

Content selection in abstractive summarization with biased encoder mixtures

First Author¹[0000-1111-2222-3333] and Second Author²[1111-2222-3333-4444]

¹ Princeton University, Princeton NJ 08544, USA

² Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com

Abstract. Current abstractive summarization models consistently outperform extractive counterparts yet are unable to close the gap with theoretical extractive upper bound. Recent research suggests that the reason lies in the lack of planning and bad sentence-wise saliency intuition. Existing solutions to the problem either require new fine-tuning sessions to accommodate the architectural changes or disrupt the natural information flow limiting the utilization of accumulated global knowledge. Inspired by text-to-image result blending techniques we propose a plug-and-play alternative that preserves the integrity of the original model, biased encoder mixture. Our approach utilizes attention masking and Siamese networks to reinforce the signal of salient tokens in encoder embeddings and guide the decoder to more relevant results. The evaluation on four datasets and their respective state-of-the-art abstractive summarization models demonstrate that biased encoder mixture outperforms the attention-based plug-and-play alternatives even with static masking derived from sentence saliency positional distribution.

Keywords: Abstractive Summarization, Content Control, Attention Mechanism.

1 Introduction

The quality of automatic summarization has improved substantially in the past decade thanks to advances in neural language modeling. The basis of the state-of-the-art approaches are pre-trained transformer language models that efficiently integrate global contextual knowledge through multi-head attention mechanism. The pre-trained models vary in both attention mechanism architecture and pre-training tasks, however the baseline generative language models such as BART [1] and Pegasus [2] have the best average performance. Recent works [3, 4, 5] showed that with proper fine-tuning procedure these generative models outperform methods with additional input [6] or complex post-generation result refinement [7, 8]. At the same time, it was also shown that abstractive summarization approaches based solely on pre-trained language models lack understanding of extractive side of the task in terms of sentence-wise saliency [6, 9].

Improving extractive intuition can significantly boost the quality of generated summaries as well as provide better control over the generation process. However, the main

issue with existing approaches is the need to alter either the architecture of the model [9] or the input format [6] which in turn requires a new fine-tuning session. This trait renders these approaches incompatible with existing state-of-the-art summarization models that were tailored to specific data representations and pre-trained language models [3, 4, 5].

Inspired by text-to-image result blending techniques [10], we propose Biased encoder mixture, an alternative solution to integrating the extractive summarization knowledge without interfering with the existing abstractive framework structure. Our method exploits the attention mechanism and Siamese networks to reinforce the signal of salient sentences in document embedding thus encouraging the decoder to better utilize the selected content. Unlike previous attention-based approaches [11, 12, 13] encoder mixture retains the original information flow and doesn't introduce additional model parameters.

We test Biased encoder mixture on four diverse datasets of CNN / Daily Mail, Xsum, ArXiv and SAMSUM. The evaluation demonstrates that the proposed approach consistently outperforms previous approaches even with static positional masking derived from extractive oracle sentence distribution. Employing an auxiliary extractive summarization model for dynamic masking further boosts the quality of generated summaries bringing up to a 7% ROUGE-2 relative increase to state-of-the-art abstractive summarization models.

2 Related work

One of the first models to surpass extractive baselines was Pointer-generator network [14] that utilized explicit copy-mechanism to bridge the gap between abstractive and extractive models. This model was further improved with auxiliary content selection module [15] that limited the copy attention of Pointer generator to the most salient parts of source document. Further development of the idea led to introduction of EASE [11] joint extractive-abstractive training that optimizes content selector to maximize the quality of abstractive summarizer. The maximal effect of joint training was achieved with SEASON [9] model that integrated the content selector in Transformer encoder-decoder abstractive summarization model by training encoder to provide sentence saliency weights for cross-attention.

A similar line of work augmented the content control signal through additional network components. Lebanoff et al. [16] proposed complementing word embeddings with highlight embeddings extracted from content selector to perform cascade summarization. Ji et al. [17] improved the approach by extending to inter-layer encoder embeddings and constructing highlight embeddings within the main model. To preserve initial embeddings Dou et al. [6] proposed GSum approach, encoding arbitrary guidance query in Siamese network fashion and passing it to additional cross-attention block in decoder layer. Cao et al. [12] adapted Bottom-up [15] attention modulation approach to Transformer architecture and proposed guiding individual cross-attention heads. Xiao et al. [13] combined attention modulation and GSum by introducing relevance attention that corrects cross-attention weights.

3 Motivation and methodology

Extractive summarization can be thought of as a variant of abstractive summarization whose phrasal vocabulary is constrained to fragments of the source document. However, in practice the neural abstractive summarization models are trained in the same fashion as neural machine translation models, which is estimation of conditional language distribution that models the probability of the next generated token. This means that to copy an exact text fragment the model must generate each token separately relying on likelihood estimation of each subsequence. Moreover, the decision of copying the fragment is based on the inference-time history of generated tokens which contradicts the natural human behavior of initially extracting the salient content and only then fusing and paraphrasing the fragments into sentences [18].

With the current abstractive summarization paradigm, the preliminary salient content selection can be done in two ways. Amid the rise of cloud LLMs (e.g. ChatGPT) the most popular is a two-step approach of reducing the document to the most salient fragments using the extractive summarization model and then summarizing these fragments with abstractive model [19, 20]. While the approach is easy to implement it is let down by the suboptimality of extractive summarization systems.

The performance of current state-of-the-art systems is substantially lower than theorized upper bound, known as extractive oracle which is commonly [21] obtained by solving the following problem:

$$\text{Oracle}(X, y) = \arg \max_{x_i \subset X} \text{ROUGE-2}(x_i, y)$$

where x_i is subset of sentences from document X , y is the reference summary. This ideal solution is limited to the most relevant sentences and hence is not guaranteed to contain all necessary information for deduction of factually correct abstractive summary. On the other hand, the existing extractive summarization systems fail to include this information either. It was shown [22] that the extracted subsets usually distort information and create discourse and coreference resolution errors leading to unintended results in second-stage abstractive summarization systems.

A more reliable way to filter out the salient content is through input suppression. The idea is to assign weights to tokens with respect to their relevance of content and use the weights as an extractive guidance in the generation process. This approach allows to retain the full information flow and at the same time ensure that the abstractive summarization model would follow the predefined content selection plan. Token weighting is commonly performed through either attention mechanism [15, 12, 11, 13] or additional architectural components/input [6, 9, 16, 17]. In the era of pre-trained language models, the former plug-and-play approach has a higher preference as it doesn't interfere with knowledge accumulated during pre-training stage.

3.1 Attention distribution discrepancy

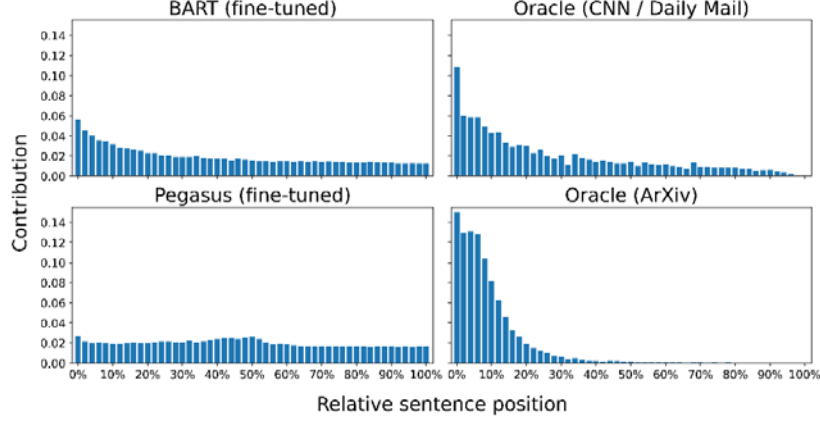


Fig. 1. Comparison of sentence-wise attention and Extractive Oracle positions

Integrating the extractive knowledge has been repeatedly shown to significantly boost the quality of generated summaries [6, 23, 12]. However, none of the works investigated the reasons behind the such positive changes.

We measured the cumulative sentence-wise attention of state-of-the-art abstractive summarization models using the ALTI [24] method and compared the positional distribution to Extractive Oracle on CNN \ Daily Mail and ArXiv datasets (Fig. 1). Both datasets are known to exhibit strong leading sentence bias and the models are expected to reflect it in their attention patterns. CNN / Daily Mail focuses on news summarization and the news articles are written in accordance with inverted pyramid scheme. This scheme dictates that the information in the text must be delivered in order to its importance, so the reader skimming through the introductory sentences would get the same idea of the described event as the one who reads the full text. Similarly, scientific articles from ArXiv usually base their abstracts on the introduction section of the article and thus the respective sentences have the highest saliency. However, contrary to extractive distribution, BART is weakly biased and Pegasus possess almost uniform attention distribution. Therefore, the benefits of extractive knowledge integration come from weak sentence-wise saliency intuition of pure abstractive summarization models.

3.2 Biasing attention with encoder mixture

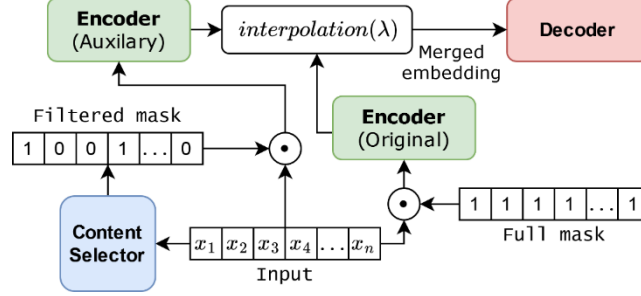


Fig. 2. Illustration of Biased encoder mixture

Correcting the attention distribution is the key to boosting the performance of fine-tune models. A straightforward approach is to utilize the existing binary attention masking mechanism to condition the model on the salient sentences. The disadvantage of hard masking is the total loss of secondary information as if the input text was filtered before passing to the abstractive summarization model. This method was tested in plug-and-play [17] and joint-training [11] scenarios and resulted in performance degradation even with additional pre-training.

Soft masking use the weight scheme to apply a relatively small penalty on selected positions to promote the remaining content. This approach has been tested separately on encoder self-attention [23] and decoder cross attention [17, 23, 12] with varying degree of success. The most successful variant [12] in terms of efficiency and consistency is to apply soft attention masking to decoder cross-attention of selected layers and attention heads. However, unlike other soft masking alternatives partial masking can't be used in zero-shot fashion as the optimal set of layers and attention heads differ from model to model and thus must be tuned on validation set.

We hypothesize that a similar result may be achieved by altering embeddings directly. The existing works applied embedding modulation [17, 16] to encoder part of encoder-decoder models and achieved a significant improvement in supervised setting. However, these methods can't be used without fine-tuning as they modify input format and alter the intermediate embeddings of encoder layers.

To adapt embedding modulation to plug-and-play scenario the points of modification must be limited to model agnostic checkpoint, the final output of the encoder. This approach has been used in text-to-image DALL-E-2 [10] model to merge the results of different prompts by interpolating the textual embeddings obtained by CLIP [25] encoder. Following this idea, we propose Biased encoder mixture (Fig. 2), which exploits binary attention masking and Siamese architecture to merge the generation results of the original and saliency-filtered variants of the input document. Merging allows to fill the information gaps created by binary masking and reinforces the signal of the salient tokens in the original model.

Biased encoder mixture (BEM) augments the encoder-decoder abstractive summarization model with auxiliary biased input encoder. We consider two auxiliary models,

the original encoder and identity encoder. The latter is obtained by additionally training the original encoder to output embeddings that would force the decoder to generate an identical copy of the input text. This supplements the saliency information with extraction signal that would boost the probability of copying the fragments from the source document. For interpolation we employ linear (lerp) and spherical (slerp) as they have different transformation trajectories [10]. The bias mask is obtained in two ways, statically from oracle sentence distribution (Fig. 1) and dynamically from sentence ranking predicted by extractive summarization model. In both cases the masking positions are obtained by sampling the top-p percentile of distribution.

4 Experiments

In this section we show the results of different attention modulation methods conditioned on extractive prior knowledge. To explain the effect on summary generation process we provide analysis of the methods in terms of content changes.

4.1 Experimental Setting

Table 1. Statistics of experimental datasets

Dataset	# examples in Train/Val/Test	# words		Extractiveness		
		Source	Target	Comp.	Coverage	Density
CNN / Daily Mail	287K / 13K / 11K	698.6	49.53	0.09	88%	3.77
Xsum	203K / 11K / 11K	383.17	21.74	0.1	64%	1.06
ArXiv	203K / 6K / 6K	5179.22	257.44	0.04	87%	3.94
SAMSum	14K / 818 / 819	97.23	21	0.25	68%	1.46

Datasets. For the experiments we chose four abstractive summarization datasets that were used in recent works [3, 5, 4]. CNN / Daily Mail [14] contains news articles and associated editor highlights acting as summaries from CNN and Daily Mail. Xsum [26] is a more abstractive counterpart with one-sentence summaries web scraped from BBC online news portal. ArXiv [27] is a collection of article-abstract pairs from ArXiv science article archive. SAMSum [28] is a human-annotated dialogue summarization dataset, which is created by asking linguists to create messenger-like conversations and then asking another group of linguists to write summaries. Dataset length statistics and extractive characteristics [29] are reported in Table 1.

Baselines. Besides current state-of-the-art abstractive summarization models we compare the performance of our methods with common extractive baselines and popular attention-based content control methods. As for abstractive (Original) we use BRIO (BART) [3] for CNN / Daily Mail, BRIO (Pegasus) [3] for Xsum, Pegasus [2] for ArXiv, BART [1] for SAMSum. For extractive we use Extractive Oracle [21] and

BERTSumExt [21] model (denoted as BERTEText). For attention manipulation we consider three methods. Standard binary attention masking of chosen subset of source sentences, Layer attention [12] which applies the binary mask to cross attention of selected decoder layer and Relevant Attention [13] which modulates the cross attention with query-source similarity matrix (in our case query is concatenated sentence subset).

Table 2. Optimal parameters for attention correction methods

Data	Masking	top-p	Layer attention	Relevance attention	BEM-id	BEM-original
CNN / Daily Mail	static	0.25	5	0.05	0.1 / slerp	0.1 / lerp
	dynamic		3	0.05	0.1 / slerp	0.2 / slerp
Xsum	static	0.2	2	0.01	0.1 / lerp	0.2 / lerp
	dynamic		1	0.01	0.1 / slerp	0.1 / slerp
ArXiv	static	0.26	4	0.01	0.1 / lerp	0.1 / lerp
	dynamic		3	0.03	0.1 / slerp	0.2 / slerp
SAMSum	static	0.32	5	0.03	0.1 / slerp	0.1 / slerp
	dynamic		3	0.03	0.1 / lerp	0.1 / slerp

Implementation details. Due to input length limitations of backbone abstractive summarization models, we truncate the input documents to the first 1024 tokens. To train identity encoders we freeze the decoders abstractive summarization models and train encoders on corresponding dataset on auto-encoding task (generate the original input). Dynamic mask is obtained from sentence saliency ranking predicted by BERTEText model. The top-p sampling percentile for both static and dynamic masking is determined by 90th percentile of reference-source sentence count ratio to ensure reference-source sentence alignment and accommodate for saliency ranking error. The interpolation method and λ coefficient for biased encoder mixtures is determined for each dataset independently with grid search (grid size for coefficient search is 0.1). Similarly, for Layer attention and Relevant attention we use grid search to find optimal masking layer and relevance modulation coefficient. The exact values can be found in Table 2.

4.2 Results

The results of attention correction experiments are reported in Table 3. Each data subsection is split into static masking and dynamic masking blocks, each showing the performance of Extractive Oracle, the original abstractive summarization model and the results of attention biasing methods.

As can be seen with static oracle distribution-based attention masking the improvements in the majority of cases is marginal. Binary masking being the strictest form of attention control only degrades the performance of the original model as it completely restricts the input information. A more precise approach such as layer attention is more stable yet improves only certain ROUGE aspects. Relevance attention has comparable performance however is less efficient on more abstractive datasets such as XSum. Biased encoder mixture is more consistent with performance improvements. An identity auxiliary encoder with attention masked input (BEM-id) is the most stable approach as it is the only one to show the improvements on SAMSum dataset. A simpler Siamese

BEM variant (BEM-original) better utilizes the attention guidance on BRIO contrastive-tuned models suggesting a correlation with encoder discrimination power.

Table 3. Results of attention correction experiments

Data	Model	Bias method	Static			Dynamic		
			R1	R2	RL	R1	R2	RL
CNN / Daily Mail	BERTExt	-	42.21	20.25	38.87			
		Oracle	53.88	32.73	49.97			
	BRIO (BART)	-	47.14	23.75	44.33			
		Binary	46.38	22.55	43.28	33.36	12.60	32.02
		Layer attention	47.68	23.70	44.98	48.69	24.56	45.46
		Relevance attention	47.22	23.81	44.31	47.50	24.06	45.02
		BEM-id	47.44	24.36	44.46	47.40	24.45	44.89
		BEM-original	47.63	24.21	44.77	49.08	25.57	46.18
Xsum	BERTExt	-	21.55	4.90	14.41			
		Oracle	27.52	7.59	21.46			
	BRIO (Pegasus)	-	47.30	24.46	39.29			
		Binary	45.90	22.82	37.45	34.77	14.12	27.48
		Layer attention	47.55	24.47	39.49	47.48	24.60	39.58
		Relevance attention	47.48	24.60	39.39	47.55	24.65	39.47
		BEM-id	47.40	24.43	39.24	47.70	24.95	39.41
		BEM-original	48.16	24.84	39.81	47.88	24.81	39.54
ArXiv	BERTExt	-	41.25	15.63	29.88			
		Oracle	53.24	23.31	32.42			
	Pegasus	-	43.44	16.22	25.66			
		Binary	42.54	14.85	24.90	42.64	14.95	24.64
		Layer attention	43.44	16.34	25.67	43.70	16.41	25.66
		Relevance attention	43.46	16.37	25.67	43.62	16.19	25.63
		BEM-id	44.18	16.54	25.61	44.64	16.86	25.81
		BEM-original	43.62	16.30	25.64	44.52	17.13	26.03
SAM-Sum	BERTExt	-	34.07	12.48	30.69			
		Oracle	45.21	17.58	39.55			
	BART	-	53.39	28.32	44.12			
		Binary	49.51	24.14	40.60	34.20	15.84	28.47
		Layer attention	53.37	28.26	44.00	54.57	29.33	45.30
		Relevance attention	53.11	27.99	43.65	53.65	28.21	44.03
		BEM-id	53.20	28.76	44.45	52.69	27.95	43.70
		BEM-original	53.29	28.15	44.18	53.93	28.61	44.52

With extractive summarization dynamic attention masking the quality improvements are much more noticeable. Layer attention stabilizes and show quality gains on all ROUGE aspects. Relevance attention on the other hand struggles to utilizes the improved guidance signal and shows similar scores. In contrast, BEM ranking reverses as BEM-original demonstrates the best overall performance. The Siamese variant achieves almost 8% and 6% relative ROUGE-2 improvement over the original model on CNN / Daily Mail and ArXiv respectively. At the same time the efficiency is much lower on datasets with higher abstractiveness of reference summaries as extractive strategies aren't optimal and thus the extractive summarizer can't properly estimate the alignment of source sentences with the reference summary.

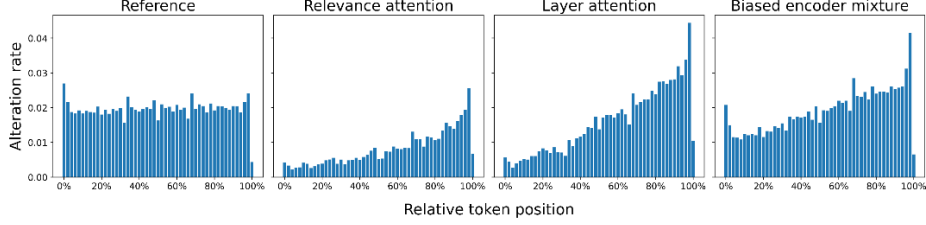


Fig. 3. Relative positions of the original generated summary altered by attention methods.

To better understand the effect of attention modulation we investigated the alteration patterns of the original summary on CNN / Daily Mail (Fig. 3). In practice, the original generation can disagree with the reference on any token, i.e., the attention modulation method should have a uniform token alteration pattern. However, Relevance attention tends to introduce the changes at the latter stages of generation, thus preserving the possible errors in the beginning of generated summary. While Layer attention shares this pattern it is more radical with corrections, overwriting the tokens as early as at 40% of generated length and guaranteeing an alternative ending. As can be seen the distinguishing trait of Biased encoder mixture is a more uniform alteration pattern with much slighter bias towards concluding tokens, meaning that the generation process may start diverging from the original even at the first token.

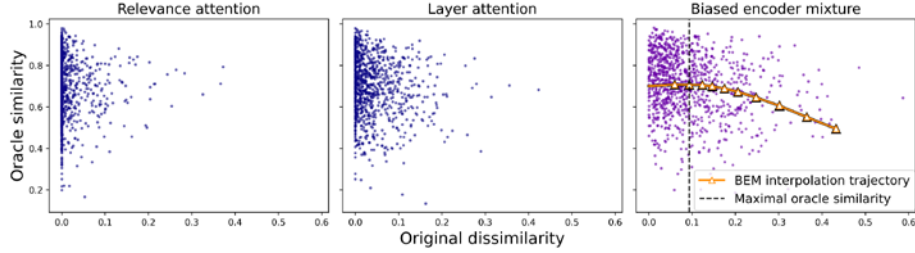


Fig. 4. Semantic distribution of summaries by attention correction methods

We additionally inspected the semantic characteristics of attention biased summaries with Sentence Transformers¹ (Fig. 4). Beside later summary generation changes Relevant attention is also conservative about their semantic nature keeping the majority of new summaries within 0.9 of cosine similarity with the original. Following the previous trend, Layer attention has a wider range of semantic changes and better utilizes the extractive signal to align with estimated Oracle yet still remains discreet. In contrast Biased encoder mixture seems to much less frequently preserve the semantic meaning of the summary diverging by 0.6 at extreme cases. The additional interpolation trajectory plot shows that Biased encoder mixture initially increases similarity with Oracle sentences as interpolation coefficient increases but past 0.2 value starts moving away from both Oracle and the original summaries suggesting a significant corruption of information flow.

¹ We used 'all-mpnet-base-v2' model available at <https://www.sbert.net>

5 Conclusion

In this paper we proposed Biased encoder mixture (BEM) a plug-and-play approach for content control in encoder-decoder abstractive summarization models based on Siamese networks and attention masking. We tested the approach on integration of extractive summarization knowledge in state-of-the-art pure abstractive summarization models. The evaluation on four popular datasets demonstrates that BEM is efficient even with static content selection strategies, outperforming the existing attention manipulation-based counterparts. Application of dynamic masking derived from extractive summarization further boost the quality of summaries bringing up to 7% ROUGE-2 relative improvements over the original model. Analysis of alteration patterns reveals that unlike other methods BEM produces much more diverse summaries having almost uniform positional editing distribution which aligns with positional error rate between the original generated summary and the reference. Semantic analysis shows that BEM is more likely to diverge from the original content, however the average magnitude of changes correlates with interpolation coefficient.

Acknowledgements. The work of Daniil Chernyshev (experiments, survey) was supported by Non-commercial Foundation for Support of Science and Education "INTELLECT". The work of Boris Dobrov (general concept, interpretation of results) was supported by the Russian Science Foundation (project 21-71-30003).

References

1. M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7871–7880, 2020.
2. J. Zhang, Y. Zhao, M. Saleh and P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," *Proceedings of the 37th International Conference on Machine Learning*, pp. 11328-11339, 2020.
3. Y. Liu, P. Liu, D. Radev and G. Neubig, "BRIO: Bringing Order to Abstractive Summarization," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2890-2903, 2022.
4. X. Zhang, Y. Liu, X. Wang, P. He, Y. Yu, S.-Q. Chen, W. Xiong and F. Wei, "Momentum Calibration for Text Generation," *arXiv:2212.04257*, 2022.
5. Y. Zhao, M. Khalman, R. Joshi, S. Narayan, M. Saleh and P. J. Liu, "Calibrating Sequence likelihood Improves Conditional Language Generation," *arXiv:2210.00045*, 2022.
6. Z.-Y. Dou, P. Liu, H. Hayashi, Z. Jiang and G. Neubig, "GSum: A General Framework for Guided Neural Abstractive Summarization," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4830-4842, 2021.

7. A. Fabbri, P. K. Choubey, J. Vig, C.-S. Wu and C. Xiong, "Improving Factual Consistency in Summarization with Compression-Based Post-Editing," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9149-9156, 2022.
8. M. Ravaut, S. Joty and N. Chen, "SummaReranker: A Multi-Task Mixture-of-Experts Re-ranking Framework for Abstractive Summarization," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4504-4524, 2022.
9. F. Wang, K. Song, H. Zhang, L. Jin, S. Cho, W. Yao, X. Wang, M. Chen and D. Yu, "Salience Allocation as Guidance for Abstractive Summarization," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6094-6106, 2022.
10. A. Ramesh, P. Dhariwal, A. Nichol, C. Chu and a. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," *arXiv:2204.06125*, 2022.
11. H. Li, A. Einolghozati, S. Iyer, B. Paranjape, Y. Mehdad, S. Gupta and M. Ghazvininejad, "EASE: Extractive-Abstractive Summarization End-to-End using the Information Bottleneck Principle," *Proceedings of the Third Workshop on New Frontiers in Summarization*, pp. 85-95, 2021.
12. S. Cao and L. Wang, "Attention Head Masking for Inference Time Content Selection in Abstractive Summarization," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5008-5016, 2021.
13. W. Xiao, L. Miculicich, Y. Liu, P. He and G. Carenini, "Attend to the Right Context: A Plug-and-Play Module for Content-Controllable Summarization," *arXiv:2212.10819*, 2022.
14. A. See, P. J. Liu and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, p. 1073-1083, 2017.
15. S. Gehrmann, Y. Deng and A. Rush, "Bottom-Up Abstractive Summarization," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 4098-4109, 2018.
16. L. Lebanoff, F. Dernoncourt, D. S. Kim, W. Chang and F. Liu, "A Cascade Approach to Neural Abstractive Summarization with Content Selection and Fusion," *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*, pp. 529-535, 2020.
17. J. Ji, Y. Kim, J. Glass and T. He, "Controlling the Focus of Pretrained Language Generation Models," *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3291-3306, 2022.
18. H. Jing and K. R. McKeown, "The Decomposition of Human-Written Summary Sentences," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 129-136, 1999.
19. Y. Suhara, X. Wang, S. Angelidis and W.-C. Tan, "OpinionDigest: A Simple Framework for Opinion Summarization," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5789-5798, 2020.

20. H. Zhang, X. Liu and J. Zhang, "SummIt: Iterative Text Summarization via ChatGPT," *arXiv:2305.14835*, 2023.
21. Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, p. 3730–3740, 2019.
22. S. Zhang, D. Wan and M. Bansal, "Extractive is not Faithful: An Investigation of Broad Unfaithfulness Problems in Extractive Summarization," *arXiv:2209.03549*, 2022.
23. S. Cao and L. Wang, "HIBRIDS: Attention with Hierarchical Biases for Structure-aware Long Document Summarization," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 786–807, 2022.
24. J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano and M. R. Costa-jussà, "Towards Opening the Black Box of Neural Machine Translation: Source and Target Interpretations of the Transformer," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 8756–8769, 2022.
25. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," *arXiv:2103.00020*, 2021.
26. S. Narayan, S. B. Cohen and M. Lapata, "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, 2018.
27. J. Xu, Z. Gan, Y. Cheng and J. Liu, "Discourse-Aware Neural Extractive Model for Text Summarization," *arXiv:1910.14142*, 2019.
28. B. Gliwa, I. Mochol, M. Biesek and A. Wawer, "SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization," *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, 2019.
29. M. Grusky, M. Naaman and Y. Artzi, "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pp. 708–719, 2018.

6 Appendix

Table 3. Results with static oracle distribution masking

Data	Model	Bias method	Coef	R1	R2	RL
CNN / Daily Mail	Oracle	-	-	53.88	32.73	49.97
	BRIO (BART)	-	-	47.14	23.75	44.33
		Binary	-	46.38	22.55	43.28
		Layer attention (layer 5)	-	47.68	23.70	44.98
		Relevance attention	0.05	47.22	23.81	44.31
		BEM-id-pure (slerp)	0.9	46.95	24.51	44.12
		BEM-id (slerp)	0.9	47.44	24.36	44.46
		BEM-original (lerp)	0.9	47.63	24.21	44.77
Xsum	Oracle	-	-	27.52	7.59	21.46
	BRIO (Pegasus)	-	-	47.30	24.46	39.29
		Binary	-	45.90	22.82	37.45
		Layer attention (layer 2)	-	47.55	24.47	39.49
		Relevance attention	0.01	47.48	24.60	39.39
		BEM-id-pure (lerp)	0.9	47.41	24.34	39.39
		BEM-id (lerp)	0.9	47.40	24.43	39.24
		BEM-original (lerp)	0.8	48.16	24.84	39.81
ArXiv	Oracle	-	-	53.24	23.31	32.42
	Pegasus	-	-	43.44	16.22	25.66
		Binary	-	42.54	14.85	24.90
		Layer attention (layer 4)	-	43.44	16.34	25.67
		Relevance attention	0.01	43.46	16.37	25.67
		BEM-id-pure (lerp)	0.9	44.07	16.51	25.75
		BEM-id (lerp)	0.9	44.18	16.54	25.61
		BEM-original (lerp)	0.9	43.62	16.30	25.64
SAM-Sum	Oracle	-	-	45.21	17.58	39.55
	BART	-	-	53.39	28.32	44.12
		Binary	-	49.51	24.14	40.60
		Layer attention (layer 5)	-	53.37	28.26	44.00
		Relevance attention	0.03	53.11	27.99	43.65
		BEM-id-pure (lerp)	0.9	52.84	27.94	44.12
		BEM-id (slerp)	0.9	53.20	28.76	44.45
		BEM-original (slerp)	0.9	53.29	28.15	44.18

Table 4. Results with dynamic extractive summarization masking

Data	Model	Bias method	Coef	R1	R2	RL
CNN / Daily Mail	BERTEExt	-	-	42.21	20.25	38.87
	BRIO (BART)	-	-	47.14	23.75	44.33
		Binary	-	33.36	12.60	32.02
		Layer attention (layer 3)	-	48.69	24.56	45.46
		Relevance attention	0.05	47.50	24.06	45.02
		BEM-id (slerp)	0.9	47.40	24.45	44.89
		BEM-original (slerp)	0.8	49.08	25.57	46.18
Xsum	BERTEExt	-	-	21.55	4.90	14.41
	BRIO (Pegasus)	-	-	47.30	24.46	39.29
		Binary	-	34.77	14.12	27.48
		Layer attention (layer 1)	-	47.48	24.60	39.58

		Relevance attention	0.01	47.55	24.65	39.47
		BEM-id (slerp)	0.9	47.70	24.95	39.41
		BEM-original (slerp)	0.9	47.88	24.81	39.54
ArXiv	BERTEExt	-	-	41.25	15.63	29.88
	Pegasus	-	-	43.44	16.22	25.66
		Binary	-	42.64	14.95	24.64
		Layer attention (layer 3)	-	43.70	16.41	25.66
		Relevance attention	0.03	43.62	16.19	25.63
		BEM-id (slerp)	0.9	44.64	16.86	25.81
		BEM-original (slerp)	0.8	44.52	17.13	26.03
SAM-Sum	BERTEExt	-	-	34.07	12.48	30.69
	BART	-	-	53.39	28.32	44.12
		Binary	-	34.20	15.84	28.47
		Layer attention (layer 3)	-	54.57	29.33	45.30
		Relevance attention	0.03	53.65	28.21	44.03
		BEM-id (lerp)	0.9	52.69	27.95	43.70
		BEM-original (slerp)	0.9	53.93	28.61	44.52