture rich intra-sentence and inter-sentence relationships, addressing limitations of previous models that focused primarily on sentence-level features or used simpler attention mechanisms.

## 3 SISM MODEL

First, we define the extractive summarization task. Given a document D consisting of a sequence of N sentences (s1, s2, ..., sN), select a subsequence of n (n < N) sentences to form the summary of document D. To achieve this goal, each sentence si is scored and labeled with yi ∈ {0, 1}, indicating whether si should be considered as a candidate sentence for the summary. During supervised learning, given document D and model parameters θ, the objective is to maximize the probability of each sentence label (y1, y2, ..., yN):

$$P(y|D;\theta) = \prod_{i=1}^{N} P(y_i|D;\theta) \tag{1}$$

We propose SISM, which consists of three parts: sentence encoder, document encoder, and sentence extractor.

### 3.1 Sentence Encoder

The sentence encoder's role is to obtain sentence vectors containing rich word information based on word embeddings. Assuming a sentence s consists of T words (w1, w2, ..., wT), these words are input into the sentence encoder. To capture the dependency relationships between adjacent words, a bidirectional Long Short-Term Memory (BiLSTM) network is used to process these words:

$$\begin{aligned} h_i^{\rightarrow} &= LSTM^{\rightarrow}(w_i, h_{i-1}^{\rightarrow}) \\ h_i^{\leftarrow} &= LSTM^{\leftarrow}(w_i, h_{i+1}^{\leftarrow}) \end{aligned} \tag{2}$$

To ensure that sentences of different lengths obtain sentence vectors of the same length, we use a self-interactive attention mechanism [10] for linear connection processing of matrix H. Taking H as input, the output vector weights are:

$$a = softmax(v^T tanh(W_1 H^T)) \tag{3}$$

where W1 is a weight matrix that can be obtained through training, and v is a vector parameter. By summing the columns of matrix H using the weight vector a, we can obtain a vector representation of a sentence. However, this vector representation only contains information from a specific position in the sentence, and different positions in a sentence may contain different types of important information that should not be ignored. Therefore, we need to obtain information representing different regions of the sentence to get a vector representation containing the overall semantic information of the sentence. We extend vector v to matrix V, obtaining the output matrix weights A:

$$A = softmax(V^T tanh(W_1 H^T)) \tag{4}$$

Thus, we can obtain matrix M = AH, representing information from different regions of the sentence. Since each column of M contains similar information, we perform max-pooling on M column-wise to obtain the final sentence vector s.

### 3.2 Document Encoder

The document encoder consists of a 2-layer LSTM. The first layer reads the sentences in the document sequentially, captures their features, and obtains an initial document representation. Then, an attention mechanism is applied to obtain a vector representation containing multiple aspects of document information, including inter-sentence interaction information. The second layer reads the sentence sequence again to reduce information loss and combines the sentence vector representations with the output of the attention mechanism layer to obtain the final document representation.
Given a document D = (s1, s2, ..., sN), the hidden state update method for the first layer of the encoder at time t is:

$$h_{1t}^{\rightarrow} = LSTM^{\rightarrow}(s_t, h_{1t-1}^{\rightarrow})$$
$$h_{1t}^{\leftarrow} = LSTM^{\leftarrow}(s_t, h_{1t+1}^{\leftarrow}) \quad (5)$$

At time t, the hidden state h1th_{1t}h1t not only contains historical information prior to sentence st but also information from sentences far from st. To better describe the connections between sentences and obtain more useful information, we design a self-interactive attention mechanism to process the output of the first layer of the encoder.
First, we assign different weights to each hidden state of the first layer and sum them:

$$m_{1t} = \sum_{k=1}^{N} \alpha_{tk} h_{1k} \quad (6)$$

where αtk\alpha_{tk}αtk is the normalized weight for the k-th hidden state at time t:

$$\alpha_{tk} = \frac{exp(e_{tk})}{\sum_{i=1}^{N} exp(e_{ti})} \quad (7)$$

etk is the initial weight value calculated using only H:

$$e_{tk} = v_{tk}^{T} tanh(WH^{T}) \quad (8)$$

vtk and W are trainable model parameters. Thus, each m1t contains the connection between st and other sentences.
At time t, m1t is input into the second layer of the encoder. To reduce information loss as much as possible, the sentence vector is also input and combined with m1t. The update method for the hidden state of the second layer at time t is:

$$h_{2t} = LSTM(s_t || m_{1t}, h_{2t-1}) \quad (9)$$

where || denotes concatenation of st and m1t. This yields the vector representation of the document.
**3.3 Sentence Extractor** The sentence extractor consists of an LSTM network that can detect and compute the salience of each sentence and label it. Given document D and the hidden states of the document encoder (h21, h22, ..., h2N), the extractor makes the following prediction for the label of the i-th sentence:

$$P(y_i|s_i, D) = softmax(MLP(h_{2i}, h_{di})) \quad (10)$$

where MLP(h2i, hdi) represents a multi-layer network calculated as:

$$MLP(h_{2i}, h_{di}) = Wtanh(W_1 h_{2i} + W_2 h_{di}) \quad (11)$$

where W1, W2, and W are trainable neural network parameters; hdi is the hidden state of the sentence extractor calculated as:

$$h_{dt} = LSTM(s_t, h_{dt-1}) \quad (12)$$

hd0 is the last output hidden state h2N of the document encoder. The loss function used during model training is:

$$loss = -\sum_{i=1}^{N} logP(y_i|s_i, D) \quad (13)$$

Finally, the prediction result for whether sentence si should be selected as part of the summary is:

$$\hat{y}_i = argmax_{y_i \in \{0,1\}} P(y_i|s_i, D) \quad (14)$$

## 4 EXPERIMENTAL VALIDATION

We aimed to investigate the following questions:
1) Compared to baseline models, can the attention mechanism in SISM improve the performance of extractive summarization?
2) How does the length of the generated summary (i.e., 75B, 275B, and full length (3 sentences)) affect the results?
3) We compared SISM with two baseline models that generate summaries by selecting significant sentences from the original document:
4) **LEAD**: A standard model that selects the first 3 sentences of the document as the summary [32, 33].
5) **NN-SE**: A neural network model for extractive summarization, including a hierarchical document encoder and an attention-based sentence extractor [7].

### 4.1 Dataset and Experimental Details

We constructed the model training and testing dataset based on CNN news [7]. The CNN news dataset is widely used in automatic question-answering system research. Each document contains the original news text and highlighted text written

by news editors, which can serve as true abstractive summaries and be used as reference summaries. Cheng et al. [7] used a rule-based method to label each sentence in the document with 0 or 1 (1 indicates that the sentence matches the highlighted text; 0 indicates otherwise) with a verified sentence label accuracy of 85%.
The dataset statistics are shown in Figure 1. Since over 95% of sentences in the dataset do not exceed 50 words and over 95% of documents do not exceed 60 sentences, we set the sentence length to 50 and the document length to 60. For the document encoder and sentence extractor, we use LSTM units with a size of 650. The regularization dropout rate used in the LSTM input entering the hidden layer and the sentence scoring process is 0.5. During training, we perform batch training with 20 documents per batch using the Adam optimizer with an initial learning rate of 0.001.

| | Number of neurons in the hidden layer | | | | | |
|---|---|---|---|---|---|---|
| | 5 | | 10 | | 15 | |
| BBO | Average | **0.880** | Average | **0.879** | Average | **0.882** |
| | Stdv | **0.011** | Stdv | **0.012** | Stdv | **0.012** |
| | Best | **0.902** | Best | **0.892** | Best | **0.906** |
| GA | Average | 0.858 | Average | 0.833 | Average | 0.833 |
| | Stdv | 0.009 | Stdv | 0.011 | Stdv | 0.011 |
| | Best | 0.873 | Best | 0.867 | Best | 0.902 |
| PSO | Average | 0.785 | Average | 0.785 | Average | 0.768 |
| | Stdv | 0.019 | Stdv | 0.022 | Stdv | 0.009 |
| | Best | 0.818 | Best | 0.812 | Best | 0.906 |
| DE | Average | 0.681 | Average | 0.693 | Average | 0.708 |
| | Stdv | 0.043 | Stdv | 0.037 | Stdv | 0.023 |
| | Best | 0.740 | Best | 0.741 | Best | 0.751 |
| ACO | Average | 0.643 | Average | 0.623 | Average | 0.618 |
| | Stdv | 0.043 | Stdv | 0.030 | Stdv | 0.065 |
| | Best | 0.743 | Best | 0.671 | Best | 0.709 |
| BP | Average | 0.658 | Average | 0.673 | Average | 0.720 |
| | Stdv | 0.054 | Stdv | 0.060 | Stdv | 0.045 |
| | Best | 0.747 | Best | 0.766 | Best | 0.787 |

**Figure 1** Dataset Statistics

### 4.2 Evaluation Method

We used ROUGE [29] to evaluate the quality of the summaries generated by the models on the entire CNN test set. ROUGE is a recall-based measurement method. ROUGE-N (N=1, 2, 3, 4) can measure the recall rate of N-grams between candidate summaries and reference summaries, which can be used to measure the amount of information contained in the summaries. ROUGE-L can detect the longest common subsequence, reflecting the readability and fluency of the summaries. We use ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-3 (R-3), and ROUGE-4 (R-4) to reflect the information content of the summaries, and ROUGE-L (R-L) to reflect the fluency of the summaries. We provide summaries of full length and fixed lengths (75B and 275B). To ensure fair comparison, we select the 3 highest-scoring sentences as the full-length summary.

## 5 RESULTS AND ANALYSIS

We validated the models by using ROUGE scores for summaries of 75B, 275B, and full length (3 sentences) generated by SISM and the two baseline models. The ROUGE score comparisons for summaries of different lengths for each model are shown as figure 2.

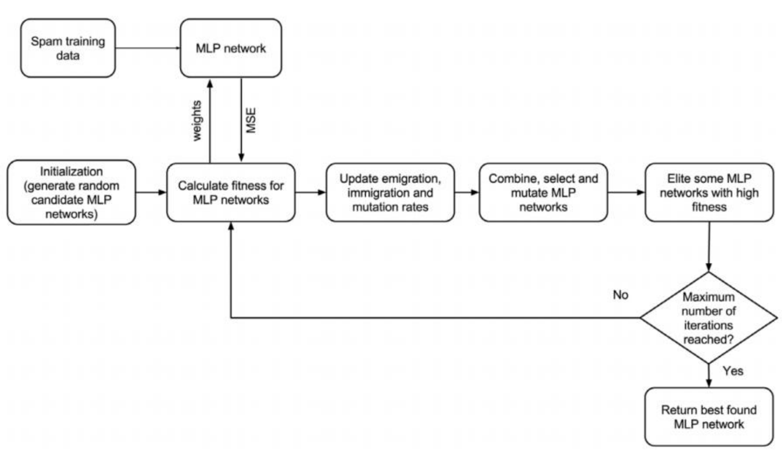


**Figure 2** The ROUGE score comparisons for summaries of different lengths for each model

### 5.1 Summarization Model Performance

To address the first research question, we compared the ROUGE scores of full-length summaries for each model. As seen in Figure 3, among the two baseline models, the NN-SE model's ROUGE-N scores are all higher than those of the LEAD model, but its R-L score is slightly lower than that of the LEAD model. This indicates that the NN-SE model generates summaries with richer information, but the LEAD model produces more fluent and coherent summaries.

SISM outperforms the baseline models in all evaluation metrics with significant improvements. Compared to the NN-SE model, SISM's R-1, R-2, R-3, R-4, and R-L scores improved by 7.4%, 24.3%, 13.4%, 7.1%, and 7.6% respectively, with 4 of these improvements being statistically significant. This demonstrates that using the self-interactive attention mechanism combined with bidirectional LSTM in the sentence encoder and document encoder of SISM helps capture the document's main ideas and select sentences with significant meaning as summaries.

### 5.2 Performance of Summaries with Different Lengths

To address the second research question, we also compared the ROUGE scores of 75B, 275B, and full-length (3 sentences) summaries generated by the three models. As shown in figure 3, SISM achieves the best results for summaries of all lengths, while the NN-SE model performs slightly better than the LEAD model among the two baseline models.

When generating 75B-length summaries, SISM shows small improvements in all ROUGE scores compared to the two baseline models. For 275B-length summaries, SISM's R-1, R-2, R-3, and R-4 scores improved by 9.3%, 12.5%, 26.4%, and 24.2% respectively compared to the NN-SE model, while its R-L score improved by 4.6% compared to the LEAD model. The improvements in R-1, R-3, and R-L for SISM are statistically significant. Overall, SISM performs better when generating longer summaries.

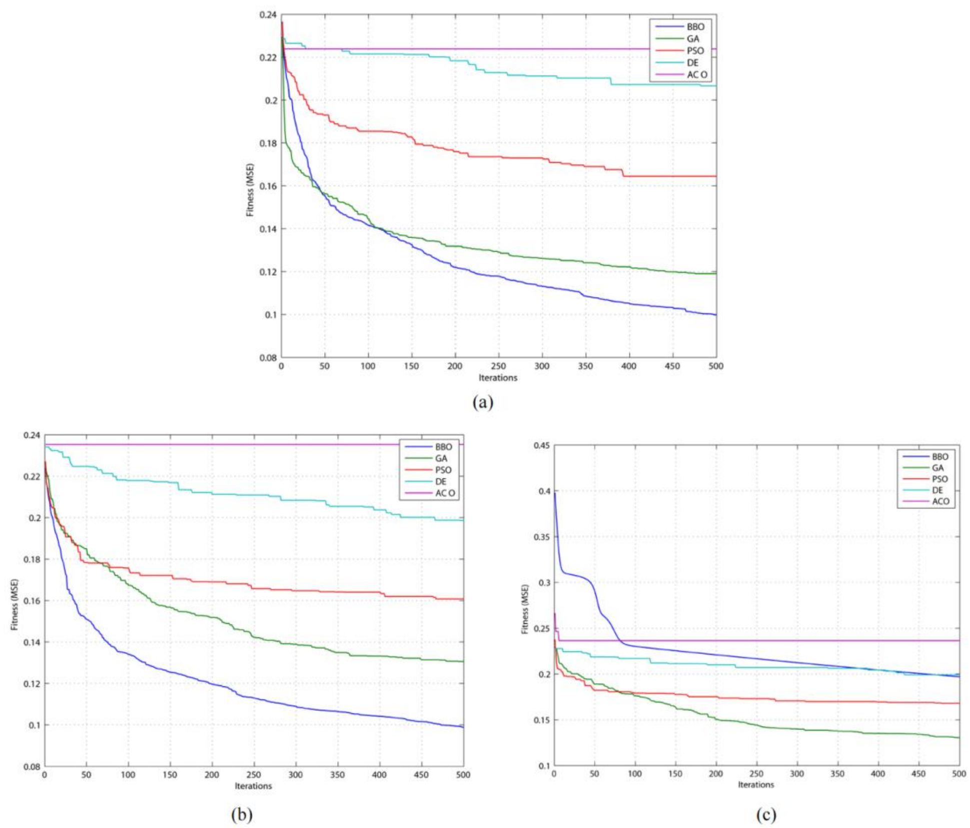


**Figure 3** The ROUGE scores of 75B, 275B, and full-length (3 sentences) summaries generated by the three models

## 6 CONCLUSION AND FUTURE WORK

In this paper, we proposed a text summarization model for automatically generating extractive summaries. The model utilizes an encoder-decoder structure and employs a self-interactive attention mechanism to effectively mine textual information and structural features. It produces summaries with higher ROUGE-N and ROUGE-L scores, indicating that the

model generates summaries with more information content and better fluency. Experimental results show that SISM significantly outperforms the LEAD and NN-SE baseline models, especially when generating longer summaries.
For future work, we plan to test SISM on different datasets to verify its effectiveness across various text domains. We will also leverage other features such as document topics, titles, and inter-paragraph relationships to capture richer and more significant document information, further improving automatic summarization performance.

## COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

## REFERENCES

[1] Nenkova, A., McKeown, K. A survey of text summarization techniques. In Mining text data. Springer, Boston, MA. 2013: 43-76.
[2] Liu, M., Ma, Z., Li, J., Wu, YC., Wang, X. Deep-Learning-Based Pre-training and Refined Tuning for Web Summarization Software. IEEE Access, 2024, 12: 92120-92129.
[3] Sutskever, I., Vinyals, O., Le, QV. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, 2024: 3104-3112.
[4] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 2014.
[5] Bahdanau, D., Cho, K., Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. 2014.
[6] Luong, M. T., Pham, H., Manning, CD. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025. 2015.
[7] Cheng, J., Lapata, M. Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252. 2016.
[8] Nallapati, R., Zhai, F., Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence. 2017.
[9] See, A., Liu, P. J., Manning, CD. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368. 2017.
[10] Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., Bengio, Y. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130. 2017.
[11] Erkan, G., Radev, DR. LexRank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, 2004, 22: 457-479.
[12] Nallapati, R., Zhai, F., Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence. 2017.
[13] Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T. Neural document summarization by jointly learning to score and select sentences. arXiv preprint arXiv:1807.02305. 2018.
[14] Liu, Y., Lapata, M. Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345. 2019.
[15] Rush, A. M., Chopra, S., Weston, J. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685. 2015.
[16] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023. 2016.
[17] Fan, A., Grangier, D., Auli, M. Controllable abstractive summarization. arXiv preprint arXiv:1711.05217. 2018.
[18] Xiao, W., Carenini, G. Extractive summarization of long documents by combining global and local context. arXiv preprint arXiv:1909.08089. 2019.
[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. Attention is all you need. In Advances in neural information processing systems, 2017: 5998-6008.
[20] Zhang, X., Wei, F., Zhou, M. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. arXiv preprint arXiv:1905.06566. 2019.
[21] Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., Huang, X. Extractive summarization as text matching. arXiv preprint arXiv:2004.08795. 2009.
[22] Wang, H., Liu, Q., Gao, Z., Nie, Y., Liu, Z., Zheng, H. Neural multi-task learning for aspect-based sentiment analysis. In 2019 International Joint Conference on Neural Networks (IJCNN). IEEE. 2019: 1-8.
[23] Narayan, S., Cohen, S. B., Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745. 2018.
[24] Xu, J., Gan, Z., Cheng, Y., Liu, J. Discourse-aware neural extractive text summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 5021-5031.

[25] Jia, Y., Ye, Y., Feng, Y., Lai, Y., Yan, R., Zhao, D. Modeling discourse cohesion for discourse parsing via memory network. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics2018, 2: 429-434.
[26] Dong, Y., Shen, Y., Crawford, E., van Hoof, H., Cheung, JCK. BanditSum: Extractive summarization as a contextual bandit. arXiv preprint arXiv:1809.09672. 2018.
[27] Narayan, S., Cohen, S. B., Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745. 2018.
[28] Fabbri, A. R., Li, I., She, T., Li, S., Radev, D. R. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. arXiv preprint arXiv:1906.01749. 2019.
[29] Wang, X., Wu, Y. C., Ji, X., Fu, H. Algorithmic discrimination: examining its types and regulatory measures with emphasis on US legal practices. Frontiers in Artificial Intelligence, 2024, 7: 1320277.
[30] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., Artzi, Y. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675. 2019.
[31] Scialom, T., Dray, P. A., Lamprier, S., Piwowarski, B., Staiano, J. QuestEval: Summarization asks for fact-based evaluation. arXiv preprint arXiv:2103.12693. 2021.
[32] Wasson, M. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, 1998: 2.
[33] Brandow, R., Mitze, K., Rau, LF. Automatic condensation of electronic publications by sentence selection. Information Processing & Management, 1995, 31(5): 675-685.
[34] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.
[35] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. Language models are unsupervised multitask learners. OpenAI blog, 2019, 1(8): 9.
[36] Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N. Generating wikipedia by summarizing long sequences. arXiv preprint arXiv:1801.10198. 2018.
[37] Chen, X., Liu, M., Niu, Y., Wang, X., Wu, YC. Deep-Learning-Based Lithium Battery Defect Detection via Cross-Domain Generalization. IEEE Access, 2024, 12: 78505-78514
[38] Wang, X., Wu, Y. C., Ma, Z. Blockchain in the courtroom: exploring its evidentiary significance and procedural implications in US judicial processes. Frontiers in Blockchain, 2024, 7, 1306058.
[39] Ma, Z., Chen, X., Sun, T., Wang, X., Wu, Y. C., Zhou, M. Blockchain-Based Zero-Trust Supply Chain Security Integrated with Deep Reinforcement Learning for Inventory Optimization. Future Internet, 2024, 16(5): 163.
[40] Wang, X., Wu, Y. C., Zhou, M., Fu, H. Beyond Surveillance: Privacy, Ethics, and Regulations in Face Recognition Technology. Frontiers in Big Data, 2024, 7, 1337465.
[41] Kedzie, C., McKeown, K., Daume III, H. Content selection in deep learning models of summarization. arXiv preprint arXiv:1810.12343. 2018.