Considering Nested Tree Structure in Sentence Extractive Summarization with Pre-trained Transformer

Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito and Manabu Okumura
Tokyo Institute of Technology
{kwonjingun,kobayasi,kamigaito}@lr.pi.titech.ac.jp
oku@pi.titech.ac.jp

Abstract

Sentence extractive summarization shortens a document by selecting sentences for a summary while preserving its important contents. However, constructing a coherent and informative summary is difficult using a pre-trained BERT-based encoder since it is not explicitly trained for representing the information of sentences in a document. We propose a nested tree-based extractive summarization model on RoBERTa (NeRoBERTa), where nested tree structures consist of syntactic and discourse trees in a given document. Experimental results on the CNN/DailyMail dataset showed that NeRoBERTa outperforms baseline models in ROUGE. Human evaluation results also showed that NeRoBERTa achieves significantly better scores than the baselines in terms of coherence and yields comparable scores to the state-of-the-art models.

1 Introduction

Document summarization is a task of creating a concise summary from a given document while keeping the original content. In general, sentence extraction methods, which select sentences in a document to create its summary, have the advantages of truthfulness compared with abstractive methods (Cao et al., 2018) and of fluency compared with word extraction methods (Xu et al., 2020).

Neural networks have achieved great success in sentence extraction-based document summarization (Cheng and Lapata, 2016; Zhou et al., 2018). Recently, Liu and Lapata (2019) proposed BERTSUM, which utilizes BERT (Devlin et al., 2019) for sentence representations to create a summary. Although the use of BERT resulted in significant performance improvement, this method decides the selection for each sentence independently. Xu et al. (2020) proposed DISCOBERT by considering inter-sentence information through discourse graphs to construct a coherent summary. Although they achieved remarkable scores in ROUGE, it was still difficult to construct a coherent summary compared to BERTSUM in human evaluation. Zhong et al. (2020) attempted to change the paradigm by formulating summary-level extraction with a RoBERTa encoder and achieved the state-of-the-art results on the CNN/DailyMail dataset.

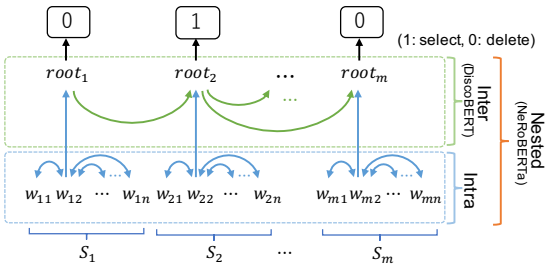

Figure 1: Different from the previous work, DISCOBERT (Xu et al., 2020), NeRoBERTa selects sentences by considering both intra- and inter-sentence relationships as a nested tree structure.

In spite of the successful results of the above BERT-related methods, their sentence representations have room for improvement. As Liu et al. (2019) reported, "[CLS]", a pre-defined token for indicating sentence representations on BERT, is insufficient to express sentence information. Even in RoBERTa, it is also a problem due to the lack of next sentence prediction in its pretraining step. Therefore, for further improving summarization performance, we need to consider how to represent sentences in a BERT-related model and how to capture relationships between such sentence representations. It is a key to create a coherent and informative summary with sentence extraction methods.

To tackle this problem, we propose a nested tree-based extractive summarization model on RoBERTa (NeRoBERTa). NeRoBERTa can extract coherent sentences for a summary of a given document by utilizing nested tree structures[1] of two

[1] Kikuchi et al. (2014) considered the nested tree structure in the traditional non-neural tree-trimming method. Their method extracted words by tracking their parent words and

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4039–4044
November 7–11, 2021.

different trees, syntactic and discourse dependency trees (Zhao and Huang, 2017). Figure 1 shows the proposed NeRoBERTa to select sentences from a given document. Different from the previous works that focused on inter-sentence information using discourse graphs (Ishigaki et al., 2019; Xu et al., 2020), NeRoBERTa considers both intra- and inter-sentence information (syntactic and discourse graphs) together as a nested tree. The nested tree is encoded as a vector space representation through a graph attention network (Veličković et al., 2018) on a BERT-based encoder. In this tree, we can explicitly represent sentence information at "root" words for each syntactic dependency tree without relying only on "[CLS]" tokens.

This representation is useful to extract informative and coherent sentences in that it can capture keywords in a sentence for considering textual coherence to other sentences. Furthermore, based on the representation, we can also capture interactions between sentences through discourse dependency trees, succeeding in extracting coherent sentences. It is also possible to consider even long-distance relationships as higher-order dependency relationships in this structure, such as relationships between children and their ancestors. Thus, NeRoBERTa considers textual coherence through both syntactic and discourse trees to capture long-distance interactions between sentences.

Experimental results on the CNN/DailyMail dataset showed that our NeRoBERTa outperforms RoBERTa-based strong baselines in ROUGE. Unlike the previous work (Xu et al., 2020), NeRoBERTa successfully constructs a coherent summary and is comparable to the state-of-the-art methods in human evaluation.

2 Nested Tree Structure

In this section, we describe how we construct two different types of graphs for a nested tree structure: a discourse graph and a syntactic graph.

We obtain discourse dependency relationships between sentences in a document through an RST parser. A given document can be parsed into a tree format with the RST parser, where each leaf node is an EDU, a text span in the document. Each text span has two types, nucleus and satellite. While the nucleus spans contain semantically salient information, the satellite spans support and modify the nucleus ones.

We use the recent state-of-the-art RST parser[2] (Kobayashi et al., 2020) to build an RST discourse tree (RST-DT) for all documents and convert it to an Inter-Sentential RST-DT (ISRST-DT). The ISRST-DT is first converted into a dependency-based discourse tree (ISDEP-DT) using the method described in (Hirao et al., 2013). Then, parent-child dependency relationships for each sentence can be formed. We construct a directed graph for the discourse dependencies (Ishigaki et al., 2019).

A dependency parser is used to build up the syntactic dependency relationships between words (Manning et al., 2014). We construct an undirected graph for the syntactic dependencies by following the previous settings (Marcheggiani and Titov, 2017).

3 Our Model

Ishigaki et al. (2019) consider dependency information through hierarchical attention modules (Kamigaito et al., 2018) trained in supervised attention for dependency heads (Kamigaito et al., 2017). Unlike the previous work, our model uses constructed graph information through graph encoder layers that directly focus on the relationships between nodes defined by edges in the graph. We explain the details of our model in this section.

Let w_i be the i-th token in a document $D = \{w_1, w_2, ..., w_n\}$. Our model predicts $p(1|D, k)$, the probability of the k-th sentence in D being kept in a summary through the following modules.

3.1 Pre-trained Document Encoder

We append "[CLS]" and "[SEP]" tokens between sentences to encode a whole document (Liu and Lapata, 2019). Then, BERT is used to build up a representation h_i for each token w_i as follows:

$$\{h_1, h_2, ..., h_n\} = \mathrm{BERT}(\{w_1, w_2, ..., w_n\}).$$

Instead of BERT, we consider RoBERTa as well. However, RoBERTa cannot be directly used in place of BERT for sentence-level extraction because RoBERTa does not consider the two types of tokens for the segment boundaries. To address this issue, we use randomly initialized segment embeddings, $W_{type} \in \mathrm{R}^{2,768}$, instead of the original embeddings for keeping the same condition as BERT. The number comes from the pre-trained

sentences to construct a summary for a given document.

[2]We used the RST-parser using the RoBERTa embeddings

segment embedding weights of the original BERT, which indicate the next sentence prediction step. Then, the encoded hidden states, $\{h_1, h_2, ..., h_n\}$, are fed into our graph encoders.

3.2 Graph Encoders

Graph Notation: Let V_d and V_s be nodes for sentences and words, and E_s and E_d be edges between the nodes in V_s and V_d, respectively. We denote constructed discourse and syntactic graphs as $G_d = (V_d, E_d)$ and $G_s = (V_s, E_s)$, respectively. We append undirected edges between "[CLS]" and "root" tokens in each sentence to E_s because the parent of a "root" token would be a sentence representation.

GAT Networks: We use Graph Attention Networks (GAT) (Veličković et al., 2018) to encode each graph G on hidden states of BERT as follows:

$$f_i = \mathrm{F}^2(h_i), h_i \in \mathrm{R}^{d,n}, \quad (1)$$

$$n_i = \mathrm{N}(\mathrm{drop}(f_i) + h_i), \quad (2)$$

$$\alpha_{i,j} = \mathrm{Softmax}_j(\mathrm{L}(\mathrm{F}^1[W_n n_{nl} \parallel W_n n_{nl}])), \quad (3)$$

$$h'_i = \|_{k=1}^{K} \mathrm{T}(\sum_{j \in N_i} \alpha_{i,j}^k W_a^k h_j), h'_i \in \mathrm{R}^{K \times d,n}, \quad (4)$$

$$h''_i = \mathrm{ReLU}(\mathrm{M}(h'_i)), h''_i \in \mathrm{R}^{d,n}, \quad (5)$$

$$h_i^G = \mathrm{N}(\mathrm{drop}(h''_i) + n_i), \quad (6)$$

where F^{i} indicates i-th times stacked feed-forward networks. N is layer normalization. W_n and W_a are learnable weights. L and T denote a non-linearity activation function, LeakyReLU, and a hyperbolic tangent, respectively. $\alpha_{i,j}$ indicates normalized attention coefficients through a softmax function. $\parallel$ indicates concatenation, and n_i represents connected nodes to node i in graph G. ReLU is an activation function. M is a learnable weight. After h_i is fed into the graph encoder, we obtain h_i^G, which contains either syntactic or discourse graph information based on all tokens.

The syntactic and discourse graphs are independently encoded. Then, they are concatenated as h_k^{root}=ReLU$(W(h_{r(k)}^{G_s} \parallel h_{r(k)}^{G_d}))$, where $r(k)$ indicates the position of a root in the k-th sentence. For the final representations to predict labels, we use h_k^{root} to represent the k-th sentence.

3.3 Objective Function & Inference

We define $p(1|D,k) = \sigma(WM(h_k^{root}) + b)$, where M is a two-stacked multi-head attention, σ is a sigmoid function, and W and b are weight parameters (Liu and Lapata, 2019). Let $y_i \in \{1, 0\}$ be an oracle label and $Y = \{y_1, y_2, ..., y_n\}$ be its set for a document. We use $-\sum_{y_k \in Y} log(y_k|x,k)$ as our objective function. In the inference step, we score the k-th sentence with $p(1|D,k)$ and sort the sentences in descending order. Then, we keep the top m sentences as a summary, where m is the number of sentences to be extracted.

4 Experiments

4.1 Experimental Settings

Dataset: We used the non-anonymized CNN/DailyMail dataset (Hermann et al., 2015). Based on the standard split, we divided the dataset into 287,226, 13,368 and 11,490 articles for training, validation, and test datasets, respectively.

Parameter Settings: We used PyTorch with the Torch Geometric (Fey and Lenssen, 2019) to build up entire architectures with graph encoders. The "bert-based-uncased" and "roberta-based" models in transformers[3] were used to encode maximum 768 tokens of each tokenized document. The best model was selected based on the lowest "loss" score on the validation dataset. A greedy search was used to construct the oracle summary by maximizing the sum of ROUGE-1-F and ROUGE-2-F against the gold summary.

For the syntactic graph encoder, we stacked GAT Networks. To track n-order dependency information, we simply added n-order nodes and edges to G_d and G_s. The number of attention heads was set to 6 in each graph encoder. To represent each word vector, we used a first sub-word vector. We employed a traditional method of selecting top 3 sentences to construct a summary (Liu and Lapata, 2019). *Trigram blocking* was used to reduce redundancy and to improve informativeness for all models (Paulus et al., 2018).

Compared Methods: We compared our proposed methods with some baselines. The proposed methods are as follows:

NeRoBERTa considers our nested tree structure for both syntactic and discourse information.

SynRoBERTa and **DiRoBERTa** independently consider only either syntactic or discourse tree structure, respectively.

The baselines, which include state-of-the-art models, are as follows:

BERTSUM introduces a method for learning a sentence boundary in a BERT-based model for the document summarization task (Liu and Lapata, 2019).

[3]https://github.com/huggingface/transformers

DISCOBERT constructs a summary based on EDU-level extraction, incorporating discourse and coreference information (Xu et al., 2020).

MatchSum attempts to shift the paradigm from sentence-level to summary-level extraction during the extractive document summarization task (Zhong et al., 2020).

RoBERTa encodes input documents using a "roberta-based" model.

4.2 Automatic Evaluation

We utilized ROUGE-metrics for the evaluation. The experimental results on the CNN/DailyMail dataset are shown in Table 1. The first block contains Lead-3 and Oracle scores. The second block includes BERT-based previous studies including state-of-the-art models. The last block includes scores for our models and for re-implemented BERTSUM.

Our strong baseline RoBERTa outperformed BERTSUM. The gain might be from using a bigger dataset with the dynamic masking pattern applied in the pre-trained RoBERTa. SynRoBERTa and DiRoBERTa show that considering syntactic or discourse information was beneficial. NeRoBERTa $(n_s = \{1, 2\}, n_d = \{1\})$ (in bold), that considers syntactic and discourse information simultaneously, further improved the performance. It outperformed RoBERTa with a clear margin, specifically, 0.31 points in the R-1-F score.

As can be seen in Figure 2, RoBERTa can improve the prediction loss compared with BERTSUM. SynRoBERTa $(n_s = \{1, 2\})$, which explicitly incorporates keywords information through syntactic information, can further improve the performance of RoBERTa. This shows that considering keywords information through syntactic structures is beneficial to construct the sentence representations for considering textual coherence to other sentences.

4.3 Human Evaluation and Analysis

Human evaluation was conducted for randomly sampled 100 documents from the test dataset. "Amazon Mturk" was used for the experiments, and human evaluators graded scores from 1 to 5 (5 is the best) in terms of four evaluation criteria.[5] Because summaries from DISCOBERT were worse than ones from BERTSUM in their human evaluation (Xu et al., 2020), we evaluated only summaries from RoBERTa, NeRoBERTa $(n_s = \{1, 2\}, n_d = \{1\})$, and MatchSum. Table 2 shows the results. Coh, Infor, Read, and Redun indicate coherence, informativeness, readability, and redundancy, respectively. As we expected, the proposed NeRoBERTa, which considers a nested tree structure, could capture coherence better than our strong baseline, RoBERTa. In addition, NeRoBERTa was comparable to the current state-of-the-art model, MatchSum. The informativeness score for MatchSum was lower than RoBERTa and NeRoBERTa.

Table 3 shows example extracted sentences from a document and their discourse graph. In this example, the discourse information alone was not enough in that S3 and S10 have the same discourse information, while S3 is more similar to the third sentence in the gold summary. RoBERTa and DiRoBERTa constructed the same summary in-

Model	R-1-F	R-2-F	R-L-F
Lead3	40.12	17.52	36.44
Oracle	55.05	32.72	51.38
BERTSUM	43.25	20.24	39.63
DISCOBERT	43.77	20.85	40.67
MatchSum	44.41	20.86	40.55
BERTSUM	43.28	20.11	39.68
RoBERTa	43.55	20.40	39.94
SynRoBERTa $(n_s = \{1\})$	43.73	20.58	40.10
SynRoBERTa $(n_s = \{1, 2\})$	43.63	20.51	40.02
DiRoBERTa $(n_d = \{1\})$	43.64	20.45	40.02
NeRoBERTa $(n_s = \{1\}, n_d = \{1\})$	43.74	20.53	40.13
NeRoBERTa $(n_s = \{1, 2\}, n_d = \{1\})$	**43.86**†	**20.64**†	**40.20**†

Table 1: Experimental results on the CNN/DailyMail dataset. n_s and n_d indicate the order of dependency relationships considered for syntactic and discourse graphs, respectively. † indicates the improvement is significant with a 0.95 confidence interval estimated with the ROUGE script compared to RoBERTa.

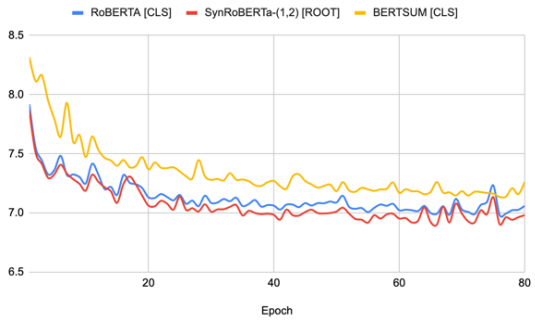

Figure 2: Validation losses for BERTSUM, RoBERTa, and SynRoBERTa $(n_s = \{1, 2\})$. "[CLS]" and "[ROOT]" indicate the tokens of sentence representations for predicting labels.

[4]The paired-bootstrap-resampling (Koehn, 2004) was used ($p < 0.05$).

[5]40 human evaluators who obtained both US high school and US bachelor degrees participated in the experiments.

Model	Coh	Infor	Read	Red
MatchSum	4.06	4.11	4.09	4.17
RoBERTa	4.02	4.14	4.09	4.12
NeRoBERTa	4.08†	4.14	4.10	4.16

Table 2: Human evaluation results. † indicates that the improvement with NeRoBERTa from RoBERTa was statistically significant.[4]

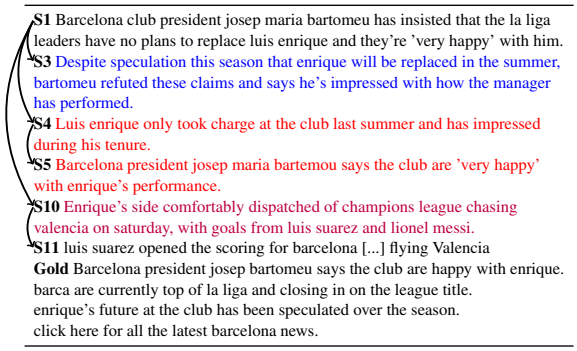

Table 3: Example extracted sentences from RoBERTa, DiRoBERTa ($n_d = \{1\}$), NeRoBERTa ($n_s = \{1, 2\}, n_d = \{1\}$), and MatchSum models. Arrows indicate the discourse graphs. The sentences in red are selected by all models. The sentence in blue is selected by NeRoBERTa and the sentence in purple is selected by RoBERTa and DiRoBERTa. S1 is the first sentence of the document. Gold denotes the gold summary.

cluding S10. On the other hand, NeRoBERTa could extract S3, which is coherent to S4 and S5, sharing important keywords "enrique" and "bartomeu". This is because our GAT network for syntactic information can capture keywords in the sentence to consider textual coherence to other sentences. Although NeRoBERTa constructed a summary with three sentences, MatchSum extracted only two sentences of S4 and S5. In this case, MatchSum might be less informative than NeRoBERTa.

5 Conclusion

In this paper, we proposed NeRoBERTa, which incorporates syntactic and discourse information as a nested tree structure to create an informative and coherent summary. The experimental results on the CNN/DailyMail dataset showed that our method improves the performance over the baseline methods both in the automatic and human evaluations.

Acknowledgements

We would like to gratefully acknowledge the anonymous reviewers for their helpful comments and feedbacks. This work was supported by Google AI Focused Research Award.

References

Ziqiang Cao, Furu Wei, W. Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. *ArXiv*, abs/1711.04434.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701. MIT Press.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1515–1520. Association for Computational Linguistics.

Tatsuya Ishigaki, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2019. Discourse-aware hierarchical attention network for extractive single-document summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 497–506. INCOMA Ltd.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2018. Higher-order syntactic attention network for longer sentence compression. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 7–12, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yuta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2014. Single document summarization based on nested tree structure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 315–320. Association for Computational Linguistics.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Top-down rst parsing utilizing granularity levels in documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8099–8106.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031. Association for Computational Linguistics.

Kai Zhao and Liang Huang. 2017. Joint syntacto-discourse parsing and the syntacto-discourse treebank. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2117–2123. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663. Association for Computational Linguistics.