Mitigating Hallucination in Abstractive Summarization with Domain-Conditional Mutual Information

Anonymous ACL submission

Abstract

A primary challenge in abstractive summarization is hallucination—the phenomenon where a model generates plausible text that is absent in the source text. We hypothesize that the domain (or topic) of the source text triggers the model to generate text that is highly probable in the domain, neglecting the details of the source text. To alleviate this model bias, we introduce a decoding strategy based on domain-conditional pointwise mutual information. This strategy adjusts the generation probability of each token by comparing it with the token's marginal probability within the domain of the source text. According to evaluation on the XSUM dataset, our method demonstrates improvement in terms of faithfulness and source relevance.

Method	Text
Source	...chairman of the Scottish Chambers of Commerce economic advisory group, said: "Our latest economic data shows that many Scottish businesses will have a successful 2017...
CPMI	The Scottish Chambers of Commerce has issued a warning about the outlook for the economy in 2017.
PMI_{DC}	The Scottish Chambers of Commerce has said it expects the economy to have a "successful" year in 2017.
Domain	Economy, Businesses, GDP

Table 1: An example of hallucination in abstractive summarization. Inconsistent words are highlighted in *red* fonts, and consistent words are highligthed in *blue* fonts.

1 Introduction

Abstractive summarization is the task of generating a summary by interpreting and rewriting a source text. State-of-the-art pre-trained language models have achieved remarkable performance in this task (Lewis et al., 2019; Zhang et al., 2020). However, upon closer examination, a common issue emerges: hallucination between the source document and the generated text. Prior studies have made efforts to enhance the faithfulness of the summary to the source text, yet hallucination remains a persistent challenge (Maynez et al., 2020; Mao et al., 2021; Zhu et al., 2021; Zhang et al., 2023).

To solve this issue, we introduce a decoding strategy based on domain-conditional pointwise mutual information (PMI_{DC}) (Section 3). The motivation for PMI_{DC} is that the domain of the source text provokes the model to generate text that is highly probable in the source domain, leading to plausible but factually inconsistent text. Building on this motivation, PMI_{DC} computes how much more likely a token becomes in the summary when conditioned on the input source text, compared to when the token is conditioned only on the domain of the source text. This effectively penalizes the model's tendency to fall back to domain-associated words when the model has high uncertainty about the generated token.

This idea was inspired by conditional pointwise mutual information (CPMI) (van der Poel et al., 2022), which similarly penalizes a token's marginal probability. But CPMI does not capture the important fact that a token's probability depends highly on the source domain in summarization. For example, consider the example presented in Table 1. The source text states, "Our latest economic data shows that many Scottish businesses will have a successful 2017". CPMI undesirably introduces the term "warning", which frequently appears in the domain of economy in the training data, generating information that contradicts the source text. By contrast, PMI_{DC} lowers the probability of the term "warning" by capturing the high conditional likelihood of this term given the domain and avoids the hallucination.

We use automated metrics for evaluation on the challenging XSUM dataset (Narayan et al., 2018) achieving significant improvements in faithfulness and relevance to source texts according to metrics like AlignScore, FactCC, BARTScore, and BSFact, with only a marginal decrease in ROUGE and BertScore. This highlights the effectiveness and robustness of PMI_{DC} in abstractive summarization.

2 Preliminaries

Problem setting We adopt the problem definition in van der Poel et al. (2022). In abstractive summarization, an input source text, denoted as $\mathbf{x} \in \mathcal{X}$, is condensed into an output string represented by $\mathbf{y} = \langle y_0, \dots, y_T \rangle \in \mathcal{Y}$. This output string is a sequence of tokens from the vocabulary $\mathcal{V}$. Each sequence begins with token y_0 and ends with y_T, and the length of the output is $T + 1$. The optimal $\mathbf{y}$ that belongs to a valid string set $\mathcal{Y}$ is obtained via a scoring function as follows:

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \ \text{score}(\mathbf{y}|\mathbf{x}).$$

Utilizing beam search is a practical solution for searching possible strings. The typical beam search with an autoregressive generation model uses the following scoring function:

$$\text{score}(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^{T} \text{score}(y_t|\mathbf{x}, \mathbf{y}_{<t}) \tag{1}$$

where $\text{score}(y_t|\mathbf{x}, \mathbf{y}_{<t}) = \log p(y_t|\mathbf{x}, \mathbf{y}_{<t})$ is a token-level log probability computed by the model.

Pointwise Mutual Information PMI scoring utilizes mutual information between the input and output. This penalizes the generation of tokens that are marginally likely but not related to the input. The formula for PMI scoring can be expressed as follows:

$$\begin{aligned} \text{score}(y_t|\mathbf{x}, \mathbf{y}_{<t}) = &\log p(y_t|\mathbf{x}, \mathbf{y}_{<t}) \\ &- \log p(y_t|\mathbf{y}_{<t}) \end{aligned} \tag{2}$$

Conditional Pointwise Mutual Information (CPMI) van der Poel et al. (2022) have demonstrated a connection between hallucinations and token-wise predictive entropy, denoted as $H(p) = -\sum_{y \in \mathcal{V}} p_y \log p_y$. A model tends to hallucinate a token if the entropy is high. Hence, instead of penalizing the marginal probability of y_t in Equation 2 all the time, CPMI does this only when the entropy at the t-th decoding step is higher than a threshold.

$$\begin{aligned} \text{score}(y_t|\mathbf{y}_{<t}, \mathbf{x}) = &\log p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t}) - \\ &\lambda \cdot u_t \cdot \log p(y_t|\mathbf{y}_{<t}) \end{aligned} \tag{3}$$

where $u_t = \mathbb{1}\{H\left(p_\theta(y_t|\mathbf{x}, y_{<t})\right) > \tau\}$.

3 Domain-conditional Scoring Strategy

Our approach improves upon CPMI by conditioning the probability of a generated token on the source domain. In our domain-conditional strategy (PMI_{DC}), we employ the following scoring function:

$$\begin{aligned} \text{score}(y_t|\mathbf{y}_{<t}, \mathbf{x}) = &\log p_\theta(y_t|\mathbf{x}, \mathbf{x}_{dom}, \mathbf{y}_{<t}) - \\ &\lambda \cdot u_t \cdot \log p_\phi(y_t|\mathbf{x}_{dom}, \mathbf{y}_{<t}) \end{aligned} \tag{4}$$

$\mathbf{x}_{dom}$ is a domain prompt (Holtzman et al., 2021), a subset of tokens in $\mathbf{x}$ that contains information about the source domain (explained in detail below). This seemingly simple extension is well grounded in the previous observation that a summarization model is likely to hallucinate as it "templatizes" the summaries of source texts that share the same domain or topic (e.g., the transfer of a soccer player) (King et al., 2022). Accordingly, our method can account for different marginal probabilities of the same token depending on the source domain and effectively outperforms CPMI, as will be demonstrated later.

To compute the marginal probabilities $p(y_t|\mathbf{y}_{<t})$, we use a smaller language model, ϕ, while θ is a larger summarization model. The hyperparameters λ and τ can be optimized by random grid-search.

Domain Prompt Design To condition the generation probability of a token on the source domain, we incorporate domain information into the prompts of both the summarization and language models (*i.e.*, $\mathbf{x}_{dom}$). We explore three types of domain information: (1) domain-specific keywords, (2) the first sentence of the source text, (3) a random sentence in the source text (details are discussed below).

Figure 1: Example of Domain Prompt.

We assume that domain-specific keywords prime the models, enabling them to calculate the conditional probability of a token within the specified domain. We use the open-source module KeyBERT (Grootendorst, 2020) to extract three keywords from each source text (see Appendix A.4). We define domain-specific tokens as those that are not proper nouns and are frequently occurring words. We expect that these selected keywords effectively represent the source document with high similarity.

Method	Model	# Samples	*Faithfulness* AlignScore	FactCC	*Relevance* BARTScore↑	BS-Fact	*Similarity* Rouge-L	BERTScore
Beam	BART	11333	60.02	21.43	-1.8038	88.86	35.90	**91.52**
PINOCCHIO		10647[1]	57.83	16.97	-2.0958	88.81	27.98	89.91
CPMI		11333	60.09	21.53	-1.8038	88.85	**35.90**	91.52
PMI_{DC}		11333	**60.78***	**21.82**	**-1.7988***	**88.89***	35.81	91.50

Table 2: Comparison with decoding methods on BART-large. PMI_{DC} improves faithfulness and source relevance, with a slight decrease in target similarity. $*$ indicates statistical significance (p-value < 0.001) based on the paired bootstrap analysis versus CPMI.

Method	FT	AlignScore	BARTScore↑	Rouge-L
Random		**97.64**	-2.6629	11.09
FactPEG	✓	68.70	-1.9201	34.36
PMI_{DC}		60.78	**-1.7988**	**35.81**

Table 3: Comparison with fine-tuned model. Random denotes the use of a randomly selected sentence from the source text as a summarization. FactPEG represents the summarization results obtained from a fine-tuned model with the objective of faithfulness.

Method	Text
FactPEG	The crypto-currency, Bitcoin.
PMI_{DC}	The price of the virtual currency Bitcoin has fallen sharply in the wake of comments made by one of its most prominent developers.
Source	Mike Hearn, a Zurich-based developer ... published a blog calling Bitcoin a "failed" project ... Bitcoin's price fell quite sharply over the weekend ...

Table 4: An example of FactPEG summary. The model trained with the objective of faithfulness tends to focus only on factual consistency, leading to a reduction in the summarization capability of pre-trained model.

The first sentence of a source text often guides the domain for the remainder of the text, making it a reliable indicator of the source domain. However, acknowledging that this assumption may not always be robust, we consider using a random sentence from the source text as an alternative indicator of the source domain.

In addition to the domain information mentioned above, we also include a simple priming phrase in the prompt. We have discovered that using an appropriate lexical form yields better results than simply inputting the domain. We referred to the prompt design outlined by Yuan et al. (2021) to implement this prompting approach. The 18 phrases we explore include expressions such as "keyword," "in summary," and "in other words" (Appendix D).

4 Experimental Setup

Dataset We use the eXtreme Summarization Dataset, XSUM (Narayan et al., 2018), which includes BBC articles as source documents and single-sentence summaries as gold summaries.

Baselines We analyzed three baseline decoding methods: standard beam search, PINOCCHIO (King et al., 2022), and CPMI (van der Poel et al., 2022). Furthermore, we analyzed FactPEG (Wan and Bansal, 2022), which underwent separate fine-tuning using FactCC and ROUGE with the source.

Models For the summarization model, we utilized encoder-decoder structures of BART (Lewis et al., 2019) and PEGASUS (Zhang et al., 2020). As for the language model, a GPT-2-based model (Radford et al., 2019) was employed. Each of these models was pre-trained on the XSUM dataset. More details can be found in Appendix B.

Evaluation Metrics We have categorized the evaluation into three key divisions: **Faithfulness**, **Relevance** (with the source), and **Similarity** (with the target). For Faithfulness, we used AlignScore (Zha et al., 2023) and FactCC (Kryscinski et al., 2020). To measure Relevance to the source and informativeness, we employed BARTScore (Yuan et al., 2021) and BS-FACT. Lastly, to assess Similarity to the target, we utilized ROUGE-L and BERTScore (Zhang* et al., 2020).

5 Results

We present the results from BART, which are higher than those in PEGASUS. Complete result

[1]For PINOCCHIO, we have only 10,647 samples due to rejected paths. The original paper presented results for 8,345 samples after manual removal. Thus, our reported values may differ.

Type	Domain	AlignScore	BARTScore	Rouge-L
Word	Random	60.47	-1.7993	35.82
	Keyword	60.78	-1.7988	35.81
Sentence	First	61.45	-1.7706	35.52
	Random	60.57	-1.7993	35.83
	Keyword	61.16	-1.7784	35.60

Table 5: Domain comparison. Results were obtained by varying the domain under the conditions of using the BART model and the prompt `that is to say`.

Method	AlignScore	BARTScore↑	Rouge-L
PMI	60.06	-1.8041	35.88
PMI_{DC} w/o u_t	60.57	-1.7992	35.76
PMI_{DC} w/ u_t	**60.78**	**-1.7988**	**35.81**

Table 6: Effectiveness of uncertainty aware scoring. PMI refers to eq.2, PMI_{DC} w/o u_t denotes the removal of the uncertainty-aware scoring term in eq.4. PMI_{DC} w/o u_t refers to eq.4. The results show the impact of u_t.

including PEGASUS is available in Table 11. The prompt used in all cases was "That is to say," and the domain consisted of three keywords extracted from the source.

In Table 2, we compared the summarization performance of different decoding strategies with BART. Our results revealed PINOCCHIO exhibited suboptimal performance overall, and CPMI showed performance that was nearly on par with standard beam search. However, PMI_{DC} showed significant improvement in terms of faithfulness and relevance.

In Table 5, the term *Type* indicates whether this subset is at the word or sentence level, while *Domain* refers to a subset of tokens within the source. Notably, the *Keyword* approach within the word-level domain demonstrates robust performance. Therefore, we selected the *Keyword* approach for our domain prompt.

5.1 Comparison with Fine-tuned Model

FactPEG (Wan and Bansal, 2022) reduces hallucinations by incorporating factual metrics into the training process. It combines ROUGE with the source and FactCC to produce faithful summaries. In Table 3, FactPEG outperforms PMI_{DC} in terms of faithfulness (AlignScore). On the other hand, PMI_{DC} achieves a more balanced performance across different metrics.

FactPEG is trained with a focus on faithfulness, which has led to the loss of other summarization abilities. For instance, using a random sentence as a summary (as shown in the top row) demonstrates high faithfulness but a notable drop in the other two categories. Therefore, solely targeting faithfulness may risk the summarization capabilities of pre-trained models, as illustrated in Table 4.

5.2 Effectiveness of Transitioning to the PMI Objective

Recall that in PMI_{DC}, the marginal probability of a token conditional to the domain $p(y_t|\mathbf{x}_{dom}, \mathbf{y}_{<t})$ is utilized only when the model's uncertainty of a token is higher than a threshold (*i.e.*, u_t). Here, we verified whether this uncertainty-aware scoring is more effective than without u_t.

The first and second rows in Table 6 demonstrate the conversion of scores to PMI regardless of uncertainty. We emphasized the significance of improving faithfulness without sacrificing the fluency of summarization. To ensure the generation of faithful tokens while preventing a decrease in the performance of existing summarization models, it is more effective to replace only specific uncertain tokens that are suspected of hallucination, rather than adjusting all tokens using PMI.

5.3 Error Analysis

Using PMI_{DC}, we effectively controlled hallucinated terms. However, there are some failure cases, which can be classified into three cases. The first case occurs when the keyword extractor fails to extract the appropriate domain-related keywords (Table 8). In such cases, PMI_{DC} could not adequately correct the probability of domain-associated tokens. The second case is that it still has difficulties in handling proper nouns or numbers (Table 9). This is a persistent challenge for general language models, and our approach did not completely address this issue. The third case arises from the constraint of the domain. Penalizing marginally likely tokens sometimes avoid direct expressions, resulting in ambiguity (Table 10).

6 Conclusion

By employing PMI_{DC}, we successfully mitigated hallucination through uncertainty-aware scoring, without the need for fine-tuning. Our experiments clearly demonstrate the substantial advantage of our approach over conventional CPMI.

Limitations

Based on our evaluation, it is risky to solely rely on PMI while using entropy as a measure of hallucinations mathematically. We must consider the optimal points that our scoring system can achieve in beam search. Additionally, PMI is not always the superior choice compared to maximum likelihood.

We did not conduct human evaluations. Human annotation remains the most accurate method for assessing hallucinations. As mentioned earlier, automatic metrics are not flawless in measuring hallucinations. Nevertheless, it's worth noting that human judgment of the faithfulness of summaries is also imperfect (Maynez et al., 2020).

Ethical Concerns

We do not anticipate any ethical concerns with this work beyond those already documented in abstractive summarization systems and other text generators (van der Poel et al., 2022; Zhou et al., 2023; Xiao and Wang, 2021).

References

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022. Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 555–571, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods.

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.

Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. 2021. Constrained abstractive summarization: Preserving factual consistency with constrained generation.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Wan and Mohit Bansal. 2022. FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP Findings)*, Virtual.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

A Related Work

A.1 Understanding hallucinations

In abstractive summarization, *hallucinations* refer to generating content that deviates from the source material and are categorized as intrinsic and extrinsic hallucinations (Maynez et al., 2020). Intrinsic hallucinations result from generating content that contradicts the input source document's information, while extrinsic hallucinations occur when the source material is ignored (Ji et al., 2023). Our focus is on summarization, where a good summary encapsulates the content of the source document. Therefore, reducing hallucinations entails increasing *faithfulness* and *factual consistency* between the source document and the generated summary.

Zhang et al. (2023) demonstrated the snowball effect of hallucination, where if a pre-trained model provides inaccurate responses, it tends to generate subsequent incorrect explanations. The root cause of this phenomenon is the *initial committal*, where language models are trained on data in which the correct answer precedes the explanation. In other words, if the initially generated answer is incorrect, subsequent explanations tend to justify and align with this inaccuracy. Therefore, it is important to correct hallucinated content in the early stages.

A.2 Mitigating hallucinations

Various approaches have been proposed to tackle the challenge of hallucination in text generation (Li et al., 2022).

Lexically constrained decoding modifies beam search to control specific words in the output without changing the model. CAS (Mao et al., 2021) enhances factual consistency in summarization. It uses dynamic beam search to create constrained token sets focused on entities and noun phrases, improving the accuracy and faithfulness of abstractive summarization.

PINOCCHIO (King et al., 2022) is a modified beam search algorithm for text generation that uses a set called $\mathcal{R}$ to avoid disallowed paths. It tackles inconsistencies by adjusting predicted scores and backtracking using a heuristic function f_c that considers eight binary checks. High entropy and multiple backtracks result in discarded generations.

Context-aware decoding (CAD) (Shi et al., 2023) attempted to decrease hallucination in PMI by adding prompts to the unconditional term. It differs from our work in a way that they adjusted the score of all tokens with PMI and use the same prompt for all input documents.

CPMI (van der Poel et al., 2022) a significant inspiration for our work, introduced a beam-search technique to combat hallucination. It addresses the tendency of language models to generate overly general text by utilizing mutual information and internal entropy in a scoring function to detect and mitigate hallucination.

Furthermore, in a similar task utilizing uncertainty, Xiao and Wang (2021) proposed an uncertainty-aware beam search that penalizes the use of entropy. Our approach differs in that we do not consistently penalize uncertain tokens; instead, we score them with PMI when they surpass a certain threshold.

FactPegasus (Wan and Bansal, 2022) enhances abstractive summarization by reducing hallucinations through factuality integration. It modifies sentence selection by combining ROUGE metrics with the FactCC, aiming to produce faithful summaries. FactPegasus employs fine-tuning with Corrector, Contrastor, and Connector modules. Although it improves factual accuracy, it lacks in informativeness. Our work complements more balanced abstractive summarization approach.

A.3 Automatic Metrics

We have categorized the evaluation into three key dimensions: Faithfulness, Relevance (with the source), and Similarity (with the target).

To assess faithfulness, we employed **AlignScore** (Zha et al., 2023) and **FactCC** (Kryscinski et al., 2020). AlignScore divides the source document into approximately 350 segments, evaluating factual consistency with the generated text. FactCC assesses whether the generated text aligns factually with the source document, using a binary format.

To compare the relevance of the generated text with the source document, we used **BARTScore** (Yuan et al., 2021) and **BS-FACT** for evaluating their informativeness. BARTScore, which is based on the BART model, comprehensively evaluates both the informativeness and factual accuracy of the generated text. BS-FACT, derived from BERTScore, measures the precision of alignment between the generated text and the source text.

Finally, to measure Similarity with the target, we utilized **ROUGE-L** (Lin, 2004) and **BERTScore** (Zhang* et al., 2020). These metrics, traditionally used for evaluating generated text, differ from previous methods as they compare the generated text not with the source document but with the gold summary (*i.e., target*).

A.4 Keyword Extractor

We used the open-source module, KeyBERT (Grootendorst, 2020) to extract keywords from the source document. KeyBERT provides a sentence-level corpus containing labeled keywords and keyphrases extracted from random Wikipedia articles. This corpus utilizes a self-labeling method based on contextual word features, demonstrating a close alignment with human-labeled data. KeyBERT employs a bidirectional LSTM for keyword and keyphrase extraction using this self-labeled corpus.

B Implementation Details

Summarization models In our experiments, we followed a setup similar to that described in the work by van der Poel et al. (2022) to ensure a fair comparison. We conducted our experiments using computing clusters equipped with NVIDIA RTX 3090 GPUs, allocating a single GPU for each experiment. We use the checkpoint BART-LARGE-XSUM (`https://huggingface.co/facebook/bart-large-xsum`) and PEGASUS-XSUM (`https://huggingface.co/google/pegasus-xsum`).

Language model We trained two language models, since the BPE step differed for BART-large and PEGASUS. Both architectures are from the GPT-2 family architecture (Radford et al., 2019) (available at `https://huggingface.co/gpt2`). The configurations for the language models are as follows: both have 512 embeddings, 6 layers, and 8 heads. However, there is a variation in the output vocabulary size, with BART having 50,265 and PEGASUS 96,103. The maximum token length for both models is set to 2,048 tokens, and they operate with an update frequency of 32. Both models share a learning rate of 5.0×10^{-4}. In terms of validation metrics, BART-large included a loss of 3.16744 and a perplexity of 24.57401, while PEGASUS consisted a loss of 3.25238 and a perplexity of 26.68345.

Why do we need an additional model? We have employed two types of models: a larger summarization model (BART-large: 406M, PEGASUS: 223M) and a smaller language model (GPT-2-based model: 45M). There are two reasons why we chose to use a model with an additional decoder-only structure instead of the decoder of the existing summary model.

Firstly, an extra forward pass is required for the unconditional (*i.e.*, domain-conditional) term. Therefore, employing a smaller language model is faster. This is also related to the latest research on speeding up additional forwarding (*e.g.* speculative sampling techniques, (Chen et al., 2023)).

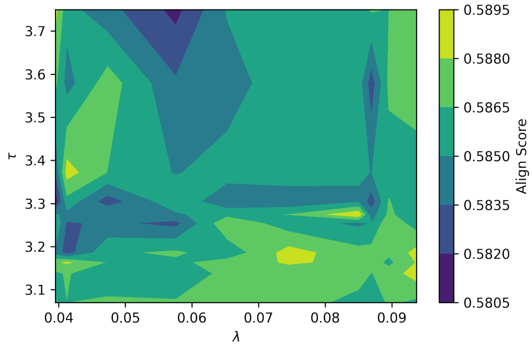
Secondly, a decoder-only structure, trained for the next token prediction, provides a more suitable unconditional distribution than an encoder-decoder structure. This is because the decoder in the encoder-decoder structure requires the encoder output for cross-attention. Even if all encoder inputs were padded, we did not obtain an appropriate unconditional distribution. The reason for this is that there are some samples with no source document in the training dataset. So, if the encoder input is entirely padded, the decoder only reflects the distribution of the corresponding outlier sample, not the distribution in the entire dataset.

C Searching Hyperparameters

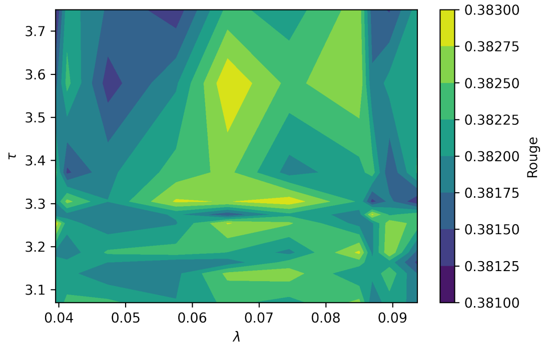
We used the same hyperparameters as CPMI, as reported in their paper. For BART, we set τ to 3.5987 and λ to 6.5602×10^{-2}. Our method outperformed CPMI, demonstrating effective summarization without hallucination (see Table 2). For PEGASUS, we determined the hyperparameters by examining the AlignScore with 3,000 samples from the validation set, using CPMI, not PMI_{DC}. The values we obtained are $\tau = 3.304358$ and $\lambda = 7.4534 \times 10^{-2}$. Note that CPMI relied on human-annotated data at the token level (Zhou et al., 2021). This approach is not only extremely costly and challenging but also lacks precision. However, since we have removed such human intervention, PMI_{DC} is more applicable.

D Prompt Design

To search for the best prompt, we referred to the prompt set proposed by Yuan et al. (2021). They used manually devised seed prompts and gathered paraphrases to construct our prompt set in order to find suitable prompts within a search space. The seed prompts, along with some examples of paraphrased prompts, are shown in Table 7. We have discovered that it is more effective to add additional prompts to make them more lexical and natural, rather than simply using the domain as the prompt. Specifically, we obtained the phrase '`that is to say`'. We used all entries in the prompt set by prefixing the language model input and appending the summarization model input. Furthermore, we found that consistently using lexically natural prompts was better than relying solely on domains in terms of faithfulness and relevance.



(a) PEGASUS. CPMI. AlignScore



(b) PEGASUS. CPMI. ROUGE

Figure 2: Searching for hyperparameters. For PEGASUS, we utilized the same hyperparameter settings for comparison with CPMI. We considered 10x10 hyperparameter pairs through a random uniform grid search on 3,000 samples in a validation set using alignscore. Alternatively, we can also be identified using ROUGE, suggesting that the optimal configuration may vary depending on experimental results.

Seed	Prompt Set		
keywords	Keywords Concepts	Topics Features	Components Points
in summary	In summary When all is said and done	To be brief Bringing up the rear	Last of all In short
in other words	In other words Take for example	That is to say To put it another way	To rephrase it Case in point

Table 7: Seed prompts and examples of final prompts.

E Error Analysis

Method	**Text**
Domain	bia, falkirk, bi
Source	However, the Bairns boss has underlined that any forward signing will need to exhibit even more quality than two of his promising youngsters. "If I bring another striker in he's got to be better than young Botti **Bia-Bi** and Scott Shepherd," said Houston. "I would be looking for the more experienced type, and another defender would come in handy as well." Eighteen-year-old Bia-Bi, a London-born Scot who has progressed through **Falkirk**'s academy, glanced in a fine equalising header against Cowdenbeath on Saturday to ensure Houston's side left Central Park with a point...
PMI_{DC}	Falkirk manager Peter Houston has not ruled out bringing in a new striker in the January transfer window.
Gold	Peter Houston is still seeking to fine-tune his Falkirk squad, with a striker and defender pinpointed as priorities.

Table 8: Case 1 error. Inconsistent words are highlighted in *red* fonts. Extracted keywords may not fully reflect domains of source text. In this example, the domain should be more related to terms like transfer or football rather than specific names of individuals or institutions. Therefore, the terms closely associated with transfer (such as *January*) were not adequately penalized.

Method	**Text**
Domain	invest, richest, investment
Source	The **investment** follows "several months of negotiations", a company statement to the Saudi stock exchange said. The prince, who is one of the world's **richest** men, owns stakes in many well-known companies, including News Corporation. He also has investments in a number of media groups in the Arab world. "Our investment in Twitter reaffirms our ability in identifying suitable opportunities to **invest** in promising, high-growth businesses with a global impact," Prince Alwaleed said."
PMI_{DC}	Saudi Arabia's Prince Alwaleed bin Talal has bought a 10% stake in Twitter in a deal worth $2bn (31.8bn).
Beam	Saudi Arabia's Prince Alwaleed bin Talal has agreed to buy a 10% stake in Twitter for $3bn (32.3bn).

Table 9: Case 2 error. Inconsistent words are highlighted in *red* fonts. The appropriate domain, but not properly regulated the numbers. Hallucinations related to proper nouns, numbers and statistics, have long been significant issues in general language models. Our approach could not completely address this issue.

Method	**Text**
Domain	claire, marathon, equestrian
Source	When **Claire** was told she would spend the rest of her life in a wheelchair after a spinal injury, she wanted to get back on her feet as quickly as possible and regain her independence. For the past three months she has been training intensively for the **marathon** using a robotic walking suit to prove she is just as determined as in her sporting days. ... former champion British **equestrian** Lucinda Green. "There's a lot of people who are worse off than me and haven't got the support I've got, so I want to raise as much as I can. "But, when the marathon is over, Claire thinks that for the first time in six years, she will be delighted to return to her wheelchair.
PMI_{DC}	A paralysed equestrian rider is taking part in the London Marathon in a bid to become the first person in the world to walk unaided.
Beam	Claire Gwynne, who was paralysed from the chest down in 2006, is taking part in the London Marathon.

Table 10: Case 3 error. Inconsistent words are highlighted in *red* fonts. Constraints of domain-conditional term can prevent direct expressions, potentially resulting in ambiguity and generation of incorrect results. In this example, penalizing the domain term *Claire* allowed for the removal of the hallucinated term *Gwynne*. However, apart from this, the conveyed information remained somewhat incorrect.

Method	Model	# Samples	*Faithfulness* AlignScore	FactCC	*Relevance* BARTScore↑	BS-Fact	*Similarity* Rouge-L	BERTScore
Beam	BART	11333	60.02	21.43	-1.8038	88.86	35.90	**91.52**
PINOCCHIO	BART	10647[2]	57.83	16.97	-2.0958	88.81	27.98	89.91
CPMI	BART	11333	60.09	21.53	-1.8038	88.85	**35.90**	91.52
PMI_{DC}	BART	11333	**60.78**	**21.82**	**-1.7988**	**88.89**	35.81	91.50
Beam	PEGASUS	11333	59.28	22.02	-1.9636	88.64	38.02	**91.91**
CPMI	PEGASUS	11333	59.31	21.91	-1.9617	88.64	38.01	91.91
PMI_{DC}	PEGASUS	11333	**59.40**	**22.09**	**-1.9590**	**88.64**	**38.06**	91.91

Table 11: Comparison with decoding methods on BART-large and PEGASUS. PMI_{DC} improves faithfulness and source relevance, with a slight decrease in target similarity.