Improving Faithfulness in Abstractive Summarisation Using Attributions

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown impressive performance in generating concise and fluent summaries. However, the generated summaries can still contain information that is inconsistent with the input article, which is known as faithful hallucination. This paper proposes a simple and effective approach to improve faithfulness in abstractive summarisation by leveraging attribution at inference time. Our method incorporates attribution mechanism to explicitly identify the most influential input sentences that contribute to the generated summary and steers the model to refine the summary based on these attributed sentences. We evaluate our approach on multiple summarisation benchmarks, including CNN/DailyMail, XSum, and CCSum, measuring both faithfulness and similarity to the reference. Our experiment results show that attribution-guided summarisation consistently reduces faithfulness hallucination compared with several decoding-based approaches, while maintaining comparable semantic similarity to the reference.

1 Introduction

Improving faithfulness in summarisation is essential to improving user trust and avoiding the spread of misinformation, especially in high-stakes domains such as news and healthcare. A *faithful* summary should accurately reflect the information provided in the input document. Despite recent advances in abstractive summarisation by LLMs, the generated summaries remain prone to hallucinations and factual inconsistencies, i.e., the summaries can include information that is fabricated or unsupported by the input document (Huang et al., 2025; Subbiah et al., 2024). While LLMs can identify salient information in the input document, they do not always condition the generation process on the provided context. Instead, they may rely on parametric knowledge acquired during pretraining (Longpre et al., 2021; Wang et al., 2023; Xu et al., 2024b; Li et al., 2025) or exhibit positional bias (Liu et al., 2024b; Ravaut et al., 2024), resulting in contextually inconsistent output.

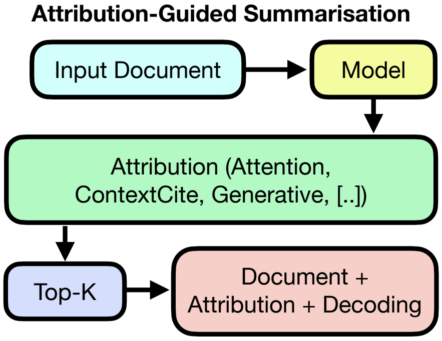
Attribution-Guided Summarisation
Input Document
Model
Attribution (Attention, ContextCite, Generative, [..])
Top-K
Document + Attribution + Decoding

Figure 1: Illustration of attribution-guided summarisation pipeline.

In this paper, we aim to mitigate faithfulness hallucinations in summarisation by eliciting influential information with *attributions* and constraining the generation process with this information. Attributions identify which parts of the input document most strongly influence or contribute to specific elements of a model's generated output, creating a traceable connection between source information and summary content. Model attribution may not align perfectly with human judgment, but it reveals what information the model considers important during generation and provides useful insights into the decision-making process of the model. We use attribution to trace the relevant evidence that contributes to the generated response and recalibrate the model's attention to the original context. Specifically, we are interested in the following research questions: i) Which attribution method is most effective in selecting influential information from the input document? ii) How can we leverage these attribution signals to improve the faithfulness of generated summaries?

To answer these questions, we investigate the effectiveness of different attribution methods for identifying influential information, including attention weights (Bahdanau et al., 2015), ContextCite (Cohen-Wang et al., 2024), and generative attribution (Wright et al., 2025; Zhang et al., 2023; Li et al., 2023a). We extract attribution at the sentence level and propose a simple yet effective approach to encourage the model to focus on the attributed sentences during generation. Unlike prior work that primarily focuses on complex frameworks that require additional training or non-trivial decoding methods, our approach uniquely leverages sentence-level attributions with an effective prompting strategy that can be applied to any existing LLMs without additional training, making it both more practical and adaptable across different domains and models.

We evaluate the performance of our attribution-guided summarisation approach on XSum, CCSum and CNN/Daily Mail datasets with Llama3.1-8b and Mistral-7b model. Compared with several decoding-based approaches for improving faithfulness, our framework consistently improves the faithfulness of generated summaries while maintaining a reasonable level of semantic similarity to the reference.

2 Background

Our approach relies on a two-step pipeline, where we i) identify influential sentences in the input using attribution methods (Section 2.1), and ii) improve the faithfulness of the model w.r.t. such parts of the input using prompting or contrastive decoding (Section 2.2).

2.1 Attribution Methods

Attribution involves identifying the input segments that contribute most to the generated texts produced by the model. In our work, we focus on attribution methods solely driven by the model's internal mechanisms, without retrieving from input sources or training a separate attribution model based on human-annotated citations. We aim to explore whether the model attribution can encourage faithful generation in an unsupervised and cost-efficient manner.

Attention-based attribution Attention-based attribution (Clark et al., 2019) uses attention weights to identify which input tokens the model is looking at while generating each output token. We consider the attention-based attribution of input context in sentence level. Given each output token, we measure the importance of each input token by the average of the attention weights across all layers and all attention heads. We compute the average of the attention weights for each input sentence and define the attribution score of each sentence as the maximum attention weight across all output tokens.

Perturbation-based Attributions Perturbation-based methods are attribution techniques that quantify input importance by measuring changes in model outputs when inputs are systematically modified or perturbed, such as through occlusion (Zeiler and Fergus, 2014; Ribeiro et al., 2016; Mohebbi et al., 2023; Zhao and Shan, 2024; Cohen-Wang et al., 2024). In this work, we use the recently proposed ContextCite (Cohen-Wang et al., 2024), a perturbation-based attribution method that identifies which parts of the input context most causally influence a model's generation by systematically ablating context elements and measuring changes in output probabilities.

Generative attribution These methods use the LLM itself to directly generate attributions. We prompt the LLM to extract influential sentences from the input document and generate a summary based on these sentences. These extracted sentences are treated as generative attribution.

2.2 Contrastive Decoding

Contrastive Decoding (CD) is a text generation technique that produces high-quality continuations by maximising the difference between log-probabilities of an *expert* language model and a smaller *amateur* model (Li et al., 2023b; Liu et al., 2021). CD has been used for a number of applications, ranging from increasing the faithfulness of generated text to a given document (Shi et al., 2024; Malkin et al., 2022) to model alignment (Liu et al., 2024a). In this work, we use the recently proposed Context-Aware Decoding (CAD; Shi et al., 2024), a method for improving the faithfulness of a model to a given context. CAD adjusts the output distribution by contrasting the output probabilities when the model is provided with and without the context, promoting generated tokens that are more relevant to the context. More formally, let $x_{<t}$ be a sequence of previous tokens, and let $p_{\text{LM}}\left(x_t \mid x_{<t}\right) \propto \exp\left[f_{\text{LM}}(x_{<t})\right]$ denote the next-token distribution defined by a model f_{LM}. Given a context c we want the model to be faithful to, CAD

defines the next-token distribution as:

$$p_{\mathrm{LM}}\left(x_t \mid x_{<t}, c\right) \propto \exp\left[(1+\alpha) f_{\mathrm{LM}}(x_{<t}, c) - \alpha f_{\mathrm{LM}}(x_{<t})\right], \quad (1)$$

where $\alpha \in \mathbb{R}_+$ is a user-specified hyperparameter used to increase the likelihood of tokens selected by the model conditioned on the context c.

3 Method

In this section, we describe our attribution-guided summarisation framework. Given an input document $D = \{x_1, x_2, \ldots, x_n\}$ with n sentences and a pre-trained summarization model M, our goal is to extract a set of sentences from D that contribute most to the initial predictions, and then to steer M to focus on those influential sentences when producing the refined summary. Fig. 1 gives an overview of our pipeline.

We propose to encourage the LLM to focus on the input context by incorporating attribution at inference time with either prompting or contrastive decoding. Post-hoc attribution and generative attribution are two different ways to extract influential information from the context. With post-hoc attribution methods (e.g., attention weights or ContextCite attribution), we extract the k most influential sentences according to the attribution scores after generating the initial summary, and append those sentences to the input context to refine the summary. For generative attribution, we use LLM itself to extract attributed sentences. Our prompt templates for attribution-guided summarisation experiments and for extracting generative attribution can be found in Appendix B

We also develop a strategy to guide the generation by attribution using contrastive decoding so that the model can pay more attention to the attributed sentences. Specifically, we adjust the output distribution by applying a contrastive function between the logits when the model is provided the document and the logits when the model is provided both the input document and attributed sentences. According to Eq. (1), we treat the attributed sentences as context c. $f_{\mathrm{LM}}(x_{<t}, c)$ refers to the setting when the model has access to both the input document and attributed sentences, while $f_{\mathrm{LM}}(x_{<t})$ refers to the output distribution when the model is only provided the document,.

4 Experiments

Datasets We evaluate the performance of LLMs guided by different attributions on three abstractive news summarisation datasets, including XSum (Narayan et al., 2018), CCSum (Jiang and Dreyer, 2024), and CNN/Daily Mail (Nallapati et al., 2016).

Table 1: Performance of Llama3.1-8b and Mistral-7b model with attribution-guided approaches on news summarisation datasets, in comparison with decoding-based baselines.

Model	ROUGE-L ↑	BERT-F1 ↑	Summa-C ↑	FactScore ↑
		XSum		
Llama3.1-8b	**20.62**	68.29	23.37	85.81
+ DoLA	20.52	**68.70**	23.25	85.30
+ CAD	19.57	68.27	23.27	86.67
+ Attribution	19.09	67.52	23.35	**89.23**
+ Attribution+CD	18.61	67.06	**23.47**	86.32
Mistral-7b	16.30	65.25	23.48	84.66
+ DoLA	16.22	65.41	23.86	86.53
+ CAD	15.65	64.86	23.63	84.98
+ Attribution	**17.61**	**66.19**	**30.41**	**87.57**
+ Attribution+CD	15.91	65.24	23.93	81.47
		CCSum		
Llama3.1-8b	33.95	72.41	29.41	97.69
+ DoLA	32.71	72.21	28.23	97.50
+ CAD	32.83	72.12	29.05	97.32
+ Attribution	**34.26**	**72.47**	**32.16**	**97.96**
+ Attribution+CD	33.53	72.42	31.68	96.56
Mistral-7b	31.05	71.35	29.44	96.36
+ DoLA	30.85	71.66	27.74	96.20
+ CAD	29.78	70.25	29.00	95.64
+ Attribution	32.36	72.19	31.68	**96.44**
+ Attribution+CD	**32.74**	**72.47**	**33.69**	95.93
		CNN/DM		
Llama3.1-8b	**25.48**	**62.05**	35.81	**92.20**
+ DoLA	24.58	61.75	35.65	88.15
+ CAD	25.23	61.56	36.89	87.86
+ Attribution	24.77	60.53	40.88	88.58
+ Attribution+CD	23.18	58.31	**41.13**	83.88
Mistral-7b	25.83	**62.47**	35.22	95.56
+ DoLA	24.55	61.80	36.56	94.83
+ CAD	**25.99**	62.38	36.74	95.47
+ Attribution	25.62	62.34	**37.13**	**95.62**
+ Attribution+CD	24.75	61.59	36.85	93.18

Baselines As a baseline, we use the LLM to directly generate the summary given the input document, without providing any attributed sentences. We also compare our attribution-guided approach with CAD and DoLA decoding, which effectively improve faithfulness in summarisation (Gema et al., 2024).

Evaluation Metrics We use ROUGE-L (Lin, 2004) and BERTScore-F1 (Zhang et al., 2020) to measure the similarity between the generated summary and the reference summary. To evaluate the faithfulness of generated summaries, we adopt Summa-C (Laban et al., 2022) and FactScore (Min et al., 2023) to measure how well the information in the generated summary is grounded by the input document.

Ablation study For attribution-based prompting, we experiment with appending the attributed sentences before or after the input document. For

Table 2: Performance of different attribution methods with Llama3.1-8b-Instruct model on XSum dataset.

Method	ROUGE-L ↑	BERT-F1 ↑	Summa-C ↑	FactScore ↑
Baseline (N/A)				
Baseline	20.22	68.29	23.31	87.18
Generative				
+ Attribution	18.79	67.31	23.22	**88.51**
+ Attribution + CD	18.29	67.17	23.48	87.29
+ Attribution (prefix)	18.64	67.00	23.13	88.13
+ Attribution + CD (with mask)	**19.10**	**67.95**	**23.49**	86.20
Attribution only	17.62	66.47	23.33	88.38
ContextCite				
+ Attribution	18.77	67.41	23.17	87.79
+ Attribution + CD	18.33	67.21	23.62	87.27
+ Attribution (prefix)	**19.08**	67.52	23.23	88.67
+ Attribution + CD (with mask)	**19.08**	**67.77**	23.29	87.08
Attribution only	17.58	66.79	**23.64**	**89.14**
Attention				
+ Attribution	18.84	67.34	23.18	88.47
+ Attribution + CD	18.62	67.06	23.47	86.32
+ Attribution (prefix)	19.11	67.48	23.10	**89.23**
+ Attribution + CD (with mask)	**19.31**	**68.14**	23.36	86.72
Attribution only	17.44	66.71	**23.74**	87.83

attribution-guided contrastive decoding, we propose to mask the attributed sentences in the input. Experiment details are available in Appendix A.

4.1 Discussion

Table 1 shows the performance of the attribution-guided summarisation approach compared to several baselines on three summarisation datasets. With each model, we report the performance of two variants of our methods: attribution-based prompting (Attribution) and attribution-guided contrastive decoding (Attribution+CD). On CCSum dataset, incorporating attributed sentences during generation consistently improves both ROUGE-L and faithfulness metrics for both Llama3.1-8b and Mistral-7b model. On the XSum and CNN/DM datasets, our approaches also enhance Summa-C and FactScore while maintaining ROUGE-L and BERTScore-F1 scores at levels comparable to the baselines. This observation suggests that while attribution signals can encourage the model to produce output faithful to the context, lexical alignment with the reference summary may be reduced.

To investigate which attribution method is most effective in identifying influential sentences, we summarise the performance of attention-based attribution, ContextCite and generative attribution with Llama3.1-8b model on the XSum dataset in Table 2. Overall, attention-based attribution achieves the best balance in terms of both ROUGE-L and faithfulness metrics. When appending these attributed sentences before the input document, the model achieves a FactScore of 89.23 while preserving reasonable ROUGE-L and BertScore-F1 scores.

5 Related Work

Faithful Summarisation The issue of faithfulness hallucinations in LLM summarisation is well-known — e.g., Pagnoni et al. (2021) found that over 60% of summaries from state-of-the-art models contain hallucinations. Broadly, prior work falls into two paradigms: i) *fine-tuning-based approaches*, which modify training objectives or data to encourage actuality (e.g., (Feng et al., 2024)), and ii) *training-free* (inference-time) techniques, which intervene during decoding or post-processing to mitigate hallucinations without retraining the model (e.g., (Xu et al., 2024a; Wan et al., 2023; Li et al., 2024)). We focus on improving faithfulness via inference-time strategies and propose to guide the generation using attributions.

Attributed Text Generation Another line of research to improve factuality and reliability focuses on generating text with citations of supporting evidence. Recent studies have proposed approaches for guiding generation by attribution, with varying levels of granularity. (Gao et al., 2023) enable the model to generate text with citations to retrieved passages. (Slobodkin et al., 2024) propose to identify fine-grained sentence-level attribution and then generate text conditioned on the relevant segments. Attribution methods used in these work aim to find evidence that supports statements generated by the model, while we focus on identifying which input segments contribute most to the generated output and using them to steer the model's behavior.

Hallucination Mitigation Several approaches have been proposed to mitigate hallucinations in the generated output. Contrastive decoding (Li et al., 2023b; Chuang et al., 2024; Shi et al., 2024) can reduce hallucinations by intervening the output distribution. Our work further explores how to combine contrastive decoding with attribution to mitigate hallucinations in summarisation.

6 Conclusions

We propose attribution-guided summarisation framework to mitigate faithfulness hallucinations. Our framework leverages attribution methods to extract a set of influential sentences that contribute to the model's initial prediction and then steers the model to focus on these sentences to produce a refined summary. Experiment results on news summarisation datasets demonstrate our framework can consistently improve the faithfulness of the summaries without sacrificing the overall quality.

Limitations

Although our attribution-guided summarisation framework can help improve the faithfulness of generated summaries, it has a few limitations. Extracting attributed sentences from the input context can consume a significant amount of GPU memory depending on the specific attribution method, especially when processing long-context documents. Our current experiments are limited to news summarisation datasets with short input documents. An important direction for future work is to explore how to extend the framework to address long context problems.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In *ACL (Poster and Demonstration)*. ACL.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *ICLR*. OpenReview.net.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. In *BlackboxNLP@ACL*, pages 276–286. Association for Computational Linguistics.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. Contextcite: Attributing model generation to context. In *NeurIPS*.

Huawen Feng, Yan Fan, Xiong Liu, Ting-En Lin, Zekun Yao, Yuchuan Wu, Fei Huang, Yongbin Li, and Qianli Ma. 2024. Improving factual consistency of news summarization by contrastive preference optimization. In *EMNLP (Findings)*, pages 11084–11100. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *CoRR*, abs/2305.14627.

Aryo Pradipta Gema, Chen Jin, Ahmed Abdulaal, Tom Diethe, Philip Teare, Beatrice Alex, Pasquale Minervini, and Amrutha Saseendran. 2024. Decore: Decoding by contrasting retrieval heads to mitigate hallucinations. *CoRR*, abs/2410.18860.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55.

Xiang Jiang and Markus Dreyer. 2024. Ccsum: A large-scale and high-quality dataset for abstractive news summarization. In *NAACL-HLT*, pages 7306–7336. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Trans. Assoc. Comput. Linguistics*, 10:163–177.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023a. A survey of large language models attribution. *CoRR*, abs/2311.03731.

Gaotang Li, Yuzhong Chen, and Hanghang Tong. 2025. Taming knowledge conflicts in language models. *CoRR*, abs/2503.10996.

Taiji Li, Zhi Li, and Yin Zhang. 2024. Improving faithfulness of large language models in summarization via sliding generation and self-consistency. In *LREC/COLING*, pages 8804–8817. ELRA and ICCL.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *ACL (1)*, pages 12286–12312. 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.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. *CoRR*, abs/2401.08565.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL/IJCNLP (1)*, pages 6691–6706. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *EMNLP (1)*, pages 7052–7063. Association for Computational Linguistics.

Louis Mahon and Mirella Lapata. 2024. A modular approach for multimodal summarization of TV shows. In *ACL (1)*, pages 8272–8291. Association for Computational Linguistics.

Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2022. Coherence boosting: When your pretrained language model is not paying enough attention. In *ACL (1)*, pages 8214–8236. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*, pages 12076–12100. Association for Computational Linguistics.

Hosein Mohebbi, Willem H. Zuidema, Grzegorz Chrupala, and Afra Alishahi. 2023. Quantifying context mixing in transformers. In *EACL*, pages 3370–3392. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290. ACL.

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 *EMNLP*, pages 1797–1807. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *NAACL-HLT*, pages 4812–4829. Association for Computational Linguistics.

Mathieu Ravaut, Aixin Sun, Nancy F. Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models. In *ACL (1)*, pages 2764–2781. Association for Computational Linguistics.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *HLT-NAACL Demos*, pages 97–101. The Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *NAACL (Short Papers)*, pages 783–791. Association for Computational Linguistics.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. *CoRR*, abs/2403.17104.

Melanie Subbiah, Faisal Ladhak, Akankshya Mishra, Griffin Adams, Lydia B. Chilton, and Kathleen R. McKeown. 2024. STORYSUMM: evaluating faithfulness in story summarization. *CoRR*, abs/2407.06501.

David Wan, Mengwen Liu, Kathleen R. McKeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-aware decoding strategies for abstractive summarization. In *EACL*, pages 2856–2872. Association for Computational Linguistics.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Resolving knowledge conflicts in large language models. *CoRR*, abs/2310.00935.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Dustin Wright, Zain Muhammad Mujahid, Lu Wang, Isabelle Augenstein, and David Jurgens. 2025. Unstructured evidence attribution for long context query focused summarization. *CoRR*, abs/2502.14409.

Lei Xu, Mohammed Asad Karim, Saket Dingliwal, and Aparna Elangovan. 2024a. Salient information prompting to steer content in prompt-based abstractive summarization. *CoRR*, abs/2410.02741.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024b. Knowledge conflicts for llms: A survey. In *EMNLP*, pages 8541–8565. Association for Computational Linguistics.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *ECCV (1)*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Extractive summarization via chatgpt for faithful summary generation. In *EMNLP (Findings)*, pages 3270–3278. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *ICLR*. OpenReview.net.

Zhixue Zhao and Boxuan Shan. 2024. Reagent: A model-agnostic feature attribution method for generative language models. *CoRR*, abs/2402.00794.

A Experimental Settings

A.1 Datasets

We use the official version of XSum and CNN/-Daily Mail datasets from Huggingface Hub (Wolf et al., 2019). CCSum is available under the CC-BY-NC-4.0 License and we contacted the authors to obtain the processed data. We evaluate the models on 1,000 random test samples for each summarisation dataset.

A.2 Attribution

To extract the attributed sentences, we use NLTK sentence tokenizer (Bird and Loper, 2004) to split the input document into sentences so that we can aggregate the attribution scores in sentence level. With each attribution method, we extract three salient sentences with the highest attribution scores. We apply the same prompt templates and generation hyperparameters for both attribution extraction and summary generation. **Attention weight attribution** For attention-based attribtution, we compute the average of attention weights across all layers and all attention heads. Given each output token, we aggregate the attention weights of all input tokens and compute the average for each input sentence. The attribution score of each input sentence is defined as the maximum attention weight across all output tokens. **ContextCite attribution** We use ContextCite library (Cohen-Wang et al., 2024) to extract ContextCite attribution. Specifically, we treat the input document as the context and ensure the prompt template is the same as we used in other experiments.

Generative attribution We prompt LLMs to first extract three important sentences from the input document and then generate a summary based on the extracted information. We use greedy decoding to generate the summaries together with attribution. We set temperature to 0 and set max new tokens to 1024. The attribution score of each extracted sentence is set to 1.0.

A.3 Baselines

For context-aware decoding baseline, we use prompt template 4 and contrast with the setting in which the model is only provided the prompt instructions. For DoLA baseline, we use the Huggingface implementation and contrast the last layer with the earlier layers by setting dola_layers to low.

A.4 Hyperparameters

We use greedy decoding to generate the summaries. We set the temperature to 0 and set max new tokens to 128 for all the experiments on three summarisation datasets. When applying contrastive decoding during generation, we set the contrastive weight α to 0.5 since it shows the best performance on XSum test data based on a grid search between -0.5 and 2.

A.5 Evaluation

We implement ROUGE-L and BERTScore using the Huggingface evaluate library. BERTScore is computed by DeBERTa-xlarge-mnli model (He et al., 2021) and Summa-C is computed by SummaCConv model. We adopt the implementation of FactScore from PRISMA code base (Mahon and Lapata, 2024) and use GPT-4o-mini for both atomic fact extraction and claim verification when computing FactScore.

B Prompt Templates

We use different instructions for generating summaries on different datasets, as shown in Figure 2, 3 and 4. These instructions are also adopted for extracting generative attribution and generating the refined summaries on different datasets. Example prompt templates for CNN/Daily Mail experiments are illustrated in Figure 5 and 6.

XSum

Summarise the document below in one sentence:
<doc>

Figure 2: Prompt template for generating summaries on XSum.

CCSum

Summarise the document below in one sentence or two sentences:
<doc>

Figure 3: Prompt template for generating summries on CCSum.

CNN/Daily Mail

Summarise the document below:
<doc>

Figure 4: Prompt template for generating summaries on CNN/Daily Mail.

Attribution-guided Summarisation

Summarise the document below:
<doc>
You should pay attention to the following main points:
1. Attributed sentence
2. Attributed sentence
...

Figure 5: Prompt template for attribution-guided summarisation on CNN/Daily Mail.

Generative Attribution

Extract a list of K sentences from the input document and then generate a summary only based on the extracted facts:
<doc>
Here is the output format.
Key Sentences:
1. sentence1, 2. sentence2, ...
Summary:
summary

Figure 6: Prompt template for extracting generative attribution on CNN/Daily Mail.

C Computation Details

The experiments were run on two NVIDIA A100 GPUs with 80GB of GPU memory. The GPU hours vary depending on the experiment setting. Generating summaries for 1,000 test samples on CCSum without applying contrastive decoding takes about 1.5 hours, while it takes approximately 5 hours when contrastive decoding is used.