Demoting the Lead Bias in News Summarization via Alternating Adversarial Learning

Linzi Xing*, **Wen Xiao***, **Giuseppe Carenini**
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{lzxing, xiaowen3, carenini}@cs.ubc.ca

Abstract

In news articles the lead bias is a common phenomenon that usually dominates the learning signals for neural extractive summarizers, severely limiting their performance on data with different or even no bias. In this paper, we introduce a novel technique[1] to demote lead bias and make the summarizer focus more on the content semantics. Experiments on two news corpora with different degrees of lead bias show that our method can effectively demote the model's learned lead bias and improve its generality on out-of-distribution data, with little to no performance loss on in-distribution data.

1 Introduction

Neural extractive summarization, which produces a short summary for a document by selecting a set of representative sentences, has shown great potential in real-world applications, including news (Cheng and Lapata, 2016; Nallapati et al., 2017) and scientific paper summarization (Cohan et al., 2018; Xiao and Carenini, 2019). Typically, a general-purpose extractive summarizer learns to select the most important sentences from a document to form the summary by considering their content salience, informativeness and redundancy. However, when restricted to a specific domain, the summarizer can learn to exploit particular biases in the data, the most famous of which is the *lead bias* in news (Nenkova et al., 2011; Hong and Nenkova, 2014); namely that sentences at the beginning of a news article are more likely to contain summary-worthy information. As a result, not surprisingly, such bias is strongly captured by neural extractive summarizers for news, for which the sentence positional information tends to dominate the actual content of the sentence in model prediction (Jung et al., 2019; Grenander et al., 2019; Zhong et al., 2019a,b).

While learning a summarizer reflecting the biases in the training dataset is completely fine when the summarizer is going to be deployed to summarize documents having similar biases, it would be problematic when the model was applied to deal with documents coming from a mixture of datasets with different degrees of such biases. In this paper, we address this problem in the context of the lead bias in the news domain by exploring ways in which an extractive summarizer for news can be trained so that it learns to balance the lead bias with the content of the sentences, resulting in a model that can be applied more effectively when the target documents belong to news datasets in which the lead bias is present in rather different degrees.

Recently, Grenander et al. (2019) proposes two preliminary solutions. One is to pretrain the summarizer on an automatic generated "unbiased" corpus where the document sentences are randomly shuffled, which however has the negative effects of preventing the learning of inter-sentential information. The other, which can be only applied to RL-based summarizers, is to add an explicit auxiliary loss to directly balance position with content. Alternatively, Zhong et al. (2019b) and Wang et al. (2019) investigate strategies to train the summarizer on multiple news datasets with different degrees of lead bias, but this may still be problematic when we apply the trained summarizer to the documents with lead bias not covered in the training data. Outside the summarization area, methods have also been proposed to eliminate data biases for other NLP tasks like text classification or entailment (Kumar et al., 2019; Clark et al., 2019, 2020).

Inspired by Kumar et al. (2019), we have developed an alternating adversarial learning technique to demote the summarizer lead bias, but also maintain the performance on the in-distribution

* The first two authors contributed equally to this work.
[1] https://github.com/lxing532/Debiasing

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 948–954
August 1–6, 2021.

data. We introduce a position prediction component as an adversary, and optimize it along with the neural extractive summarizer in an alternating manner. Furthermore, in contrast with Grenander et al. (2019) and Wang et al. (2019), our proposal is model-independent and only requires one type of news dataset as training input.

In this paper, we apply our proposed method to a biased transformer-based extractive summarizer (Vaswani et al., 2017) trained on CNN/DM training set (Hermann et al., 2015) and conduct experiments on two test sets with different degrees of lead bias: CNN/DM and XSum (Narayan et al., 2018), for in-distribution and generality evaluation respectively. The experimental results indicate that our proposed "debiasing" method can effectively demote the lead bias learned by the neural news summarizer and improve its generalizability, while still mostly maintaining the model's performance on the data with a similar lead bias.

2 Proposed Debiasing Method

Our method aims to demote the lead bias learned by the summarizer and encourage it to select content based more on the semantics covered in sentences. As shown in Figure 1, our method comprises two components: one for *Summarization* (red) and the other for sentence *Position Prediction* (green).

2.1 Summarization Component

Following previous work, we formulate extractive summarization as a sequence labeling task (Xiao and Carenini, 2019, 2020; Xiao et al., 2020). For a document $d = \{s_1, s_2, ..., s_k\}$, each sentence will be assigned a score $\alpha \in [0, 1]$. The summary will then be formed with the highest scored sentences. We adopt a transformer-based model (Vaswani et al., 2017) as our basic "biased" summarization component (red in Fig. 1), as shown to be heavily impacted by the lead bias (Zhong et al., 2019a). This component contains a transformer-based encoder Enc_{θ_t} and a multilayer perceptron (MLP) decoder Dec_{θ_s}, parameterized by θ_t and θ_s respectively. We use the averaged word embedding from Glove as sentence embedding as suggested in Kedzie et al. (2018). We optimize this summarization system by minimizing the loss:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} CE(\alpha_i, y_i) \qquad (1)$$
$$\alpha_i = Dec_{\theta_s}(Enc_{\theta_t}(s_i))$$

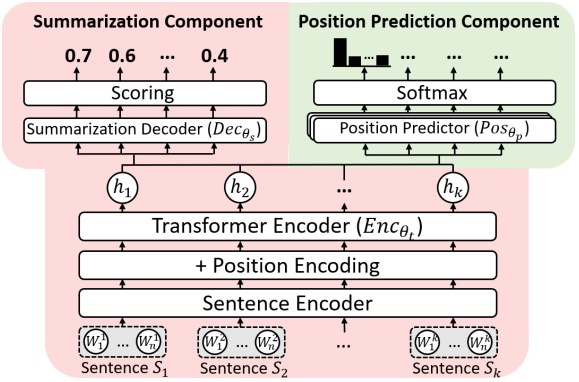

Figure 1: The overall architecture of our proposed lead bias demoting method.

where CE denotes the cross-entropy loss and $y_i \in \{0, 1\}$ is the ground truth label for sentence s_i, representing if s_i is selected to form the summary.

2.2 Position Prediction Component

Our goal is to train the summarization model to make accurate predictions based more on the sentence semantics, rather than whether the sentence is in the lead position. More specifically, we aim to design an encoder network Enc_{θ_t} to output the set of contextualized sentence representations $\mathbf{H} = \{h_1, ..., h_k\}$ which cover less sentence positional information, so that the following decoder Dec_{θ_s} will make predictions depending less on such positional information. To achieve this, the first step is to understand how much and in what form the positional information is encoded in Enc_{θ_t}. Therefore, we propose a position prediction network to learn to predict the position of sentences in a document based only on $\mathbf{H}$. Intuitively, the higher accuracy this component can achieve, the more positional information is contained in $\mathbf{H}$. This position prediction component will then play the role of an adversary module to demote the influence of lead bias presented in the training phase of the summarization component.

Concretely, because predicting the exact position for each sentence would require an extremely large set of labels with a skewed distribution, we choose to predict the portion of the document each sentence belongs to. In particular, once we obtain the set of contextualized sentence representations $\mathbf{H}$ from the encoder network Enc_{θ_t}, we initialize a MLP (parameterized by θ_p and followed by Softmax) as the position prediction component Pos_{θ_p} (green in Fig 1). In essence, this component Pos_{θ_p} takes $\mathbf{H}$ as input and outputs a M-dimensional multinomial distribution for each

sentence to represent its position in a document. More formally, $Pos_{\theta_p}(h_i) = (\hat{p}_1^{(i)}, .., \hat{p}_j^{(i)}, .., \hat{p}_M^{(i)})$ where $\sum_{j=1}^{M} \hat{p}_j^{(i)} = 1$. $\hat{p}_j^{(i)}$ is the predicted probability of the ith sentence belongs to the jth portion of a document when the document is divided into M parts (M is a tunable hyperparameter). We use the cross-entropy loss to optimize Pos_{θ_p} to extract sentence positional signals encoded in the system:

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N} CE(Pos_{\theta_p}(h_i), p_i) \qquad (2)$$

where p_i is the true position of sentence i.

2.3 Alternating Adversarial Learning

To demote the influence of positional bias and balance it with the sentence semantics in the summarization system, we want to modify the encoder to produce **H**, which can still be accurate for summary generation but fail at sentence position prediction. We achieve this by alternatingly executing "Position learning" and "Position debiasing", as proposed in Kumar et al. (2019) and presented in Algorithm 1. In the "Position learning" phase, once a pretrained summarization system is obtained, we first fix its weights and train an adversary network $Pos^*_{\theta_p}$ (sentence position predictor) to extract the positional information contained in the encoder. Then in the "Position debiasing" phase, we fix the weights of $Pos^*_{\theta_p}$ and update the parameters of the summarization component to maximize the position prediction loss of adversary (L_{adv} in eq 3) while minimizing the summarization loss L_1:

$$\begin{aligned} L_3 &= \beta L_1 + (1-\beta) L_{adv} \\ L_{adv} &= -\frac{1}{N}\sum_{i=1}^{N} CE(Pos^*_{\theta_p}(h_i), U_M) \end{aligned} \qquad (3)$$

To maximize the position prediction loss, the fixed adversary $Pos^*_{\theta_p}$ should ideally output the uniform distribution, $U_M = (\frac{1}{M}, ..., \frac{1}{M})$, for the position prediction of each sentence. β is the trade-off parameter tuned at validation stage to control the degree of lead bias demoting.

In practice, we notice that reusing the same adversary for all iterations will make the positional signals not weakened but instead encoded in a different way. To avoid this problem, we follow Kumar et al. (2019) to use multiple adversaries (parameterized with $[\theta_p^{(1)}, ..., \theta_p^{(N)}]$ in Algorithm 1), making it more difficult for the encoder to keep the sentence positional signals by encoding them into a more implicit format for position predictor to learn.

Algorithm 1: Alternating Adversarial Learning

Result: $\theta_t, \theta_s, [\theta_p^{(1)}, \cdots, \theta_p^{(N)}]$
Randomly initialize θ_t, θ_s;
while *not converged* **do**
  Sample a minibatch of training samples;
  Update θ_t, θ_s with gradients regarding L_1;
end — Pretraining
while $i <= N$ **do**
  for *p steps* **do**
    Randomly initialize $\theta_p^{(i)}$;
    Sample a minibatch of training samples;
    Fix θ_t, θ_s, update $\theta_p^{(i)}$ with gradients regarding L_2;
  end — Position Learning
  for *s steps* **do**
    Sample a minibatch of training samples;
    Randomly select $\theta_p^{(cur)}$ from $[\theta_p^{(1)}, \cdots, \theta_p^{(i)}]$;
    Fix $\theta_p^{(cur)}$, update θ_t and θ_s with gradients regarding L_3;
  end — Position Debiasing
  $i = i + 1$;
end

3 Experiments and Analysis

3.1 Datasets

We use the standard CNN/DM dataset (204,045 training, 11,332 validation and 11,334 test data) (Hermann et al., 2015) for training since it is one of the mainstream news datasets with observed lead bias (Jung et al., 2019; Grenander et al., 2019). For model evaluation, we use the test set of CNN/DM to evaluate model's in-distribution performance, as well as the test set of XSum (Narayan et al., 2018), which consists of 11,334 datapoints, to evaluate model's generality when transferred to less biased data. The empirical analysis in Narayan et al. (2018) and Jung et al. (2019) shows the documents and summaries in XSum are shorter and have less lead bias compared to CNN/DM.

3.2 Experimental Design

Baselines: We compare our proposal with various baselines (see Table 1). The top section of Table 1 presents ***Lead*** baseline and ***Oracle***. For CNN/DM, lead baseline refers to ***Lead-3*** and for XSum, it refers to ***Lead-1***. The middle section of Table 1 contains the basic transfomer-based summarizer accepting "sentence representation + position encoding" as input, and its two variants, one without positional encoding, while the other with only positional encoding as input. The bottom section contains ***Shuffling*** (Grenander et al., 2019), which

Model	CNN/DM				XSum			
	R1	R2	RL	Mean	R1	R2	RL	Mean
Lead	40.30	17.52	36.54	31.45	16.32	1.60	11.96	9.96
Oracle	56.04	33.10	52.29	47.14	30.98	8.98	23.51	21.16
Basic Transformer (Vaswani et al., 2017)	41.02	18.39	37.39	32.27	16.79	1.84	12.33	10.32
– No Position Encoding	37.82↓	15.59↓	34.32↓	29.24	**18.29**↑	**2.53**↑	**13.45**↑	**11.42**
– Only Position Encoding	40.13↓	17.36↓	36.38↓	31.29	16.22↓	1.62↓	11.90↓	9.91
Learned-Mixin (Clark et al., 2019)	40.72↓	18.27	37.17↓	32.05	16.67	1.91↑	12.28	10.29
Shuffling (Grenander et al., 2019)	**41.00**	**18.43**	**37.37**	**32.27**	16.98↑	1.96↑	12.48↑	10.47
Our Method	40.88↓⇓	18.37	37.27↓	32.18	17.20↑⇑	1.99↑⇑	12.63↑⇑	10.61

Table 1: The ROUGE-1/2/L F1 scores and "Mean" (mean of ROUGE-1/2/L) on CNN/DM and XSum test data. The best and second best performances over the basic transformer are in **bold** and underlined. ↑/↓ indicates the results are significantly higher/lower than Basic Transformer and ⇑/⇓ indicates the results are significantly higher/lower than Shuffling ($p < 0.01$ with bootstrap resampling test (Lin, 2004)).

Model	$\mathbf{D}_{early}$	$\mathbf{D}_{middle}$	$\mathbf{D}_{late}$
Lead-3	49.33	30.90	19.80
Oracle	49.51	47.02	43.81
Basic Transformer	44.30	31.91	22.65
– No Position Encoding	16.07	16.88	18.59
– Only Position Encoding	$48.65^{*\dagger\ddagger}$	30.97	19.70
Learned-Mixin	40.45	31.82	22.70
Shuffling	42.69	31.91	$22.99^{*\dagger\ddagger}$
Our Method	42.67	$32.18^{*\dagger\ddagger}$	$22.85^{*\dagger}$

Table 2: Avg. of ROUGE-1/2/L F1 scores on D_{early}, D_{middle} and D_{late}. Results significantly better than Basic Transformer on ROUGE-1/2/L are marked with $*$, †, and ‡ respectively.

is a method proposed lately for summarization lead bias demoting, and ***Learned-Mixin*** (Clark et al., 2019), which is a general debiasing method proposed to deal with NLP tasks when the type of data bias in the training set is known and bias-only model is available. In our case, the data bias is lead bias and the bias-only model is the transformer trained with only positional encoding as input.

Implementation Details: All the transformer-based models have the same setting as the standard transformer (Vaswani et al., 2017), with 6 layers, 8 heads per layer, and $d_{model} = 512$. We use Adam to train all the models with scheduled learning rate with warm-up (initial learning rate $lr = 2e - 3$). We choose the top-3 sentences to form the final summary for CNN/DM and the top-1 sentence for XSum due to the different average summary lengths. The class number of sentence position M is set to 10 and trade-off parameter β is set to 0.9 (searched from 0 to 1, by increasing 0.1 for each step). We tune these hyper-parameters on a "balanced" validation set sampled from the standard CNN/DM validation data.

3.3 Results and Analysis

Table 1 reports the performance of the chosen baselines and our proposal on CNN/DM test set, which has the same data distribution as the training data, and XSum test set, which is from another news resource and with much less lead bias than CNN/DM.

From the middle section of Table 1, we observe that if we withhold the position cues (*No Position Encoding*) by using only semantic representation as input, the model's performance drops considerably on CNN/DM, but remarkably increase on XSum. In contrast, if we merely use position cues as input (*Only Position Encodings*), the decrease of the performance on CNN/DM becomes much more modest, while there is substantial performance drop on XSum. These results confirm that positional signal is a rather important feature for bias-relied neural summarizers. However, relying too much on it will also limit model's generality when applied to the dataset with less bias than the training samples. Therefore, seeking strategies to balance the semantics and position features is crucial for the neural extractive summarization for news.

When we compare the lead bias demoting methods presented at the bottom of Table 1, our proposal and *Shuffling* give significant performance boosting on XSum, while *Learned-Mixin* results in performance decrease on both datasets. Comparing our method and *Shuffling* directly, while they are essentially equivalent on maintaining the performance on the in-distribution CNN/DM data (0.09 difference in terms of the average of ROUGE scores (ROUGE-Mean)), our method provides a significant improvement on XSum, and outperforms *Shuffling* and the basic transformer by 0.14 and 0.29 on ROUGE-Mean respectively. It is noteworthy that the transformer without position encoding achieves

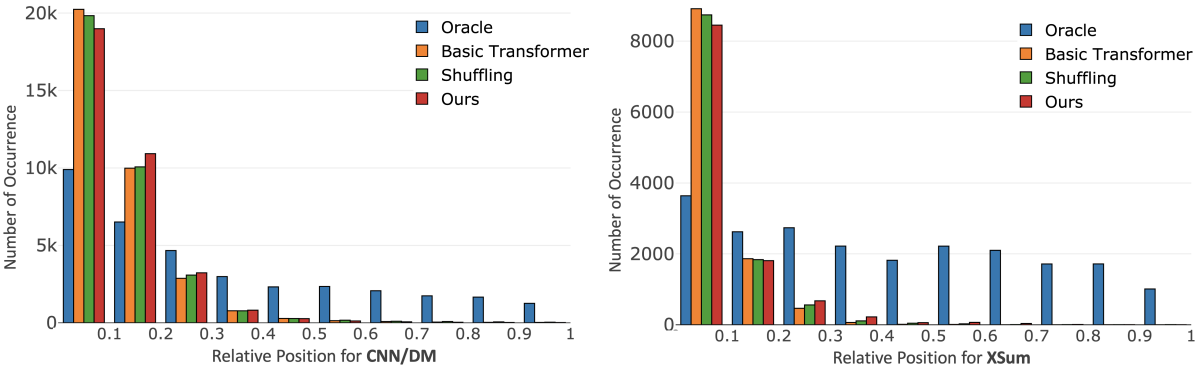

Figure 2: Relative position distributions of selected sentences in the original document of two testing corpora (CNN/DM and XSum), obtained by different lead bias demoting strategies.

the best performance on XSum. However, it is the worst system on in-distribution data. Throughout all the comparisons, our proposal can best balance the sentence position and content semantics.

To more deeply investigate the behavior of our demoting method on the documents whose summary sentences are from different document portions, we follow Grenander et al. (2019) to create three subsets, D_{early}, D_{middle}, D_{late}, from the CNN/DM testing set. Documents are ranked by the mean of their summary sentences' indices in ascending order, and then the top-ranked 100 documents, the 100 documents closest to the median, and the bottom-ranked 100 documents are selected to form D_{early}, D_{middle}, D_{late}[2]. Results in Table 2 show that even if our model does not match the basic transformer on documents in D_{early}, it does yield benefits for both D_{middle} and D_{late} with significant improvements, while the competitive baseline *Shuffling* only achieves that on D_{late}.

Position of Selected Content: To more explicitly investigate how well the prediction of different models fits the ground-truth sentence selection (*Oracle*), we compare the relative position of the selected content of our method with the undebiased model (*Basic Transformer*) and the most competitive debiased model (with *Shuffling*), as illustrated in Figure 2. We can observe that: (1) CNN/DM contains much more lead bias than XSum, shown by a more right-skewed histogram for *Oracle*. Thus, the basic transformer trained on it is also heavily impacted by the lead bias and tends to select sentences $\in [0, 0.1]$ with much higher probability even on the less biased XSum. (2) While *Shuffling* and our method can both effectively demote the extreme trend towards selecting sentences in the lead position, our method seems to be sightly better at encouraging the model to select sentences with higher relative position.

[2]Due to the common generation mechanism of oracles, the number of sentences in the oracle is not fixed. For fair comparison, we only consider adding documents with the oracle having exactly 3 sentences into D_{early}, D_{middle}, D_{late}.

4 Conclusion and Future Work

We propose a lead bias demoting method to make news extractive summarizers more robust across datastets, by optimizing a position prediction and a summarization component in an alternating way. Experiments indicate that our method improves model's generality on out-of-distribution data, while still largely maintaining its performance on in-distribution data. As such, it represents the best viable solution when at inference time input documents may come from an unknown mixture of datasets with different degrees of position bias.

For the future, we plan to explore more sophisticated and effective methods (e.g., adjusting the lead bias online) and infuse them together with neural abstractive summarization models, known to generate more succinct and natural summaries. Another interesting direction for future work can be exploring the potential of applying our proposed bias demoting strategy to other tasks, which can also be framed as the sequence labeling problem and possibly troubled by biases in the training data (e.g., Topic Segmentation (Xing et al., 2020) and Semantic Role Labeling (Ouchi et al., 2018)).

Acknowledgments

We thank the anonymous reviewers and the UBC-NLP group for their insightful comments and suggestions. This research was supported by the Language & Speech Innovation Lab of Cloud BU, Huawei Technologies Co., Ltd.

References

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, Berlin, Germany. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. 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 6019–6024, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, 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*, volume 28, pages 1693–1701. Curran Associates, Inc.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, Gothenburg, Sweden. Association for Computational Linguistics.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. 2019. Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization. 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 3324–3335, Hong Kong, China. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. 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 4153–4163, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 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. AAAI'17, page 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.

Ani Nenkova, Sameer Maskey, and Yang Liu. 2011. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 3, Portland, Oregon. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019. Exploring domain shift in extractive text summarization. *CoRR*, abs/1908.11664.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, Suzhou, China. Association for Computational Linguistics.

Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help ! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.

Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. 2020. Improving context modeling in neural topic segmentation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 626–636, Suzhou, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019a. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Danqing Wang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019b. A closer look at data bias in neural extractive summarization models. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 80–89, Hong Kong, China. Association for Computational Linguistics.