CSSS: A Novel Candidate Summary Selection Strategy for Summary-level Extractive Summarization

Shuai Gong and **Zhenfang Zhu*** and **Wenqing Wu** and **Zhen Zhao** and **Dianyuan Zhang**
ShanDong JiaoTong University

Abstract

Summary-level extractive summarization selects a summary with the highest semantic similarity to the document through a matching model, resulting in insufficient use of information between different candidate summaries. This paper presents a novel candidate summary selection strategy (CSSS), regarding candidate summaries as mathematical sets, and selecting the candidate summary has the highest semantic similarity to corresponding mutually exclusive sets. The strategy reduces the dependence on the matching model by exploiting the set relationship, could be effectively applied to both unsupervised and supervised extractive summarization. In order to fit this strategy better, we construct a contrastive learning framework to learn effective vector representation for each candidate summary. Experimental results show that we achieve state-of-the-art performance in both the unsupervised and supervised extractive summarization on CNN/DailyMail dataset. Experiments on Xsum and Reddit datasets also show the effectiveness of CSSS.

1 Introduction

[1] Extractive summarization task aims to distill the important information into a concise summary by selecting the text fragments from the original document. The task can generally be divided into two categories: sentence-level and summary-level.

Sentence-level methods (Nallapati et al., 2017; Liu and Lapata, 2019; Wang et al., 2020; Jia et al., 2021a) extract several sentences one by one from the original document to form a summary. These methods tend to select highly generalized sentences while ignoring the coupling of multiple sentences, because they make binary decisions independently for each sentence rather than considering the semantic of the entire summary.

In order to solve these problems, Zhong et al. (2020) proposed a summary-level method. It first extracts the salient sentences from the original document to form a pruned document, then enumerates all possible combinations of sentences extracted from it as the candidate summaries, finally uses a text matching model to select the final summary with the highest semantic similarity to the original document. The method utilizes the sequential information between candidate summaries measured by the ROUGE (Lin and Hovy, 2003) scores when training the matching model. However, such sequential information is not available at the final summary selection stage, which limits matching model performance, and makes the method unable to generalize to unsupervised summarization.

In this paper, we propose a novel candidate summary selection strategy (CSSS) to solve the problems in summary-level extractive summarization. The strategy first regards each candidate summary as a mathematical set,then selects the best candidate summary by computing the semantic similarity of all its corresponding mutually exclusive sets.

CSSS reduces the dependence on the matching model, making it effective for both supervised and unsupervised extractive summarization. In supervised extractive summarization, we construct a simple contrastive learning framework to obtain better vector representations for each candidate summary. Contrastive learning has obtained impressive results on many natural language processing tasks, such as sentiment analysis (Ke et al., 2021; Li et al., 2021) question answering (Ye et al., 2021) and automatic text summarization (Zhong et al., 2020; Liu and Liu, 2021; Cao and Wang, 2021). Our framework takes candidate summary pairs as the positive and negative examples and uses contrastive objective to train matching models. While in unsupervised summarization, we explore the effect of different encoders on CSSS. We find that even with the base version of BERT, we obtain a state-of-the-art extractive result on CNN/Daily Mail.

The contributions of our work are as follows:

[1] * is corresponding author.

(1) We proposed a novel candidate summary selection strategy for summary-level extractive summarization. The strategy exploits the relationship between different candidate summaries, reduces the dependence on the matching model, and could be effectively applied to both unsupervised and supervised extractive summarization.

(2) We construct a simple framework for contrastive learning of extractive summarization, which can be used to produce superior candidate summary embeddings.

(3) Our proposed methods have achieved state-of-the-art extractive performance compared with strong baselines on three benchmark datasets.

2 Methodology

2.1 Problem Definition

We regard summary-level extractive summarization task as a text matching problem (Liu and Liu, 2021; Cao and Wang, 2021). Given a single document D consisting n sentences, $D = \{s_1, s_2, \ldots, s_n\}$, we first use a sentence-level extractor to get a pruned document containing m sentences. Then, we generate all k-sentence combinations of the pruned document, and rearrange the sentence order according to the original position in the source document to form candidate summaries. Following these steps, we obtain C_m^k candidate summaries in total, the best of which is selected by CSSS.

2.2 Candidate Summary Selection Strategy

The candidate summary selection strategy (CSSS) utilizes the set relationship between candidate summaries to select the best summary. We show the selection process in and principal idea of this strategy in Figure 1.

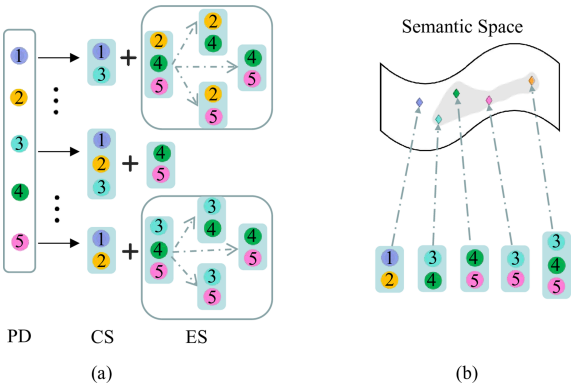

Figure 1: The selection process and principal idea of CSSS. We refer to PD as a pruned document, CS as a candidate summary and ES as the corresponding mutually exclusive sets.

.

The selection process of CSSS is shown is Figure 1(a). CSSS divides the pruned document into two parts, candidate summary (CS) and its corresponding mutually exclusive sets (ES). We score each CS by computing the semantic similarity between the CS and each summary in the ES. The highest semantic similarity is final score of the CS.

The principal idea of CSSS is that if CS is the best summary, it should be semantically closest to the ES, which can be seen in Figure 1(b). Zhong et al. (2020) implement this idea by training a matching model, which is then used to select candidate summary with the highest semantic similarity to the original document.This method is effective, but does not utilize the information between different candidate summaries in the selection stage. We revisit this idea from the perspective of pruned the document. Since the pruned documents are generated by a sentence-level extractor, each selected sentence has a high degree of generalization about the original document. Therefore, for a pruned document, the best summary extract from it, should also be semantically closest to the remaining content.

CSSS can be effectively applied to both unsupervised and supervised extractive summarization. In the unsupervised extractive summarization, we simply use LEAD-k model (which selects the first k sentences in the document.) to prune the original document. After we obtain candidate summaries, we use the original BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021) to obtain their vector representations. While in the supervised extractive summarization, we use BERTSUM (Liu and Lapata, 2019) without trigram blocking to extract

sentences from the original document. Then, we construct a simple contrastive learning framework to obtain better vector representations for each candidate summary.

2.3 Contrastive Learning Framework for Summarization

We use a contrastive learning-based training objective to fine-tune language models BERT and RoBERTa, where the training examples are candidate summary pairs. Then the candidate summaries are encoded using the fine-tuned language model. Formally, given a positive pair (C_i, C_j) and negative sample sets N for a document D, the training objective is:

$$l_d = -log\frac{exp(sim(h_i, h_j)/\tau)}{\sum_{C_k \in N} exp(sim(h_i, h_k)/\tau)}, \quad (1)$$

where h_i, h_j and h_k denote the vector representations for candidate summary C_i, C_j and function C_k, $sim(h_i, h_j)$ calculates the cosine similarity between h_i and h_j. τ is a temperature, which is set to different values in different datasets.

The contrastive learning acquires effective representation by bringing semantically near positive samples together while pushing negative samples apart (Gao et al., 2021). A crucial problem in contrastive learning is how to construct positive and negative samples. We design the training samples according to the quality of candidate summaries. Intuitively, candidate summaries with large differences in quality should have the farthest distances in the semantic space. We describe the process of constructing training samples as follows.

Training Sample Construction Given a document D with its gold summary C_* and n candidate summaries $\left\{C_1', C_2', ..., C_n'\right\}$ obtained from the pruned document, we score each candidate summary C_i' by calculating the ROUGE value between C_i' and the C_*. Then we sort the candidate summaries in descending order based on the ROUGE-1 score, obtaining n sorted candidate summaries $C_1, C_2, \ldots, C_n$. We take the first two candidate summaries (C_1, C_2) as the positive pair for each document. For the construction of negative samples, one approach is to use all candidate summaries in the same batch. Gao et al. (2021) demonstrate that adding other negative examples that contradict positive examples can significantly improve model performance. Inspired by their work, we take the lowest-quality summary in each document as a hard negative.

For a mini-batch that contains K documents, the final contrastive learning objective can be designed as follows:

$$L = -\sum_{i=1}^{i=K} log\frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{j=K}(e^{sim(h_i, h_j^+)/\tau} + e^{sim(h_i, h_j^-)/\tau}}, \quad (2)$$

where h_i, h_i^+ are the vector representation of candidate summaries C_1 and C_2 in document i, h_j^+, h_j^- are the vector representation of C_1 and C_n in the $j-th$ document in the mini-batch.

3 Experiment

3.1 Datasets

We conduct experiments on three datasets to verify the effectiveness of CSSS, namely, CNN/Daily Mail (Hermann et al., 2015), XSum (Narayan et al., 2018) and Reddit (Kim et al., 2019). CNN/Daily is a widely used news dataset for extractive summarization. We use the non-anonymized version as See et al. (2017). XSum is also a news dataset, in which each article contains a one-sentence summary to answer the question "What is this article about?". Reddit is a highly abstractive dataset collected from a discussion forum platform. We use the TIFU-long version of Reddit.

3.2 Implementation Detail

We use base version of BERT and RoBERTa in all experiments. We conduct our experiments on two NVIDIA RTX 6000 GPUs and train the model with a batch size of 256. The length of each candidate summary is limited to 80 tokens. More experimental details can be seen in Appendix A.

3.3 Experimental Results

The effect of temperature τ on the experimental results is placed in Appendix B. Here we use the best value of τ over each dataset.

3.3.1 Results on Unsupervised Summarization

In unsupervised extractive summarization, we conduct our experiments on CNN/Daily Mail dataset. The results are shown in Table 1. We list strong baselines with different in the first section.Although these methods are effective, they are essentially sentence-level methods with individual scoring process. CSSS can implement transition from sentence-level to summary-level methods easily in extractive summarization. The second section presents in the table presents our results. We can

Model	R-1	R-2	R-L
LEAD-3	40.49	17.66	36.75
TextRank (Mihalcea and Tarau, 2004)	33.85	13.61	30.14
LexRank (Erkan and Radev, 2004)	34.68	12.82	31.12
PacSum (Zheng and Lapata, 2019)	40.70	17.80	36.90
FAR (Liang et al., 2021)	40.83	17.85	36.91
STAS (Xu et al., 2020b)	40.90	18.02	37.21
STAS+PacSum (Xu et al., 2020b)	41.26	18.18	37.48
CSSS(BERT)	**42.50**	**19.89**	**38.62**
CSSS(RoBERTa)	**42.70**	**19.90**	**38.77**

Table 1: Results on CNN/Daily Mail test set in unsupervised summarization.R-1,R-2,R-L denote the scores of ROUGE-1,ROUGE-2,ROUGE-L.

see that our CSSS outperforms all strong baselines by wide margins.

3.3.2 Results on Supervised Summarization

Model	R-1	R-2	R-L
LEAD-3	40.49	17.66	36.75
ORACLE	52.59	31.23	48.87
BERTSUMEXT$_{base}$ (Liu and Lapata, 2019)	43.25	20.24	39.63
BERTSUMEXT$_{large}$ (Liu and Lapata, 2019)	43.85	20.34	39.90
DiscoBERT$_{base}$(Xu et al., 2020a)	43.77	20.85	40.67
ETCSum$_{base}$(Narayan et al., 2020)	43.84	20.80	39.77
ARedSum$_{base}$(Bi et al., 2021)	43.43	20.44	39.83
ThresSum$_{large}$(Jia et al., 2021b)	44.59	21.15	40.76
DifferSum$_{large}$(Jia et al., 2021a)	44.70	21.36	40.83
MATCHSUM$_{base}$(Zhong et al., 2020)	44.41	20.86	40.55
CSSS(BERT$_{base}$)	**45.16**	**22.08**	**41.34**
CSSS(RoBERTa$_{base}$)	**45.49**	**22.36**	**41.67**

Table 2: The supervised extractive results on CNN/Daily Mail.

Results on CNN/Daily Mail Table 2 shows our results on CNN/Daily Mail. The first part shows the LEAD-3 baseline and the ORACLE upper bound, the second part lists sentence-level extractive summarization models in recent years, the third part presents the summary-level method proposed by Zhong et al. (2020).

We present our method in the fourth part. The results show that we outperform all extractive models by a large margin. Compared with the state-of-the art sentence-level model namely DifferSum (Jia et al., 2021a), we achieve achieves 0.46/0.72/0.51 improvements on ROUGE-1, ROUGE-2, and ROUGE-L when only using base version of BERT. Compared with summary-level method, MATCHSUM, CSSS further uses the set relationship between candidate summaries to select the summary,and has a significant improvement in results, which proves that CSSS is more effective in selecting summaries.

Results on XSum and Reddit Different from CNN/Daily Mail, XSum and Reddit are two datasets with short summaries. We did our experiment on these two datasets to study whether CSSS could perform better than other strong extractive models when dealing with short sentence summaries. Besides, we investigated the effect of the number of sentences in the pruned document on final results.

Model	R-1	R-2	R-L
	XSum		
BERTEXT (Zhong et al., 2020)	22.86	4.48	17.16
MATCHSUM (Zhong et al., 2020)	24.86	4.66	18.41
CSSS (Num=3)	**25.77**	**5.61**	18.91
CSSS (Num=4)	25.51	5.42	19.06
CSSS (Num=5)	25.10	5.58	**19.20**
	Reddit		
BERTEXT	23.86	5.85	19.91
MATCHSUM	25.09	6.17	20.13
CSSS (Num=3)	26.63	6.57	21.25
CSSS (Num=4)	**27.12**	**7.10**	**21.57**
CSSS (Num=5)	26.71	6.77	21.34

Table 3: Extractive results on XSum and Reddit. Num indicates how many sentences extracted as a pruned document.

As shown in Table 3, we compare our CSSS

with BERTEXT and MATCHSUM. BERTEXT is a sentence-level method using BERTSUM (Liu and Lapata, 2019) without trigram blocking as the sentence extractor. We outperform both two extractive methods by a large margin. We found that there is an optimal number of pruned sentences for different datasets, which is approximately two more than the maximum number of summary sentences.

3.4 Ablation Studies

Model	R-1	R-2	R-L
	CNNDM		
CSSS (BERT$_{Ori}$)	44.15	21.49	40.39
CSSS (BERTcl)	**45.16**	**22.08**	**41.34**
	Xsum		
CSSS (BERT$_{Ori}$)	25.43	5.49	18.63
CSSS (BERTcl)	**25.77**	**5.61**	**18.91**
	Reddit		
CSSS (BERT$_{Ori}$)	26.01	6.26	20.61
CSSS (BERTcl)	**27.12**	**7.10**	**21.57**

Table 4: The results with fine-tuned BERT and RoBERTa on three datasets. BERT$_{Ori}$ is the original base version of BERT, BERT$_{cl}$ is a fine-tuned BERT by the contrastive learning-based training objective.

We performed ablation study on three datasets to demonstrate the effectiveness of the contrastive learning-based training objective. The results are shown in Table 4. We can see a significant improvement after using the fine-tuned pretrained model.

3.5 Analysis

3.5.1 Positional Bias

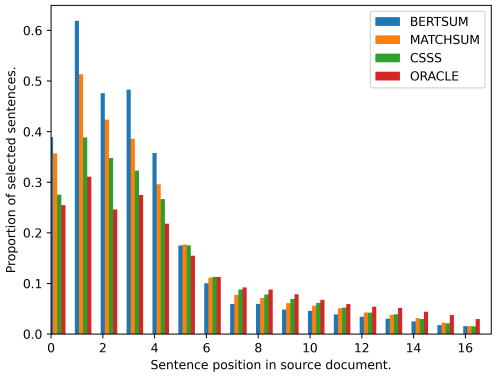

Figure 2: The proportion of summary sentences at different positions in the source document

.

The positional bias on the CNN/Daily Mail dataset is a common phenomenon, which usually dominates neural extractive summarizers to select the sentences at the beginning of a document (Xing et al., 2021).For the summaries generated by the different models, we looked at the position of the sentences in the source document. The proportion of summary sentences at different positions in the source document is shown in Figure 2.

We compare CSSS with BERTSUM, MATCHSUM and ORACLE. The ORACLE summary is generated by a greedy algorithm that maximizes the ROUGE-2 score against the gold summary. We can see that the distribution of the ORACLE summary sentences across the documents is fairly smooth, and can be used as a measure for other methods. Compared with BERTSUM and MATCHSUM, the summaries selected by CSSS are more similar to ORACLE summary, which demonstrates the effectiveness of CSSS in reducing positional bias.

3.5.2 Number of summary sentences

We studied the number of sentences in the summary selected by MATSCHSUM and CSSS in XSum dataset to demonstrate the superiority of CSSS. As

Summary	MATCHSUM	CSSS	ORACLE
$Num = 1$	22.4%	50.4%	100%
$Num = 2$	77.6%	46.6%	0

Table 5: The percentage of summaries with Num sentences in all summaries.

shown in Table 5, MATCHSUM select more summaries with more sentences in XSum dataset. This may because the matching model becomes difficult to select the correct summary when faced with short summary datasets, so it achieves the highest semantics of the document by selecting the summary with more sentences. Compare with MATCHSUM, the summaries selected by CSSS are not sensitive to the number of sentences and are closer to ORACLE, indicating that CSSS reduces the bias towards summaries with few sentences.

4 Conclusion

In this paper, we propose a novel candidate summary selection strategy (CSSS) for summary-level extractive summarization. We show how CSSS could be efficiently applied in both unsupervised and supervised extractive task. We also propose a task-specific contrastive learning framework to learn better vector representations for candidate summaries. Experimental results show that we beat the current state-of-the-art extractive models on three benchmark datasets, which demonstrates the effectiveness of our method.

Acknowledgements

We thank our teachers for their careful guidance.We also thank the members of our NLP group for helpful discussion.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Keping Bi, Rahul Jha, Bruce Croft, and Asli Celikyilmaz. 2021. AREDSUM: Adaptive redundancy-aware iterative sentence ranking for extractive document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 281–291, Online. Association for Computational Linguistics.

Benjamin Börschinger and Mark Johnson. 2011. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia.

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2109.09209*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Ruipeng Jia, Yanan Cao, Fang Fang, Yuchen Zhou, Zheng Fang, Yanbing Liu, and Shi Wang. 2021a. Deep differential amplifier for extractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 366–376, Online. Association for Computational Linguistics.

Ruipeng Jia, Yanan Cao, Haichao Shi, Fang Fang, Pengfei Yin, and Shi Wang. 2021b. Flexible non-autoregressive extractive summarization with threshold: How to extract a non-fixed number of summary sentences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13134–13142.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021. Classic: Continual and contrastive learning of aspect sentiment classification tasks. *arXiv preprint arXiv:2112.02714*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. 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 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. *arXiv preprint arXiv:2111.02194*.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697, Online. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

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, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159, Online. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Linzi Xing, Wen Xiao, and Giuseppe Carenini. 2021. Demoting the lead bias in news summarization via alternating adversarial learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 948–954, Online. Association for Computational Linguistics.

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

Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020b. Unsupervised extractive summarization by pre-training hierarchical transformers. In

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1784–1795, Online. Association for Computational Linguistics.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2021. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. *arXiv preprint arXiv:2109.08678*.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. 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, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Experimental Details

For different datasets, we list the training details in following Table 6.

Summary	CNNDM	XSUM	Reddit
Learning Rate	$5e^{-5}$	$5e^{-5}$	$5e^{-5}$
Training Steps	15000	6000	2000
Evaluate Steps	500	300	100
Temperature τ	0.04	0.06	0.06

Table 6: Training details for different datasets. Evaluate Steps means that we evaluate the model every 500,300 and 100 training steps for three datasets.

B Temperature τ

Temperature τ helps discriminate positive and negative samples by controlling the strength of penalties on the hard-negative samples. We present the effect of temperatures τ on the model performance in different datasets in Figure 3.

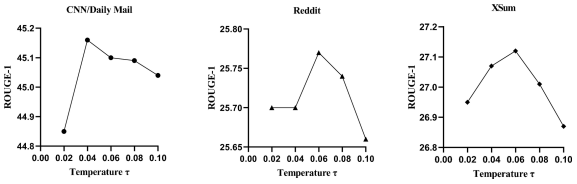

Figure 3: Effect of temperature τ on model performance in different datasets. We use ROUGE-1 as the measure of model performance.