Revealing The Intrinsic Ability of Generative Text Summarizers for Irrelevant Document Detection

Anonymous ACL submission

Abstract

In Retrieval-Augmented Generation (RAG), generative models are prone to performance degradation due to retrieved irrelevant documents. Adding irrelevant documents to the training data and retraining language models incurs significant costs. Supervised models can detect irrelevant documents in the retrieved results and avoid retraining, but they cannot counter domain shifts in the real world. By introducing a method that emphasizes the unique features of infrequent words, we reveal the ability of the cross-attention mechanism to detect irrelevant documents within the inputs of generative models. We present CODE, a novel irrelevant document detector using a closed-form expression rooted in cross-attention scores. Our experimental results validate the superiority of CODE under in-domain and cross-domain detection. For in-domain detection, CODE achieves a 5.80% FPR at 95% TPR vs. 30.3% by supervised baseline on the T5-Large and Delve domain. When sampling irrelevant documents from out-of-domain, the FPR of CODE decreases from 5.8% to 0.1%, while the FPR of the supervised baseline increases from 30.3% to 34.3%. For more insight, we highlight the importance of cross-attention, word frequency normalization, and integrating in-domain irrelevant documents during pretraining .[1]

1 Introduction

The RAG system (Lewis et al., 2020) can access external knowledge bases for up-to-date and long-tail knowledge, thereby enhancing generation quality. However, in real-world applications, the retriever may return irrelevant documents, significantly degrading performance (Shi et al., 2023). Yoran et al. (2023) and Asai et al. (2023) highlight that irrelevant documents in retrieval-augmented knowledge-sensitive tasks lead to low-quality generations. In open-domain text summarization, Giorgi et al. (2022) find through experimental simulation that irrelevant documents in retrieval results are the primary cause of declining generation quality. Case studies of RAG systems in academic fields by Barnett et al. (2024) reveal that the retriever sometimes fail to rank relevant documents first, often returning irrelevant or noisy information, causing the model to generate incorrect results.

To improve generation quality, existing methods retrain language models to counter irrelevant content (Giorgi et al., 2022; Yoran et al., 2023; Asai et al., 2023; Wang et al., 2024), which incurs high economic costs. Yoran et al. (2023) propose a supervised approach to learn the relevance between the query and retrieved documents, removing irrelevant documents before inputting them into the language model. Although this method avoids fine-tuning the generative model, it struggles with performance degradation due to domain shifts in real-world scenarios (Calderon et al., 2024; Elsahar and Gallé, 2019).

This paper highlights the significant potential of using intrinsic neuron output of generative language models to detect irrelevant documents. It should be noted that the generative models mentioned in our method below are specialized for detecting irrelevant documents, rather than the original model in the RAG system. Specifically, we demonstrate the substantial potential of the cross-attention mechanism in generative text summarizers based on the encoder-decoder architecture (Vaswani et al., 2017) for this purpose. Our initial observations indicate that rare words in input documents often signify unique features, helping the model discern their relevance. Seq2seq models pretrained with a mixture of irrelevant document data tend to assign lower cross-attention scores to rare words in irrelevant documents during text generation. Conversely, words in relevant documents typically receive higher scores. Based on these ob-

[1] Our code is available at: https://anonymous.4open.science/r/code-A5B1/

servations, we propose a pretraining method for text summarizers that incorporates irrelevant documents, enabling the cross-attention mechanism to capture differences between relevant and irrelevant documents. Building upon the pretrained model, we introduce CODE (Cross-attention based irrelevant dOcument DEtector), a method for detecting irrelevant documents based on cross-attention scores in generative language models. We categorize irrelevant documents into **In-domain** and **Out-of-domain** to verify the effectiveness of CODE for in-domain and cross-domain detection. The core contributions of this paper include:

- Proposal of a method to pretrain generative language models incorporating irrelevant documents. We subsequently introduce the CODE detector, which computes average cross-attention scores, normalized by word occurrences, between the generated summary and each document in the sequence.

- Introduction of data pipelines to build four pretraining datasets integrated with irrelevant documents. Additionally, we present four in-domain irrelevant document detection datasets and sixteen cross-domain irrelevant document detection datasets.

- An ablation study underscoring the impact of cross-attention, word frequency normalization, and the incorporation of irrelevant documents during pretraining.

2 Related Work

Retrieval-Augmented Generation. RAG system employs sparse (Robertson and Walker, 1997; Robertson et al., 2009) or dense (Karpukhin et al., 2020) retrievers to link generative models with external non-parametric knowledge bases, addressing the challenges of generative models such as accessing up-to-date knowledge (Ram et al., 2023), integrating long-tail data (Mallen et al., 2022), and preventing training data leakage (Carlini et al., 2021). RAG can also reduce the parameters of the model (Izacard et al., 2023) to reduce generation costs. The concept of RAG was first introduced by Lewis et al. (2020), who proposed using the top-K documents returned by a retriever as direct inputs to the model to enhance performance on knowledge-sensitive tasks. Beyond direct input, the results returned by the retriever can also be integrated into the model in a latent form to improve generation quality (Izacard and Grave, 2020; Borgeaud et al., 2022). RAG has been applied to enhance various text-to-text generation tasks, including Question Answering (Wang et al., 2023), Text Summarization (Bertsch et al., 2024), and Fact Verification (Huang et al., 2022). Besides text modalities, RAG has also been utilized in other modalities such as audio (Yuan et al., 2024), image (Ramos et al., 2023), and video (Pan et al., 2023).

Enhance RAG Systems by Resisting Irrelevant Documents. The results returned by the retriever can include documents irrelevant to the content to be generated, degrading the quality of RAG systems. Researchers are exploring methods to resist this issue and enhance RAG performance. Giorgi et al. (2022); Yoran et al. (2023) add irrelevant documents to training data and retrain the model to improve robustness. Asai et al. (2023) use a LLM to evaluate the relevance of retrieval results for critical generation. Wang et al. (2024) introduce a rank head to help LLMs perceive document relevance and guide final generation. These approaches require extensive training or fine-tuning, incurring high costs. Yoran et al. (2023) propose a supervised approach to learn query-document relevance, removing irrelevant documents before the retrieval results are fed into the generative model. Although this method avoids fine-tuning the generative model, it struggles with performance degradation from domain shifts in real-world scenarios (Calderon et al., 2024; Elsahar and Gallé, 2019).

3 Preliminaries and Problem Formulation

Text Summarizers Pretrained with In-domain Irrelevant Documents. Let the X denote the document consisting of a sequence of words, $\mathbb{P}(X|\mathcal{D})$ denote a document sampling distribution defined on the document set $\mathcal{D}$. Let $\mathcal{X}$ represent a sequence of documents used for summarization. We note that the documents in $\mathcal{X}$ may originate from different topics. Let the sequence of words $Y(\mathcal{X})$ denote the summary of the document set $\mathcal{X}$. Let $\mathcal{C} = \{(\mathcal{X}_i, Y_i)\}_{i=1}^{n}$ represent the pretraining set for text summarization. Each document in the sequence $\mathcal{X}_i$ is drawn from an underlying mixed document distribution $\mathbb{P}(X|\mathcal{D}_i, \mathcal{D}'_i)$ consisting of the document sets $\mathcal{D}_i$ and $\mathcal{D}'_i$. Documents in $\mathcal{D}_i$ are related to the topic to be generated, so the topics of the documents sampled from $\mathcal{D}_i$ are related to each other, and the documents sampled from $\mathcal{D}'_i$ are irrelevant documents in $\mathcal{X}_i$. $\mathcal{D}_i, \mathcal{D}'_i, \mathcal{X}_i$ are derived from the same domain, i.e., the same original dataset. We refer to documents in $\mathcal{X}_i \cap \mathcal{D}_i$ as **relevant documents**, and those in $\mathcal{X}_i \cap \mathcal{D}'_i$ as **in-domain irrelevant documents**. We use **in-domain** to indicate that both relevant and irrelevant documents are sampled from the same dataset domain, but on different topics, to distinguish them from the problem of detecting irrelevant documents that may originate from different domains.

A summarizer G processes the document set $\mathcal{X}$ to produce a summary $\hat{Y}(\mathcal{X})$. We employ the generative language model (GLM) for this task. We pretrain G to ensure that the generated $\hat{Y}(\mathcal{X}_i)$ aligns with the ground truth summary Y_i for all samples in the training set $\mathcal{C}$. As mentioned earlier, each document set $\mathcal{X}_i$ in the set $\mathcal{C}$ contains in-domain irrelevant document.

GLM-based Irrelevant Document Detection Problem. Let the generative model G be a text summarizer pretrained on the pretraining set $\mathcal{C}$. We construct irrelevant document detectors f_θ using the neuron outputs inside G. Consider $\mathcal{U}$ as a input document sequence containing relevant and irrelevant documents. For $\mathcal{U}$, we use the binary vector $V \in \{0,1\}^{|\mathcal{U}|}$ as the label vector, where V_i equals 0 if the i-th document in $\mathcal{U}$ is an irrelevant document and 1 otherwise. The irrelevant document detection dataset can be represented as $\mathcal{C}_{\text{detect}} = \{(\mathcal{U}_k, V_k)\}_{k=1}^{m}$. Notably, we allow relevant and irrelevant documents to come from the same dataset domain, in which case the problem is referred to as the **in-domain** detection problem. If the relevant and irrelevant documents come from different dataset domains, the problem is called the **cross-domain** detection problem.

4 GLM-based Irrelevant Document Detector

In this paper, we primarily focus on generative language models using the Transformer encoder-decoder architecture (Vaswani et al., 2017), specifically BART (Lewis et al., 2019) and T5 (Raffel et al., 2020). To see the influence of the model size, we select BART-Base, BART-Large, T5-Base and T5-Large. We pretrain all GLMs on each of the pretraining sets introduced in the next section.

4.1 Baselines

We concatenate the neuron outputs inside the GLM with a multi-layer perception to construct two supervised baselines. Given the potentially large number of neurons in GLMs, to reduce the computational complexity, we streamline the computation by using the input from the last encoder-decoder attention layer as the input to the multi-layer perceptron (MLP).

Frozen. First, we feed a document sequence into the GLM and obtain a generated summary. Probing the input of the last encoder-decoder attention layer, we obtain the word embeddings of the document sequence from the encoder, as well as the word embeddings of the corresponding summary from the decoder. Second, to get the embeddings of the entire sequence of the document or summary, we perform a mean pooling on the obtained word embeddings that are also adopted in references (Reimers and Gurevych, 2019; Gao et al., 2021). Finally, we feed the word embedding into a MLP to detect the irrelevant documents in the input sequence. In the supervised training phase, we freeze all parameters of the pretrained GLM and only fine-tune the parameters of the MLP.

Finetuning-all (FT-ALL). We adopt the same architecture used in the previous baseline for irrelevant detection. The only difference lies in the training stage, where the parameters of the pretrained GLM are fine-tuned along with MLP parameters.

4.2 CODE: Cross-attention based irrelevant dOcument DEtector

In this section, we propose CODE, which eliminates the need for further fine-tuning like baselines once the GLM is pretrained. Similar to baselines, we also probe the attention weights of the last cross-attention layer. But, for each document, we only calculate closed-form metric to determine whether the document is irrelevant or not.

Now we formally present our method. We concatenate all documents $\mathcal{X} = \{X_1, ..., X_m\}$ and input at once to the text summarizer G. The GLM G outputs a summary $\hat{Y}$. We input each word $\hat{y}$ in the summary $\hat{Y}$ to the decoder independently. Now we get a cross-attention matrix between the generated summary and concatenated documents. When the cross attention layer has multi-head (Vaswani et al., 2017) and each head is equipped with a unique attention matrix of the same size, we average all attention matrices across different heads into one matrix. For each word x in the concatenated document sequence and each word $\hat{y}$ in the summary sentence $\hat{Y}$, let $Att(\hat{y}, x) \in [0,1]$ denote the attention score in the attention matrix between the word $\hat{y}$ and x. We use $\frac{1}{|X_i|}\sum_{x\in X_i} Att^{\alpha}(\hat{y}, x)$ to measure the relevance between word $\hat{y}$ and input document X_i. Let $p(\hat{y})$ denote the word frequency of $\hat{y} \in \hat{Y}$ across all generated summaries. We use $\frac{1}{p^\beta(\hat{y})}$ to assign more weights to the contribution of less frequent words. We define the relevance score $r(\hat{Y}, X_i) \in \mathbb{R}_+$ between the generated summary $\hat{Y}$ and the i-th document X_i as follow,

$$r(\hat{Y}, X_i) = \frac{1}{|\hat{Y}|} \sum_{\hat{y} \in \hat{Y}} \frac{1}{p^\beta(\hat{y})} \left[\frac{1}{|X_i|} \sum_{x \in X_i} Att^\alpha(\hat{y}, x) \right]. \tag{1}$$

Hyper-parameters α and β are used to control the contribution of the attention score and word frequency in calculating the relevance. For a given threshold δ, we say that the document X_i is irrelevant if $r(\hat{Y}, X_i) \leq \delta$ and it is a relevant document, otherwise. CODE is more efficient than baselines. See Appendix A.11 for time consumption.

5 Datasets

5.1 Data Pipeline

Pipeline for Pretraining with In-domain Irrelevant Document. The source text summarization dataset includes relevant document sequences and their corresponding summaries. To create a text summarization pretraining dataset with in-domain irrelevant document, we employ a two-phase data pipeline. In the *relevant document sampling* phase, we select a sample $(\mathcal{X}, Y)$ from the source dataset, where $\mathcal{X}$ represents a document sequence and Y is its summary. Then, we randomly select two documents from the sequence $\mathcal{X}$, denoted as $\mathcal{X} = (X_1, X_2)$. We regard these two documents as relevant docments. Next, in the *irrelevant document injection* phase, we first randomly select two irrelevant documents Z_1 and Z_2 from another two different document sequences in the same dataset. These irrelevant documents are randomly at three positions: before X_1, between X_1 and X_2 and after X_2. After injection, the document sequence, along with the summary Y, constitutes a sample in our pretraining set. We note here that all irrelevant documents in the pretraining dataset originate from the same dataset domain.

Pipeline for Irrelevant Document Detection. We employ the same pipeline to create irrelevant document detection datasets. The only difference is that the detection dataset does not contain the ground truth summary. In the in-domain detection task, we sample the irrelevant document from the same source text summarization dataset, while in the cross-domain detection task, we sample the irrelevant document from a different source dataset.

5.2 Pretraining Datasets with In-domain Irrelevant Documents

We choose four English source datasets: **CNN/Daily Mail** (Nallapati et al., 2016), **SAMSum** (Gliwa et al., 2019), **Delve** (Akujuobi and Zhang, 2017; Chen et al., 2021) and **S2orc** (Lo et al., 2019; Chen et al., 2021) to build our pretraining dataset (**-PT**). The first dataset comes from the news domain, the second from dialogues, and the last two belong to the academic domain.

Each data sample in the above pretraining datasets contains two relevant documents, two irrelevant documents, and one summary. It should be noted that for the Delve and S2orc datasets, we consider each abstract paragraph as a document, and for the CNN/Daily Mail and SAMSum datasets, we mimic the operation of segmenting long texts in the RAG system by considering each chunk obtained as a document (Lewis et al., 2020). The dataset partitioning is shown in Table 1. See Appendix A.1 for the detailed statistics and construction method of each pretraining dataset.

Table 1: The major statistics of datasets. $*$ indicates shared validation set or test set. See Appendix A.1 for the detailed statistics.

Dataset	Training	Validation	Test
CNN/Daily Mail-PT	42.387K	5.298K	5.298K
SAMSum-PT	3.273K	0.409K	0.409K
Delve-PT	8K	1K	1K
S2orc-PT	20K	2K	2K
CNN/Daily Mail-ID	20K	2.5K	2.5K×5
SAMSum-ID	3.273K	0.409K	0.409K×5
Delve-ID (1K)	1K	100^*	$1K\times5^*$
Delve-ID (8K)	8K		
S2orc-ID	2K	200	2K×5

5.3 Irrelevant Document Detection Datasets

We provide an overview of the in-domain and cross-domain detection datasets (**-ID**) in the following.

In-domain detection sets consist of relevant and irrelevant documents sampled from the same dataset domain. We get four in-domain detection datasets from CNN/Daily Mail, SAMSum, Delve and S2orc, respectively.

Cross-domain detection sets comprise relevant and irrelevant documents from varying domains. For each domain from which relevant documents are sourced, irrelevant documents are extracted from the other three domains, leading to three unique cross-domain test sets. To assess detection against the documents composed of random garbled characters, we create a set with randomly generated documents using words tokenized from four summarization datasets. This results in four cross-domain test sets for each domain. Each cross-domain test set size is consistent with the in-domain set, and both types share the same training and validation datasets. In cross-domain detection, hyper-parameter tuning is exclusively done on in-domain irrelevant documents, precluding prior knowledge of cross-domain irrelevant documents during testing.

Each data sample in the above irrelevant document detection datasets contains two relevant documents and two irrelevant documents. The dataset partitioning is presented in Table 1. Each detection dataset contains a in-domain training set, a in-domain validation set, a in-domain test set and four cross-domain test sets.

6 Experiments

6.1 Experimental Setups

Pretraining Summerizers. We employ Hugging Face Transformers[2] (Wolf et al., 2020) and AdamW optimizer with default parameters. Additional pre-training details are in the Appendix A.2.1. We select the checkpoint with the lowest evaluation loss for irrelevant document detection. Generative quality is assessed using ROUGE (Lin, 2004), with results in the Table 7 in Appendix A.2.2.

Baselines. We employ a three-layer MLP with ReLU neurons. The input dimension N is twice the dimension of the attention layer. Regarding the dimension of the MLP hidden layer, we find that increasing the dimension hardly improves the detection performance. The experimental results are shown in the Appendix A.10. Therefore, we set the dimension of the first, second, and third layer is $4N$, $2N$ and N, respectively. Training setup details are reported in Appendix A.2.3.

CODE. There are two hyper-parameters α and β in CODE. We note that our method does not employ any fine-tuning in the detection phase, except that we run the hyper-parameter tuning on α and β. Thus, CODE is deterministic and does not have standard deviations. We search the hyper-parameters α in the range $[0, 2]$ with an interval of 0.1 and β in the range $[0, 2]$ with an interval of 0.2. This implies that we search for the best setting in 231 hyper-parameter combinations. We select the model with the lowest FPR at 95% TPR for testing.

6.2 Main Results

In this subsection, we present the main results. We use **TPR at 95% FPR**, **AUROC** (Fawcett, 2006) and **AUPR** (Manning and Schutze, 1999; Saito and Rehmsmeier, 2015) to evaluate the detection performance. Please refer to Appendix A.3 for further details.

CODE vs. Baselines. Figure 1 displays ROC curves for CODE (blue) and the baseline Frozen (red) using the T5-Large architecture on the in-domain detection dataset Delve-ID (1K). A substantial performance gap is evident, with CODE significantly outperforming the baseline. For instance, at a 95% TPR, CODE reduces the FPR from 30.3% to 5.8%. Comprehensive evaluation results can be found in Table 2 and Table 10 in Appendix A.3, highlighting that CODE consistently outperforms the baselines across almost all settings.

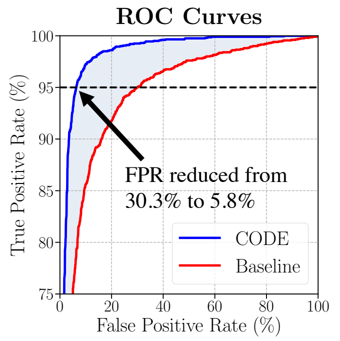

Figure 1: The ROC curves of CODE and Frozen evaluated on T5-Large and Delve-ID (1K).

Fine-tuning Dataset Size. To assess the impact of fine-tuning dataset size, we conducted experiments on Delve-ID using various set sizes. Interestingly, we observed that CODE exhibits low sensitivity to the set size, with consistent performance, such as a 5.80% FPR on Delve-ID (1K) compared to 5.55% on Delve-ID (8K) with the T5-Large architecture. In contrast, both baselines show sensitivity to the set size, with notable differences in performance, such as a 25.63% FPR on Delve-ID (1K) compared to 18.28% on Delve-ID (8K) using the T5-Large architecture.

Pretraining Checkpoint. We explored the impact of checkpoint selection during the pretraining phase on irrelevant document detection. To illustrate, we tracked the summarization and detection performance of checkpoints during pretraining using the T5-Large architecture on Delve. In Figure 2 (a), we plotted pretraining validation loss against the detection FPR of CODE at each checkpoint. Our findings show that during the initial four epochs of pretraining, validation loss consistently

[2] https://huggingface.co/

Table 2: Evaluation results of CODE and baselines for in-domain irrelevant document detection. All values are percentages. ↑ indicates that larger values are better, and ↓ indicates that smaller values are better. Characters "B" and "L" denote the Base and Large models, respectively. The hyper-parameters α and β of CODE are searched by minimizing FPR at 95% TPR, and detail can be found in Table 12 in Appendix A.3.

	Models	FPR (95%) TPR (↓)	AUROC (↑)	AUPR (↑)
		CODE/Frozen/FT-ALL		
Delve-ID (1K)	T5-L	**5.80**/30.30/25.63	**98.08**/92.87/94.59	**97.03**/93.57/92.60
	T5-B	**32.30**/65.97/57.75	**90.08**/84.52/85.21	**83.76**/82.62/82.92
Delve-ID (8K)	T5-L	**5.55**/16.85/18.28	**98.16**/93.62/95.87	**97.23**/94.01/95.18
	T5-B	**31.50**/60.22/47.98	**90.36**/86.32/87.64	**84.34**/85.40/87.49
S2orc-ID	T5-L	**1.08**/10.40/6.05	**99.54**/96.01/97.69	**99.27**/95.59/97.32
	T5-B	**2.53**/15.82/11.65	**99.00**/96.68/96.87	**97.95**/96.51/96.01
SAMSum-ID	T5-L	**0.60**/5.50/0.65	**99.87**/98.67/99.68	**99.87**/98.78/98.60
	T5-B	**0.61**/8.44/1.22	**99.66**/99.21/97.46	**99.43**/99.00/96.68
CNN/Daily Mail-ID	T5-L	**0.00**/0.20/0.32	**99.99**/99.85/99.77	**99.99**/99.81/99.79
	T5-B	**0.12**/0.82/0.29	**99.96**/99.62/99.80	**99.96**/99.56/99.70

decreases, leading to a notable reduction in detection FPR. This suggests that domain-specific pretraining enhances detection within those domains. However, as the pretraining continues, we observed an increase in validation loss, indicating potential overfitting. Intriguingly, the detection FPR remains relatively stable, implying that while overfitting may occur during pretraining, it might not significantly impact the detection performance of CODE.

Attention Layer. In CODE, we input the output from the final cross-attention layer into the detector. Both T5 and BART architectures consist of multiple cross-attention layers, prompting us to investigate how the choice of cross-attention layers impacts detection performance, as shown in Figure 2 (b). Our findings consistently show that the lowest FPR at 95% TPR and the highest AUROC consistently occur in the cross-attention layer closest to the final layer, which is adjacent to the output layer, across all configurations. Additionally, in Figure 2 (b), we observed that the last three layers exhibit similar detection FPRs. This indicates that performance variation is minimal when selecting attention layers near the output.

Document Similarity. Detection performance is notably affected by the degree of similarity between irrelevant and relevant documents. Greater similarity between them poses a more challenging detection task. To quantify this similarity, we calculated the average cosine similarity between the embeddings of irrelevant and relevant documents within a document sequence. Specifically, we employed the Sentence-BERT model (Reimers and Gurevych, 2019) to extract document embeddings. The formal definition of similarity between irrelevant and relevant documents in dataset $\mathcal{C}$ is represented as follows, where $H(X)$ denotes the embedding vector of document X, $\mathcal{X}^{\text{irr}} \subset \mathcal{X}$ is the set of irrelevant documents in the input document sequence:

$$\text{sim}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\mathcal{X} \in \mathcal{C}} \Bigg[\frac{1}{|\mathcal{X}^{\text{irr}}|(|\mathcal{X}| - |\mathcal{X}^{\text{irr}}|)} \sum_{X \in \mathcal{X}^{\text{irr}}} \sum_{X' \in \mathcal{X} \setminus \mathcal{X}^{\text{irr}}} \frac{\langle H(X), H(X') \rangle}{\|H(X)\|_2 \cdot \|H(X')\|_2} \Bigg] \tag{2}$$

In Figure 2 (c), we depicted dataset similarity and detection performance across various domains using the T5-Large architecture. Our observations show that as irrelevant documents become more similar to relevant ones, the detection of FPR increases. This suggests a positive correlation between the similarity of relevant and irrelevant documents and detection errors. Additional results for other architectures can be found in Appendix A.6.

Cross-domain Detection. Table 2 presents the detection performance of CODE when relevant and irrelevant documents are from the same dataset domain. We anticipated this performance consistency even when fine-tuning hyper-parameters of CODE in one domain for detecting irrelevant documents in

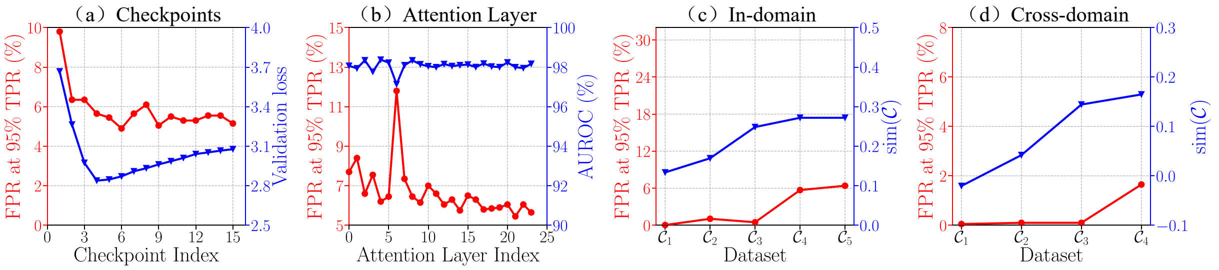

Figure 2: Performance of CODE under different settings. Results for other settings can be found in Appendix A.4, A.5, A.6 and A.7. (a) Performance of CODE vs. pretraining validation loss under different checkpoints. (b) Performance of CODE vs. different choice of attention layers. (c) Similarities between relevant and irrelevant documents vs. detection performance. $\mathcal{C}_1$ to $\mathcal{C}_5$ represent CNN/Daily Mail, S2orc, SAMSum, Delve (8K) and Delve (1K), respectively. (d) Performance of CODE vs. different domains. The relevant documents sourced from the Delve domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing SAMSum, CNN/Daily Mail, Random Domain, and S2orc.

another. Table 13, 14 in Appendix A.7.1 report the performance of CODE and the baselines in cross-domain detection, using hyper-parameters derived entirely from the in-domain detection task. Compared with Table 10, The performance of CODE is significantly improved when the domain of irrelevant documents drifts, while the performance of the supervised model is significantly reduced. For example, under the T5-Large model, when the Delve dataset is used as the source of relevant documents and CNN/Daily Mail is selected as the source of out-of-domain irrelevant documents, compared with the in-domain detection task, the FPR of CODE decreases from 5.8% to 0.1%, while the FPR of the supervised model Frozen increases from 30.3% to 34.3%. This is because models based on fully supervised learning have difficulty generalizing to data distributions out of the training domain. Figure 2 (d) depicts performance variations in diverse cross-domain detection scenarios, utilizing the T5-Large. Additional results for other pretrained models are in Appendix A.7.2. In Figure 2 (d), CODE demonstrates robust performance across different domains, although the detection FPR increases with the increase of the similarity between out-of-domain irrelevant and relevant documents, the maximum FPR does not exceed 1.64%.

7 Discussions

In this section, we investigate the effectiveness of word frequency, cross-attention and in-domain irrelevant documents used in the pretraining phase.

Effectiveness of Word Frequency Hyper-parameter β**.** Given the richer semantic content in bi-gram phrases compared to individual words, we use the bi-gram phrases as our primary unit of analysis. In CODE, for each word $\hat{y}$ in summary $\hat{Y}$, we calculate the average attention scores with words in the document X and normalize it by the frequency of $\hat{y}$ raised to the power β. We select a positive β to accentuate the effects of infrequent bi-grams. Figure 3 (a) showcases how detection error varies with different β values. Optimal results are attained with a positive β, but performance declines if β is too large, suggesting the importance of moderate emphasis on infrequent words. To understand this, we conduct the following experiment. We determine their occurrence in four domains: CNN/Daily Mail, SAMSum, S2orc and Delve, represented as $f_1(x)$ to $f_4(x)$. The total occurrence of a phrase x is $f(x) = \sum_i f_i(x)$. The metric *concentration* is defined as $\textbf{conc.}(x) = \frac{\max_i f_i(x)}{f(x)}$, representing how bi-gram phrases are concentrated among domains. In Figure 3 (b), bi-grams with fewer than five occurrences are domain-specific, whereas those with more than 128 are domain-agnostic. Emphasizing infrequent bi-grams can enhance irrelevant document detection since domain-specific phrases differ significantly across domains. Moreover, infrequent bi-grams typically exhibit higher average cross-attentions compared to their frequent counterparts, which may also benefit detection. To see this, let $\mathcal{A}(x) = \frac{1}{|\hat{Y}|}\sum_{\hat{y}\in\hat{Y}} Att^{\alpha}(\hat{y}, x)$ represent the mean cross-attention between summary $\hat{Y}$ and bi-gram x. Figures 3 (c) and (d) display the distribution of $\mathcal{A}(x)$ for bi-grams in relevant and irrelevant documents, respectively, across different bi-gram occurrence regimes. We observe higher average cross-attentions on less frequent bi-grams. However, this does not imply that frequent bi-grams are inconsequential in identifying relevant documents. Some, especially those with very high occurrence counts, may also be domain-specific terminologies. For instance, the term “Manchester United" appears 1,552 times but is exclusively found in the

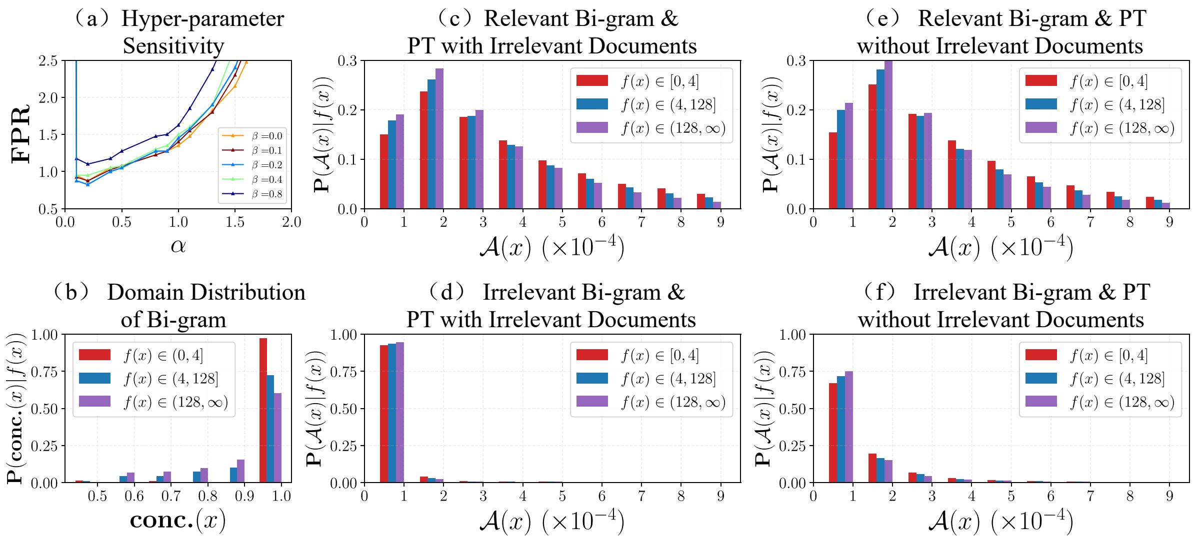

Figure 3: (a) FPR at a 95% TPR for our method under various hyper-parameters, evaluated on T5-Large and S2orc testset. Results for other settings can be found in Appendix A.8. (b) Domain distribution of bigrams with different occurrences. Figures (c) to (f) show bi-gram distributions. Bi-grams are from relevant documents in (c) and (e) and from irrelevant documents in (d) and (f). GLM is pretrained with irrelevant documents in (c) and (d) and without irrelevant documents in (e) and (f). The x and y-axis represent the cross-attention $\mathcal{A}(x)$ and conditional distribution of $\mathcal{A}(x)$ under different occurrences, respectively.

CNN/Daily Mail domain. Overemphasizing β can diminish the contribution of these domain-specific terminology, potentially degrading performance. Hence, this may explain Figure 3 (a) in which as β further increases after 0.2, the detection error increases.

Effectiveness of Cross-Attention Hyper-parameter α. Comparing Figure 3 (c) and (d), we observe that the bi-grams in relevant documents tend to have larger average cross-attentions than the irrelevant counterparts. To amplify the discrepancy between the cross-attentions of irrelevant and relevant bi-grams, an optimal choice of α is required. To see this, given the cross-attention scores of a relevant bi-gram a_1 and an irrelevant bi-gram a_2, with $0 < a_2 < a_1 < 1$, the difference in the powered cross-attention scores, $a_1^\alpha - a_2^\alpha$, can be maximized by selecting $\alpha^* = \frac{\ln|\ln a_1| - \ln|\ln a_2|}{\ln a_1 - \ln a_2} > 0$. The difference escalates when $\alpha < \alpha^*$ and contracts when $\alpha > \alpha^*$. This observation aligns with Figure 3 (a), where detection error initially diminishes with increasing α up to 0.2, and subsequently rises for all β choices.

Effectiveness of Irrelevant Documents in Pretraining. We employed the T5-Large architecture for pretraining on the Delve dataset, deliberately excluding all in-domain irrelevant documents. Comprehensive pretraining results can be found in Appendix A.9.1. Subsequent deployment of CODE on this model yielded an 80.45% FPR at 95% TPR on the Delve detection dataset. This starkly contrasts with the 5.8% FPR achieved when irrelevant documents were incorporated during pretraining. To understand the discrepancy in detection performance, we juxtapose the cross-attention distributions from Figure 3 (e) and (f) against those from Figure 3 (c) and (d). Our observations underscore that incorporating irrelevant documents during pretraining can efficaciously diminish the cross-attention scores of irrelevant bi-grams (i.e., comparing Figure 3 (f) to (d)), without impinging on the scores of relevant bi-grams (i.e., comparing Figure 3 (e) to (c)). A more detailed case study can be found in the Appendix A.9.2, where we find that including irrelevant documents in the pretraining can even improve the attention scores of rare bi-grams in relevant documents, and reduce the scores of rare bi-grams in irrelevant documents and domain-agnostic phrases.

8 Conclusions

In this paper, we reveal the intrinsic ability of text summarizers for irrelevant document detection. By exploiting the cross-attention mechanism and unique behaviors of infrequent words, we introduced CODE, a novel and efficient irrelevant document detector. Experimental results validate the superiority of CODE over the traditional supervised fine-tuning methods under in-domain and cross-domain detection. Our findings illuminate the potential of harnessing cross-attention distribution, word frequency nuances and the strategic use of in-domain irrelevant documents in the pretraining phase, setting a promising direction for future advancements in the RAG.

Limitations

Although the cross-attention mechanism in generative models based on the encoder-decoder architecture can be used to construct well-performing irrelevant document detectors, it remains to be further explored whether the self-attention mechanism within generative models based on the decoder-only architecture can be used to construct efficient irrelevant document detectors. Additionally, due to the input sequence length limitations of models such as BART and T5, the performance of irrelevant document detection among a larger number of documents still requires further investigation.

References

Uchenna Akujuobi and Xiangliang Zhang. 2017. Delve: a dataset-driven scholarly search and analysis system. *ACM SIGKDD Explorations Newsletter*, 19(2):36–46.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. *arXiv preprint arXiv:2401.05856*.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. 2024. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems*, 36.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Nitay Calderon, Naveh Porat, Eyal Ben-David, Alexander Chapanin, Zorik Gekhman, Nadav Oved, Vitaly Shalumov, and Roi Reichart. 2024. Measuring the robustness of nlp models to domain shifts. *Preprint*, arXiv:2306.00168.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. Capturing relations between scientific papers: An abstractive model for related work section generation. Association for Computational Linguistics.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Lu Wang, and Arman Cohan. 2022. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. *arXiv preprint arXiv:2212.10526*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Kung-Hsiang Huang, ChengXiang Zhai, and Heng Ji. 2022. Concrete: Improving cross-lingual fact-checking with cross-lingual retrieval. *arXiv preprint arXiv:2209.02071*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

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. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. 2019. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Junting Pan, Ziyi Lin, Yuying Ge, Xiatian Zhu, Renrui Zhang, Yi Wang, Yu Qiao, and Hongsheng Li. 2023. Retrieving-to-answer: Zero-shot video question answering with frozen large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 272–283.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Rita Ramos, Desmond Elliott, and Bruno Martins. 2023. Retrieval-augmented image captioning. *arXiv preprint arXiv:2302.08268*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stephen E Robertson and Steve Walker. 1997. On relevance weights with little relevance information. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 16–24.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations*, pages 87–92.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. *arXiv preprint arXiv:2402.17497*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Yi Yuan, Haohe Liu, Xubo Liu, Qiushi Huang, Mark D Plumbley, and Wenwu Wang. 2024. Retrieval-augmented text-to-audio generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 581–585. IEEE.

A Appendix

A.1 Supplementary Materials for Datasets

A.1.1 Detailed Construction Method of Each Pretraining Dataset

In this subsection, we introduce the construction details of pretraining datasets CNN/Daily Mail-PT, SAMSum-PT, Delve-PT, and S2orc-PT in detail.

CNN/Daily Mail-PT. For the limitation of model input length, we use samples whose source document length is less than five hundred words as samples to be injected. We split the source document in these samples into two relevant chunks.

Table 3: Additional statistics of the pretraining datasets with in-domain irrelevant document.

		# Examples	# Words (single)	# Words (all)
CNN/Daily Mail-PT	Relevant Document	105,178	avg: 204.39, std: 69.37	231,462
	Irrelevant Document	97,042	avg: 243.56, std: 17.56	255,975
	Summary	52,459	avg: 47.78, std: 21.13	85,486
SAMSum-PT	Relevant Document	8,105	avg: 60.81, std: 47.47	16,947
	Irrelevant Document	5,186	avg: 62.26, std: 49.60	13,423
	Summary	4,092	avg: 23.53, std: 12.75	8,731
Delve-PT	Relevant Document	14,261	avg: 170.81, std: 86.63	52,318
	Irrelevant Document	20,000	avg: 175.66, std: 114.74	73,732
	Summary	10,000	avg: 30.82, std: 15.71	19,667
S2orc-PT	Relevant Document	37,589	avg: 221.39, std: 178.00	113,254
	Irrelevant Document	48,000	avg: 213.80, std: 167.73	135,606
	Summary	24,000	avg: 34.72, std: 18.64	42,019

Table 4: Additional statistics of the in-domain irrelevant document detection datasets.

	Document	# Examples	# Words (single)	# Words (all)
CNN/Daily Mail-ID	Relevant	49,557	avg: 197.94, std: 68.93	148,119
	Irrelevant	48,664	avg: 243.48, std: 17.74	176,268
SAMSum-ID	Relevant	8,117	avg: 61.08, std: 48.22	16,982
	Irrelevant	5,177	avg: 63.62, std: 50.53	13,890
Delve-ID	Relevant	14,839	avg: 170.26, std: 81.93	53,356
	Irrelevant	20,200	avg: 175.56, std: 97.47	74,912
S2orc-ID	Relevant	7,767	avg: 221.15, std: 189.84	48,232
	Irrelevant	8,400	avg: 212.79, std: 165.08	53,936

We split the source documents in the remaining samples into multiple chunks and collected them as candidate irrelevant chunks. For each sample to be injected, we randomly select two irrelevant chunks to insert.

SAMSum-PT. We divide the dataset into two parts at a ratio of 1:1, one part is prepared to be injected and the other part is used to provide irrelevant chunks. For the samples to be inserted, we also split the source document into two relevant chunks. We split the input document in another part of the samples into two chunks. We collect these chunks as candidate irrelevant chunks. For each sample to be injected, we randomly select two irrelevant chunks for insertion.

Delve-PT and S2orc-PT. We view the citation markers in the summaries to find relevant abstracts and irrelevant abstracts. Specifically, we select summaries with at least two citation markers. We randomly select two markers when a summary contains multiple citation markers. Next, for each citation marker in a summary, we find the corresponding paper abstracts as relevant documents. To get irrelevant abstracts, we use Microsoft Academic Graph (MAG) (Shen et al., 2018) to determine the academic fields where the abstract belongs. For each abstract, MAG directly provides their academic fields in a hierarchical manner with a progressively finer granularity from L0 to L5. To get the irrelevant abstracts, under L3 and more specific sub-fields, we select abstracts whose fields do not intersect with relevant abstracts. We also insert two relevant abstracts into each sample.

Table 5: Additional statistics of the cross-domain irrelevant document detection test sets. A ← B means sampling the irrelevant documents from dataset B and inserting them into dataset A.

	Document	# Examples	# Words (single)	# Words (all)
CNN/Daily Mail ← SAMSum	Relevant	4,978	avg: 198.13, std: 69.76	44,682
	Irrelevant	517	avg: 62.05, std: 47.90	3,631
CNN/Daily Mail ← Delve	Relevant	4,978	avg: 198.13, std: 69.76	44,682
	Irrelevant	1,839	avg: 174.39, std: 99.39	19,185
CNN/Daily Mail ← S2orc	Relevant	4,978	avg: 198.13, std: 69.76	44,682
	Irrelevant	2,838	avg: 212.56, std: 159.84	30,116
CNN/Daily Mail ← Random domain	Relevant	4,978	avg: 198.13, std: 69.76	44,682
	Irrelevant	3,953	avg: 151.77, std: 29.58	269,393
SAMSum ← CNN/Daily Mail	Relevant	816	avg: 61.76, std: 46.67	4,582
	Irrelevant	765	avg: 244.36, std: 17.01	19,270
SAMSum ← Delve	Relevant	816	avg: 61.76, std: 46.67	4,582
	Irrelevant	672	avg: 169.83, std: 88.16	10,658
SAMSum ← S2orc	Relevant	816	avg: 61.76, std: 46.67	4,582
	Irrelevant	725	avg: 223.35, std: 186.49	15,135
SAMSum ← Random domain	Relevant	816	avg: 61.76, std: 46.67	4,582
	Irrelevant	791	avg: 151.19, std: 29.43	97,565
Delve ← CNN/Daily Mail	Relevant	1,898	avg: 165.48, std: 74.64	15,953
	Irrelevant	1,640	avg: 243.86, std: 18.04	29,370
Delve ← SAMSum	Relevant	1,898	avg: 165.48, std: 74.64	15,953
	Irrelevant	507	avg: 61.97, std: 48.11	3,605
Delve ← S2orc	Relevant	1,898	avg: 165.48, std: 74.64	15,953
	Irrelevant	1,570	avg: 207.44, std: 140.61	21,830
Delve ← Random domain	Relevant	1,898	avg: 165.48, std: 74.64	15,953
	Irrelevant	1,796	avg: 151.53, std: 29.23	178,605
S2orc ← CNN/Daily Mail	Relevant	3,829	avg: 224.92, std: 209.54	33,485
	Irrelevant	2,742	avg: 243.69, std: 17.40	38,990
S2orc ← SAMSum	Relevant	3,829	avg: 224.92, std: 209.54	33,485
	Irrelevant	517	avg: 62.05, std: 47.90	3,631
S2orc ← Delve	Relevant	3,829	avg: 224.92, std: 209.54	33,485
	Irrelevant	18,382	avg: 173.56, std: 100.11	18,382
S2orc ← Random domain	Relevant	3,829	avg: 224.92, std: 209.54	33,485
	Irrelevant	3,246	avg: 150.81, std: 29.44	247,530

A.1.2 Additional Dataset Statistics

In this subsection, we report the statistics of the pretraining datasets, the in-domain irrelevant document detection dataset, and the test sets of cross-domain irrelevant document detection. These statistics are presented in Tables 3, 4 and 5, respectively.

A.2 Supplementary Materials for Experimental Setups

A.2.1 Pretraining Setups

In this subsection, we report the pretraining hyper-parameter settings in Table 6.

Table 6: Pretraining settings of the GLMs. Characters "B" and "L" denote the model size of Base and Large, respectively. All models are trained on the Tesla A100 machine. We set warm-up steps to 200 and employ a linear learning rate scheduler.

Datasets	Models	Learning rate	# Epochs	Batch size
CNN/Daily Mail-PT	BART-B	0.00003	15	8
	BART-L	0.00003	15	4
SAMSum-PT	BART-B	0.00003	15	8
	BART-L	0.00003	15	4
Delve-PT	BART-B	0.00003	15	16
	BART-L	0.00003	15	8
S2orc-PT	BART-B	0.00003	15	8
	BART-L	0.00003	15	8
CNN/Daily Mail-PT	T5-B	0.0002	15	6
	T5-L	0.0001	15	6
SAMSum-PT	T5-B	0.0002	15	6
	T5-L	0.0001	15	6
Delve-PT	T5-B	0.0002	15	6
	T5-L	0.0001	15	6
S2orc-PT	T5-B	0.0002	15	12
	T5-L	0.0001	15	6

Table 7: Performance of the pretrained models

Datasets	Models	ROUGE-1	ROUGE-2	ROUGE-L
Delve-PT	T5-L	19.3443	3.3781	14.4185
	T5-B	17.5721	2.8855	13.4359
	BART-L	18.0474	2.7043	13.6427
	BART-B	18.3348	2.8605	13.9695
S2orc-PT	T5-L	20.4524	3.9853	15.1929
	T5-B	19.9058	3.6515	14.7904
	BART-L	20.7972	3.7129	15.4441
	BART-B	19.9070	3.4996	14.8250
SAMSum-PT	T5-L	44.3738	21.7557	38.7138
	T5-B	43.1620	20.6720	38.6918
	BART-L	50.4676	25.7701	41.8661
	BART-B	44.9713	20.4162	36.2211
CNN/Daily Mail-PT	T5-L	35.5728	12.0295	25.0173
	T5-B	33.7640	14.7571	23.3762
	BART-L	41.8007	20.1378	30.1265
	BART-B	41.4113	19.7040	29.7622

A.2.2 Performance of the Pretrained Models

In this subsection, we show the performance of text summarization on each dataset and pretrained model in Table 7. We use ROUGE [3] to evaluate the

[3] https://github.com/google-research/google-research/tree/master/rouge

Table 8: Epochs and batch size of the Frozen. Characters “B" and “L" denote the model size of Base and Large, respectively. All models are trained on the Tesla A100 machine.

Datasets	Models	# Epochs	Batch size
CNN/Daily Mail-ID	BART-B	40	64
	BART-L	40	64
SAMSum-ID	BART-B	40	64
	BART-L	40	64
Delve-ID (1K)	BART-B	40	64
	BART-L	40	64
Delve-ID (8K)	BART-B	40	64
	BART-L	40	64
S2orc-ID	BART-B	40	64
	BART-L	40	64
CNN/Daily Mail-ID	T5-B	40	64
	T5-L	40	64
SAMSum-ID	T5-B	40	64
	T5-L	40	64
Delve-ID (1K)	T5-B	40	64
	T5-L	40	64
Delve-ID (8K)	T5-B	40	64
	T5-L	40	64
S2orc-ID	T5-B	40	64
	T5-L	40	64

Table 9: Epochs and batch size of FT-ALL. Characters “B" and “L" denote the model size of Base and Large, respectively. All models are trained on the Tesla A100 machine.

Datasets	Models	# Epochs	Batch size
CNN/Daily Mail	BART-B	10	8
	BART-L	10	8
SAMSum-ID	BART-B	10	8
	BART-L	10	8
Delve-ID (1K)	BART-B	10	8
	BART-L	10	8
Delve-ID (8K)	BART-B	10	8
	BART-L	10	8
S2orc-ID	BART-B	10	8
	BART-L	10	8
CNN/Daily Mail-ID	T5-B	10	8
	T5-L	10	8
SAMSum-ID	T5-B	10	8
	T5-L	10	8
Delve-ID (1K)	T5-B	10	4
	T5-L	10	4
Delve-ID (8K)	T5-B	10	4
	T5-L	10	4
S2orc-ID	T5-B	10	4
	T5-L	10	4

quality of text summarization and performance of all pretrained models.

Additionally, the metrics used in this section are as follows:

- **ROUGE-1** measures the overlap of unigrams between the reference and the generated summary.
- **ROUGE-2** extends the concept of ROUGE-1 to bigrams, measuring the overlap of consecutive pairs of words between the reference and the generated summary.
- **ROUGE-L** calculates the longest common subsequence between the reference and the generated summary.

We also note here that on the CNN/Daily Mail dataset, the reference (Lewis et al., 2019) reports 44.16, 21.28, and 40.90 on the BART model, and the reference (Raffel et al., 2020) reports 43.52, 21.55 and 40.69 on T5 model, respectively. Our pretrained model generally has worse performance, since (1) we add the irrelevant documents in the pretrained phrase; (2) For each original dataset, a portion is used to construct the irrelevant document detection dataset. Therefore, the total amount of pretraining data is smaller than the original dataset, which may lead to a worse performance of text summarization. Although the performance of our pretraining model is worse, this does not affect the effectiveness of irrelevant document detection.

A.2.3 Training Setups of the Baselines

In this subsection, we report the training settings of the Frozen and FT-ALL. Table 8 and Table 9 present the training epochs and batch sizes.

Frozen. We use the AdamW optimizer with exponential decay rates for the first and second moments of the gradient updates setting to 0.9 and 0.999, respectively. We choose a constant learning rate scheduler with a warm-up period of 200 steps. The learning rates are selected from the set $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. The weight decay parameter is configured to be 0.0001. For each hyperparameter setting, we run three times with different random seeds. In the main paper, we report the mean value of the results, while the standard deviations are presented in Table 11. We select the model with the lowest validation loss for testing in

Table 10: Evaluation results of CODE and baselines for in-domain irrelevant document detection. ↑ indicates that larger values are better, and ↓ indicates that smaller values are better. Characters “B" and “L" denote the Base and Large model, respectively.

	Models	FPR (95%) TPR ↓	AUROC ↑	AUPR ↑
			CODE/Frozen/FT-ALL	
Delve-ID (1K)	T5-L	**5.80**/30.30/25.63	**98.08**/92.87/94.59	**97.03**/93.57/92.60
	T5-B	**32.30**/65.97/57.75	**90.08**/84.52/85.21	**83.76**/82.62/82.92
	BART-L	**11.10**/43.02/44.45	**96.09**/91.23/91.84	**93.41**/90.08/90.47
	BART-B	**19.65**/49.27/53.02	**91.60**/90.62/90.99	**93.66**/90.23/90.61
Delve-ID (8K)	T5-L	**5.55**/16.85/18.28	**98.16**/93.62/95.87	**97.23**/94.01/95.18
	T5-B	**31.50**/60.22/47.98	**90.36**/86.32/87.64	**84.34**/85.40/87.49
	BART-L	**11.10**/33.52/33.45	**96.09**/93.17/92.75	**93.41**/92.96/91.61
	BART-B	**20.30**/45.40/38.00	**94.79**/90.66/92.04	**91.30**/89.98/90.95
S2orc-ID	T5-L	**1.08**/10.40/6.05	**99.54**/96.01/97.69	**99.27**/95.59/97.32
	T5-B	**2.53**/15.82/11.65	**99.00**/96.68/96.87	**97.95**/96.51/96.01
	BART-L	**4.83**/16.18/9.47	**98.66**/96.03/96.77	**98.11**/95.45/96.15
	BART-B	**3.00**/6.94/5.07	**98.72**/97.91/97.71	**97.56**/97.55/97.26
SAMSum-ID	T5-L	**0.60**/5.50/0.65	**99.87**/98.67/99.68	**99.87**/98.78/98.60
	T5-B	**0.61**/8.44/1.22	**99.66**/99.21/97.46	**99.43**/99.00/96.68
	BART-L	**0.91**/0.65/0.28	99.43/99.70/**99.77**	99.37/99.67/**99.77**
	BART-B	**2.26**/3.83/3.67	97.23/99.15/**97.83**	94.61/**99.18**/97.83
CNN/Daily Mail-ID	T5-L	**0.00**/0.20/0.32	**99.99**/99.85/99.77	**99.99**/99.81/99.79
	T5-B	**0.12**/0.82/0.29	**99.96**/99.62/99.80	**99.96**/99.56/99.70
	BART-L	**0.14**/0.57/0.44	99.71/99.69/**99.78**	99.60/99.73/**99.75**
	BART-B	**0.18**/0.23/0.33	**99.89**/99.87/99.86	99.83/99.86/**99.86**

irrelevant document detection.

FT-ALL. We utilize the same hyper-parameter setting used in the baseline Frozen, except that the learning rate is set to the one used in the summarizer pretraining. We repeat this baseline three times with different random seeds.

A.3 Supplementary Results in In-domain Irrelevant Document Detection

In this section, we present all evaluation results of in-domain detection to show the improvement of our method compared to the baselines. Table 10 shows the performance of our proposed method and two baselines under each dataset. The details of our method and the baselines can be found in section 4. We note here that our method is deterministic and does not have an error bar. The other two baselines are randomly re-initialized with three different seeds. We take the average of the results as the final performance and calculate the standard deviation. Table 11 provides the standard deviation for different models. Table 12 provides the hyper-parameters α and β of CODE are used in the evaluation process.

The evaluation metrics used in section 6 are as follows:

- **FPR at 95% TPR** refers to the rate that a relevant document is misclassified as an irrelevant document when the true positive rate (TPR) is at 95%.
- **AUROC** is calculated as the Area Under the Receiver Operating Characteristic curve (Fawcett, 2006). The ROC curve illustrates the relationship between TPR and FPR at various thresholds. The higher the value of AUROC, the stronger the discriminative ability of the model.
- **AUPR** stands for Area Under the Precision-

Table 11: Standard deviation of the evaluation results.

	Models	FPR (95%) TPR ↓	AUROC ↑	AUPR ↑
		CODE/Frozen/FT		
Delve (1K)	T5-L	0.00 /0.94/1.34	0.00/0.21/0.91	0.00/0.16/0.76
	T5-B	0.00/1.53/7.42	0.00/0.20/9.83	0.00/0.16/12.46
	BART-L	0.00/1.17/2.49	0.00/0.19/0.39	0.00/0.20/0.40
	BART-B	0.00/1.42/0.34	0.00/0.13/0.06	0.00/0.21/0.10
Delve (8K)	T5-L	0.00/0.62/1.05	0.00/0.09/0.08	0.00/0.11/0.34
	T5-B	0.00/1.08/0.55	0.00/0.13/1.12	0.00/0.15/0.92
	BART-L	0.00/0.98/2.45	0.00/0.02/0.24	0.00/0.03/0.40
	BART-B	0.00/1.18/0.76	0.00/0.45/0.20	0.00/0.62/0.34
S2orc	T5-L	0.00/0.35/0.31	0.00/0.27/0.93	0.00/0.33/0.86
	T5-B	0.00/0.48/0.35	0.00/0.11/3.02	0.00/0.48/4.93
	BART-L	0.00/0.01/1.04	0.00/0.01/0.11	0.00/0.01/0.13
	BART-B	0.00/0.23/0.25	0.00/0.01/0.25	0.00/0.01/0.64
SAMSum	T5-L	0.00/0.46/0.24	0.00/0.03/0.01	0.00/0.04/0.02
	T5-B	0.00/0.43/0.32	0.00/0.02/0.01	0.00/0.03/0.03
	BART-L	0.00/0.11/0.06	0.00/0.01/0.02	0.00/0.01/0.01
	BART-B	0.00/0.12/0.46	0.00/0.05/0.05	0.00/0.06/0.21
CNN/Daily Mail	T5-L	0.00/0.01/0.00	0.00/0.00/0.00	0.00/0.02/0.00
	T5-B	0.00/0.01/0.01	0.00/0.01/0.00	0.00/0.00/0.01
	BART-L	0.00/0.06/0.10	0.00/0.01/0.02	0.00/0.01/0.01
	BART-B	0.00/0.02/0.46	0.00/0.01/0.05	0.00/0.01/0.21

Table 12: The hyper-parameters α and β of CODE are used in the main results. Characters “B" and “L" denote the model size of Base and Large, respectively.

	BART-B	BART-L	T5-B	T5-L
	α, β			
CNN/Daily Mail-ID	0.2, 0.0	0.2, 0.3	0.2, 0.1	0.2, 0.1
SAMSum-ID	0.2, 0.0	0.2, 0.0	0.4, 0.2	0.4, 0.4
Delve-ID (1K)	1.2, 0.2	0.2, 0.1	1.2, 0.0	0.2, 0.0
Delve-ID (8K)	0.8, 0.0	1.0, 0.1	1.0, 0.2	0.6, 0.1
S2orc-ID	0.6, 0.1	1.0, 0.1	0.6, 0.0	0.4, 0.0

Recall curve (Manning and Schutze, 1999; Saito and Rehmsmeier, 2015). The PR curve depicts the trade-off between precision and recall at various thresholds. For an ideal classifier, its AUPR score is 1.

A.4 Performance vs. Pretrained Model Checkpoints

In this section, we show how the selection of checkpoints of the pretrained model affects the detection performance of our method. Specifically, we present the relationship between the validation loss for each checkpoint on the pretrained dataset and their in-domain irrelevant document detection performance. Each figure in this section displays the validation loss and FPR at 95% TPR metric of each dataset and model at different checkpoints. We find out that the pretrained model with the smallest validation loss is generally not the pretrained model with the best detection performance, but the detection performance difference between

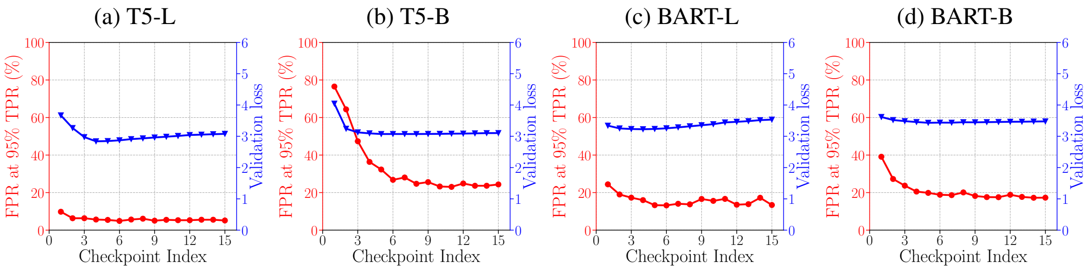

Figure 4: Performance vs. Checkpoints on Delve-ID (1K)

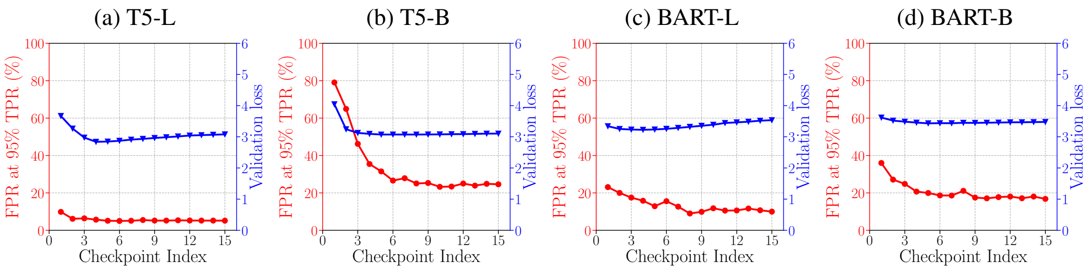

Figure 5: Performance vs. Checkpoints on Delve-ID (8K).

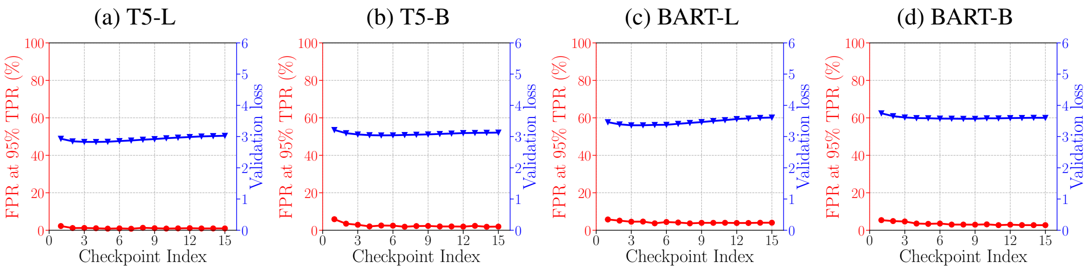

Figure 6: Performance vs. Checkpoints on S2orc-ID

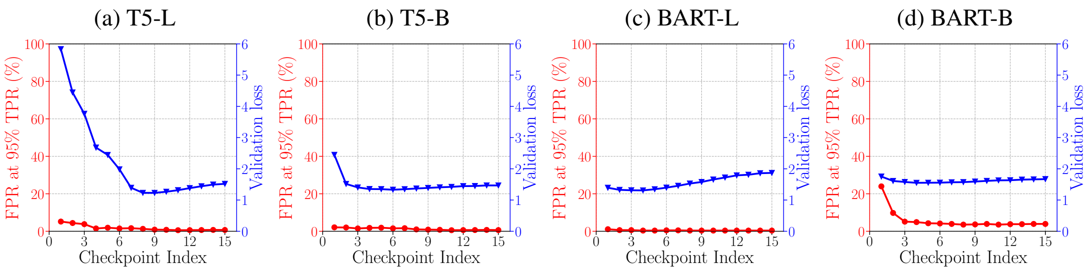

Figure 7: Performance vs. Checkpoints on SAMSum-ID

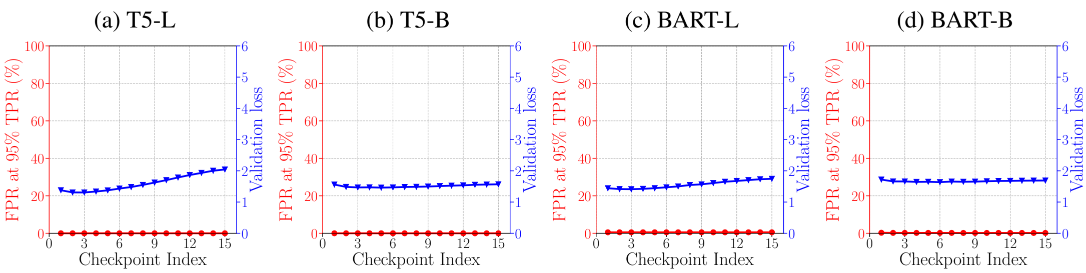

Figure 8: Performance vs. Checkpoints on CNN/Daily Mail-ID

the pretrained model with the smallest validation loss and the pretrained model with the best irrelevant document detection performance is negligible.

The correspondence between the figures and the setting is as follows:

- Figure 4: performance on Delve-ID (1K) dataset and four models.
- Figure 5: performance on Delve-ID (8K) dataset and four models.
- Figure 6: performance on S2orc-ID dataset and four models.
- Figure 7: performance on SAMSum-ID dataset and four models.
- Figure 8: performance on CNN/Daily Mail-ID dataset and four models.

A.5 Performance vs. Pretrained Model Attention Layers

In this section, we show how different attention layers affect the irrelevant document detection performance of our method. Specifically, we present the relationship between the attention layer and two evaluation metrics of irrelevant document detection. Each figure in this section displays FPR at 95% TPR and AUROC of our method on each dataset and model when different attention layers are selected. We observe that the lowest FPR at 95% TPR and the highest AUROC occur in the attention layer close to the last layer (the layer closest to the output layer) for most types of models and datasets, except BART-base, which contains only six attention layers. In fact, we can also observe that the last three layers have similar performance and this indicates that the performance varies small if the attention layers close to the output layer are selected.

The correspondence between the figures and the setting is as follows:

- Figure 9: performance on Delve-ID (1K) dataset and each model.
- Figure 10: performance on Delve-ID (8K) dataset and each model.
- Figure 11: performance on S2orc-ID dataset and each model.
- Figure 12: performance on SAMSum-ID dataset and each model.
- Figure 13: performance on CNN/Daily Mail-ID dataset and each model.

A.6 Performance vs. In-domain Irrelevant Detection Difficulty

In this section, we show how different dataset affects the in-domain irrelevant document detection performance of our method. We present the relationship between the dataset similarity and two evaluation metrics of irrelevant document detection. Figure 14 displays how FPR at 95% TPR changes with the improvement of dataset similarity, while Figure 15 displays how AUROC changes with the improvement of dataset difficulty. $\mathcal{C}_1$ to $\mathcal{C}_5$ represent CNN/Daily Mail-ID, S2orc-ID, SAMSum-ID, Delve-ID (8K), and Delve-ID (1K), respectively.

To measure the similarity of the dataset, we use the Sentence-BERT model to obtain the embedding of input documents and calculate the average cosine similarity between the embedding of relevant and irrelevant documents within a single data sample. Specifically, each data sample contains two relevant documents and two irrelevant documents. For each document X in the dataset $\mathcal{C}$, we use $H(X)$ to denote the embedding vector of document X, $\mathcal{X}^{\text{irr}} \subset \mathcal{X}$ is the set of irrelevant documents in the input document sequence. Therefore, the difficulty of the dataset $\mathcal{C}$ is defined as:

$$\text{sim}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\mathcal{X} \in \mathcal{C}} \left[\frac{1}{|\mathcal{X}^{\text{irr}}|(|\mathcal{X}| - |\mathcal{X}^{\text{irr}}|)} \sum_{X \in \mathcal{X}^{\text{irr}}} \sum_{X' \in \mathcal{X} \setminus \mathcal{X}^{\text{irr}}} \frac{\langle H(X), H(X') \rangle}{\|H(X)\|_2 \cdot \|H(X')\|_2} \right]$$

The higher the cosine similarity, the smaller the difference between relevant and irrelevant documents in the dataset, indicating it is harder to detect irrelevant documents on this dataset. We observe that when the relevant and irrelevant documents in the dataset tend to be less similar to each other (i.e., the similarity of the dataset is smaller), our method tends to have a smaller FPR at 95% TPR and a larger AUROC.

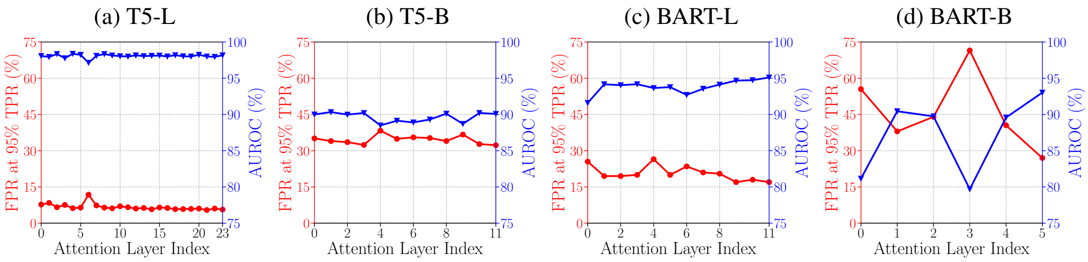

Figure 9: Performance vs. Attention Layers on Delve-ID (1K)

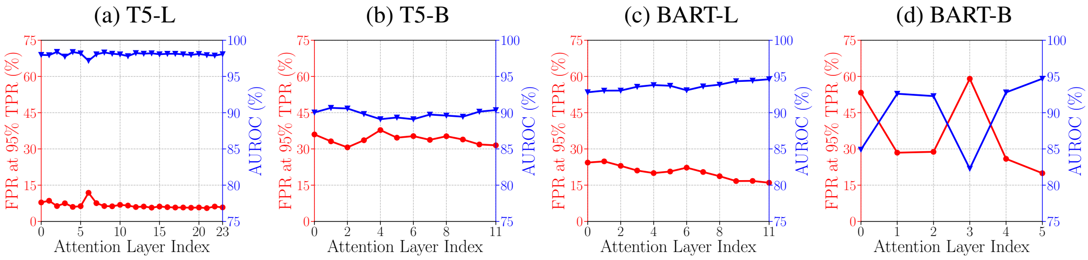

Figure 10: Performance vs. Attention Layers on Delve-ID (8K)

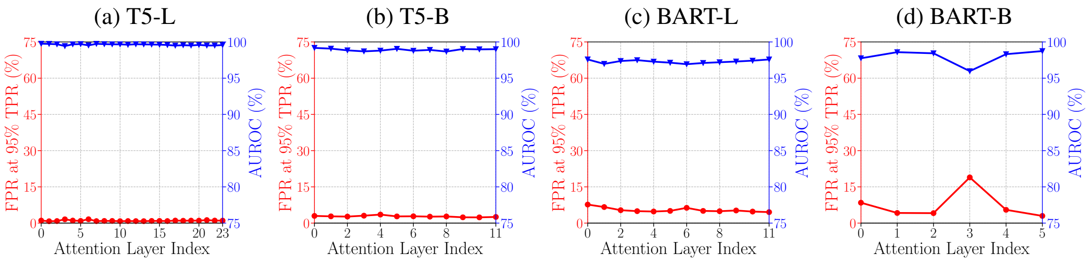

Figure 11: Performance vs. Attention Layers on S2orc-ID

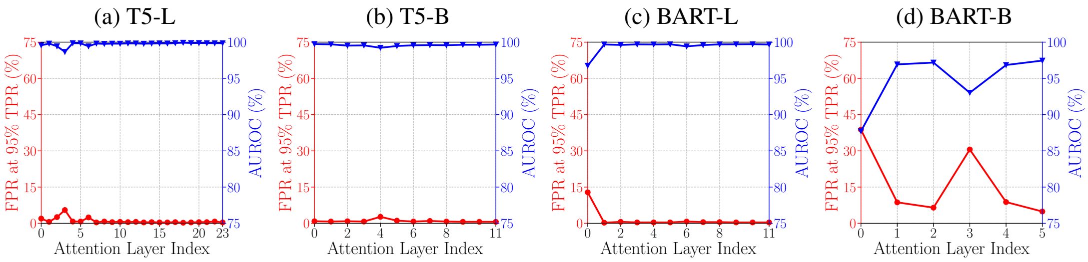

Figure 12: Performance vs. Attention Layers on SAMSum-ID

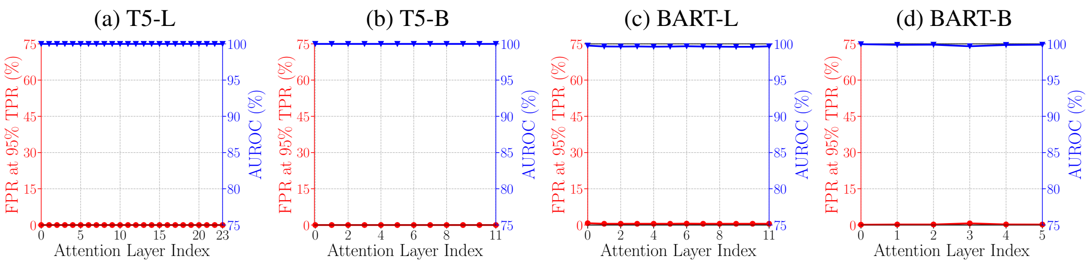

Figure 13: Performance vs. Attention Layers on CNN/Daily Mail-ID

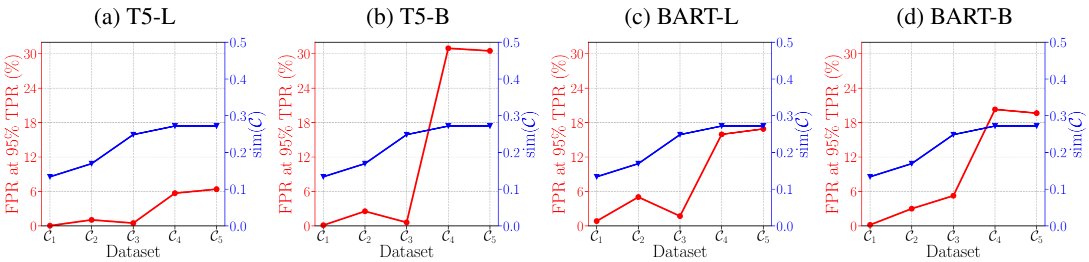

Figure 14: FPR at 95% TPR vs. sim($\mathcal{C}$) in in-domain irrelevant document detection.

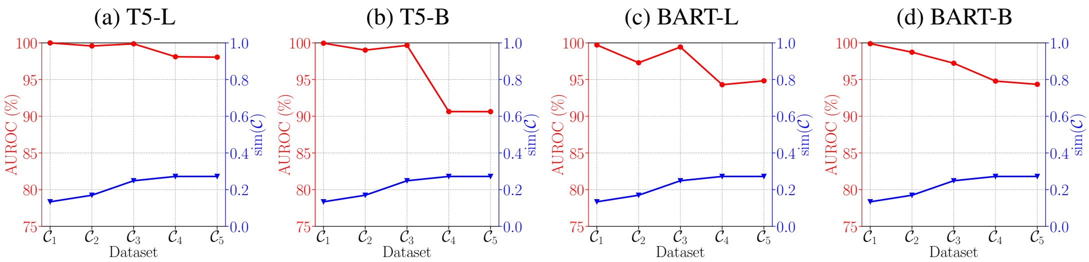

Figure 15: AUROC vs. sim($\mathcal{C}$) in in-domain irrelevant document detection.

A.7 Performance vs. Cross-domain Irrelevant Detection

In this section, we show how our method transfers across different domains. Recall that we pretrain the generative language model, find the best hyper-parameter setting, and test the detection performance on the same domain. We hope that this pretrained model together with the best hyper-parameter setting can also transfer to other domains. Therefore, we constructed cross-domain test sets to evaluate the cross-domain performance. The details of the cross-domain dataset can be found in section 5.3, A.1.2, and we use equation (2) to measure the difficulty of cross-domain datasets.

A.7.1 Results of Cross-domain Irrelevant Detection

Table 13 and Table 14 show the performance of our proposed method and two baselines under each dataset in cross-domain detection. Table 15 and Table 16 provides the standard deviation for different models.

A.7.2 Performance vs. Cross-domain Irrelevant Detection Difficulty

We present the relationship between cross-domain dataset similarity and two evaluation metrics of the irrelevant document detection. Figure 16, 17, 18, 19 display FPR at 95% TPR, while Figure 20, 21, 22, 23 display AUROC on each model and dataset.

From the figures, we observe that for most settings, FPR at 95% TPR decreases, and AUROC increases as the similarity of the dataset increases, except for one case. In Figure 17d, we observe although the S2orc ← Random domain has a smaller difficulty, FPR is two times larger than that of S2orc ← Delve domain. The performance on the AUROC metric is also worse than that of S2orc ← Delve domain in Figure 21d. We generally observe this on the smaller model, i.e., BART-Base, consisting of nearly 140M parameters. On the larger model, we do not observe this. This may be due to the fact that the large model models tend to perform better for cross-domain data. We also observe that T5 model generally performs better than BART on most cross-domain datasets. We also observe that the larger models yield better performance for both BART and T5.

Table 13: Evaluation results of CODE and baselines for cross-domain irrelevant document detection. A ← B means sampling the irrelevant documents from dataset B and inserting them into dataset A. ↑ indicates that larger values are better, and ↓ indicates that smaller values are better. Characters “B" and “L" denote the Base and Large model, respectively.

	Models	**FPR (95%) TPR** ↓	**AUROC** ↑	**AUPR** ↑
		CODE/Frozen/FT-ALL		
Delve ← S2orc	T5-L	**1.65**/27.13/8.12	**99.55**/95.39/97.95	**99.52**/96.05/98.38
	T5-B	**4.75**/38.58/35.67	**98.74**/93.87/94.01	**98.25**/94.84/94.96
	BART-L	**3.00**/22.05/41.87	**99.11**/96.39/95.29	**98.85**/96.80/96.75
	BART-B	**5.45**/30.82/42.57	**98.36**/95.30/94.67	**97.73**/96.13/95.45
Delve ← Random domain	T5-L	**0.10**/58.27/10.63	**99.96**/89.92/97.63	**99.96**/91.91/97.70
	T5-B	**0.00**/5.03/64.29	**99.99**/98.60/89.81	**99.99**/98.93/92.49
	BART-L	**0.00**/52.00/37.63	**99.99**/92.67/95.71	**99.99**/93.86/97.04
	BART-B	**2.60**/54.80/33.62	**99.23**/91.92/96.35	**99.18**/93.95/97.29
Delve ← SAMSum	T5-L	**0.05**/67.70/7.60	**99.95**/81.50/98.19	**99.95**/82.18/98.58
	T5-B	**0.00**/83.35/70.08	**99.93**/83.87/89.22	**99.94**/87.37/92.30
	BART-L	**0.00**/58.30/45.07	**99.99**/88.56/95.08	**99.99**/89.52/96.68
	BART-B	**0.10**/69.13/39.52	**99.96**/84.59/95.72	**99.96**/86.17/96.92
Delve ← CNN/Daily Mail	T5-L	**0.10**/34.30/10.03	**99.92**/93.87/97.46	**99.92**/94.77/97.46
	T5-B	**0.10**/59.85/64.34	**99.88**/90.99/90.32	**99.89**/93.01/92.97
	BART-L	**0.50**/53.40/35.82	**99.83**/88.63/95.98	**99.81**/89.13/97.19
	BART-B	**2.80**/42.77/37.05	**99.25**/92.87/96.01	**99.12**/94.10/97.11
S2orc ← Delve	T5-L	**1.10**/31.42/1.75	**99.71**/94.04/98.93	**99.71**/94.53/99.02
	T5-B	**1.70**/19.69/7.91	**99.47**/96.60/97.85	**99.34**/97.13/98.12
	BART-L	**4.47**/18.85/3.55	**98.25**/95.90/98.17	**97.47**/95.49/98.23
	BART-B	**4.20**/11.78/2.36	**98.79**/97.90/98.50	**98.67**/98.23/98.73
S2orc ← Random domain	T5-L	**0.00**/17.03/0.70	**99.99**/97.10/98.83	**99.99**/97.59/99.20
	T5-B	**0.00**/2.50/11.57	**99.99**/99.02/97.41	**99.99**/99.26/97.80
	BART-L	**0.30**/7.65/4.39	**99.93**/98.09/97.96	**99.93**/98.59/98.10
	BART-B	**2.35**/16.97/2.07	**98.13**/96.97/98.49	**98.32**/97.86/98.74
S2orc ← SAMSum	T5-L	**0.22**/14.66/1.12	**99.89**/97.14/99.04	**99.90**/97.24/99.14
	T5-B	**0.30**/15.97/9.91	**99.78**/97.15/97.51	**99.82**/97.73/97.78
	BART-L	**0.05**/3.15/0.68	**99.98**/99.19/98.78	**99.98**/99.31/99.14
	BART-B	**0.22**/7.74/0.62	**99.87**/98.47/98.80	**99.89**/98.72/ 99.16
S2orc ← CNN/Daily Mail	T5-L	**0.05**/6.08/1.44	**99.97**/98.61/98.95	**99.97**/98.73/99.02
	T5-B	**0.22**/16.24 /3.37	**99.86**/97.00/98.53	**99.88**/97.48/98.90
	BART-L	**0.43**/6.20/0.84	**99.84**/98.54/98.81	**99.75**/98.72/99.15
	BART-B	**0.40**/4.04/0.71	**99.70**/98.93/98.98	**99.61**/99.12/99.22

Table 14: Continuation of Table 13.

	Models	FPR (95%) TPR ↓	AUROC ↑	AUPR ↑
			CODE/Frozen/FT-ALL	
SAMSum ← Delve	T5-L	**0.00**/0.24/0.18	**99.98**/99.74/99.58	**99.98**/99.79/99.68
	T5-B	**1.22**/15.08/2.03	**99.77**/97.38/99.28	**99.76**/97.55/99.36
	BART-L	**0.00**/9.58/1.85	**99.99**/98.02/98.93	**99.99**/98.25/99.08
	BART-B	**0.37**/0.41/1.81	**99.82**/99.45/98.29	**99.81**/99.56/98.63
SAMSum ← S2orc	T5-L	**0.00**/0.04/0.42	**99.99**/99.79/99.62	**99.99**/99.83/99.64
	T5-B	**0.61**/7.74/2.34	**99.86**/98.49/99.21	**99.86**/98.54/99.30
	BART-L	**0.00**/21.84/0.85	**99.99**/96.16/99.34	**99.99**/96.50/99.45
	BART-B	**0.37**/0.65/1.52	**99.91**/99.29/98.49	**99.90**/99.44/98.84
SAMSum ← Random domain	T5-L	**0.00**/0.86/0.30	**99.99**/99.59/99.68	**99.99**/99.67/99.74
	T5-B	**0.00**/0.20/2.84	**99.99**/99.84/99.08	**99.99**/99.88/99.25
	BART-L	**0.00**/5.34/3.67	**99.99**/98.67/98.28	**99.99**/98.94/ 98.49
	BART-B	**0.49**/12.67/1.66	**99.83**/96.47/98.50	**99.83**/97.82/98.80
SAMSum ← CNN/Daily Mail	T5-L	**0.00**/1.75 /0.18	**99.99**/99.45/99.68	**99.99**/99.54/99.73
	T5-B	**0.73**/3.42/3.30	**99.88**/99.19/99.27	**99.88**/99.27/99.35
	BART-L	**0.00**/10.35/1.32	**99.99**/97.96/99.12	**99.99**/98.11/99.26
	BART-B	**1.59**/1.96/1.30	**99.48**/98.20/98.50	**99.32**/98.78/99.02
CNN/Daily Mail ← Delve	T5-L	**0.02**/0.33/1.35	**99.99**/99.79/99.23	**99.99**/99.83/98.91
	T5-B	**0.02**/3.79/23.15	**99.99**/99.09/83.10	**99.99**/99.21/71.82
	BART-L	**0.00**/27.87/ 73.88	**99.99**/88.16/60.47	**99.99**/82.24/59.74
	BART-B	**0.44**/23.67/25.92	**99.86**/88.75/79.19	**99.87**/85.69/67.94
CNN/Daily Mail ← S2orc	T5-L	**0.02**/0.37/2.12	**99.99**/99.79/98.94	**99.99**/99.82/98.39
	T5-B	**0.02**/5.17/9.64	**99.99**/98.96/93.28	**99.99**/99.08/86.11
	BART-L	**0.02**/23.23/63.04	**99.99**/86.37/65.03	**99.99**/75.31/60.56
	BART-B	**0.12**/21.50/33.20	**99.95**/87.56/73.51	**99.95**/81.92/63.02
CNN/Daily Mail ← Random domain	T5-L	**0.00**/ 0.09/1.60	**99.99**/99.67/99.11	**99.99**/99.75/98.72
	T5-B	**0.00**/16.51/6.48	**99.99**/97.28/95.40	**99.99**/98.03/90.04
	BART-L	**0.00**/1.49/42.90	**99.99**/99.15/76.33	**99.99**/99.26/69.34
	BART-B	**0.08**/0.00/1.03	**99.93**/99.86/99.58	**99.94**/99.91/99.58
CNN/Daily Mail ← SAMSum	T5-L	**0.02**/7.98/2.82	**99.99**/98.47/98.64	**99.99**/98.78/97.92
	T5-B	**0.50**/31.29/23.66	**99.87**/94.70/82.24	**99.87**/95.01/70.68
	BART-L	**0.04**/89.86/45.17	**99.98**/24.85/ 71.64	**99.97**/35.63/62.37
	BART-B	**3.40**/84.68/46.20	**99.28**/46.40/ 64.11	**99.35**/51.05/56.21

Table 15: Standard deviation of the evaluation results.

	Models	FPR (95%) TPR ↓	AUROC ↑	AUPR ↑
		CODE/Frozen/FT-ALL		
Delve ← S2orc	T5-L	0.00/1.64/1.40	0.00/0.25/0.33	0.00/0.19/0.23
	T5-B	0.00/1.41/4.03	0.00/0.12/0.55	0.00/0.06/0.31
	BART-L	0.00/1.82/4.17	0.00/0.12/0.45	0.00/0.23/0.27
	BART-B	0.00/2.29/2.23	0.00/0.26/0.41	0.00/0.17/0.35
Delve ← Random domain	T5-L	0.00/4.78/4.39	0.00/1.02/0.48	0.00/0.78/0.42
	T5-B	0.00/2.49/1.41	0.00/0.44/0.82	0.00/0.33/0.62
	BART-L	0.00/4.21/4.14	0.00/1.21/0.39	0.00/1.09/0.25
	BART-B	0.00/5.74/3.27	0.00/1.10/0.46	0.00/0.81/0.35
Delve ← SAMSum	T5-L	0.00/3.86/2.09	0.00/1.18/0.32	0.00/0.47/0.22
	T5-B	0.00/4.47/2.70	0.00/2.49/1.39	0.00/2.26/1.04
	BART-L	0.00/2.46/3.81	0.00/1.99/0.47	0.00/1.32/0.31
	BART-B	0.00/1.16/3.89	0.00/1.05/0.58	0.00/1.36/0.42
Delve ← CNN/Daily Mail	T5-L	0.00/4.53/1.57	0.00/0.80/0.32	0.00/0.62/0.47
	T5-B	0.00/6.95/2.16	0.00/1.29/1.03	0.00/1.19/0.80
	BART-L	0.00/5.12/4.80	0.00/1.47/0.47	0.00/1.28/0.31
	BART-B	0.00/5.06/1.56	0.00/1.15/0.31	0.00/0.93/0.28
S2orc ← Delve	T5-L	0.00/1.19/0.16	0.00/0.39/0.19	0.00/0.38/0.17
	T5-B	0.00/3.73/1.30	0.00/0.51/0.23	0.00/0.43/0.21
	BART-L	0.00/1.05/0.53	0.00/0.21/0.25	0.00/0.10/0.28
	BART-B	0.00/0.37/0.21	0.00/0.09/0.19	0.00/0.08/0.31
S2orc ← Random domain	T5-L	0.00/0.29/0.38	0.00/0.06/0.61	0.00/0.05/0.38
	T5-B	0.00/1.46/2.15	0.00/0.34/0.36	0.00/0.23/0.33
	BART-L	0.00/2.01/1.91	0.00/0.62/0.51	0.00/0.31/0.55
	BART-B	0.00/3.98/0.30	0.00/0.52/0.19	0.00/0.35/0.32
S2orc ← SAMSum	T5-L	0.00/2.21/0.12	0.00/0.44/0.25	0.00/0.63/0.19
	T5-B	0.00/1.85/1.63	0.00/0.30/0.30	0.00/0.23/0.30
	BART-L	0.00/1.97/0.55	0.00/0.31/0.22	0.00/0.29/0.16
	BART-B	0.00/1.78/0.26	0.00/0.30/0.22	0.00/0.25/0.16
S2orc ← CNN/Daily Mail	T5-L	0.00/0.89/0.13	0.00/0.13/0.23	0.00/0.12/0.18
	T5-B	0.00/2.19/0.22	0.00/0.41/0.13	0.00/0.40/0.07
	BART-L	0.00/1.42/0.56	0.00/0.27/0.39	0.00/0.20/0.25
	BART-B	0.00/0.90/0.31	0.00/0.15/0.11	0.00/0.13/0.14

Table 16: Continuation of Table 15.

	Models	FPR (95%) TPR ↓	AUROC ↑	AUPR ↑
			CODE/Frozen/FT-ALL	
SAMSum ← Delve	T5-L	0.00/0.17/0.06	0.00/0.02/0.03	0.00/0.01/0.02
	T5-B	0.00/2.97/0.12	0.00/0.46/0.02	0.00/0.50/0.02
	BART-L	0.00/0.47/1.84	0.00/0.25/0.38	0.00/0.27/0.54
	BART-B	0.00/0.16/0.34	0.00/0.10/0.19	0.00/0.07/0.13
SAMSum ← S2orc	T5-L	0.00/0.06/0.15	0.00/0.06/0.02	0.00/0.01/0.02
	T5-B	0.00/1.55/0.17	0.00/0.33/0.04	0.00/0.35/0.04
	BART-L	0.00/4.53/0.73	0.00/0.99/0.13	0.00/1.00/0.21
	BART-B	0.00/0.49/0.56	0.00/0.20/0.26	0.00/0.14/0.13
SAMSum ← Random domain	T5-L	0.00/0.17/0.15	0.00/0.03/0.02	0.00/0.03/0.03
	T5-B	0.00/0.06/0.22	0.00/0.03/0.02	0.00/0.02/0.02
	BART-L	0.00/1.51/3.62	0.00/0.37/0.82	0.00/0.29/1.09
	BART-B	0.00/5.29/0.55	0.00/0.52/0.26	0.00/0.32/0.13
SAMSum ← CNN/Daily Mail	T5-L	0.00/0.47/0.06	0.00/0.07/0.03	0.00/0.06/0.03
	T5-B	0.00/0.78/0.24	0.00/0.08/0.02	0.00/0.07/0.02
	BART-L	0.00/2.05/1.24	0.00/0.29/0.23	0.00/0.27/0.36
	BART-B	0.00/0.75/0.57	0.00/0.27/0.46	0.00/0.16/0.26
CNN/Daily Mail ← Delve	T5-L	0.00/0.25/0.73	0.00/0.03/0.35	0.00/0.03/0.74
	T5-B	0.00/0.34/2.55	0.00/0.09/4.24	0.00/0.06/7.33
	BART-L	0.00/2.15/1.41	0.00/0.50/2.77	0.00/0.87/5.14
	BART-B	0.00/1.89/1.91	0.00/0.80/1.89	0.00/1.20/4.68
CNN/Daily Mail ← S2orc	T5-L	0.00/0.40/1.15	0.00/0.05/0.59	0.00/0.05/1.19
	T5-B	0.00/0.48/1.49	0.00/0.06/1.93	0.00/0.05/4.84
	BART-L	0.00/0.81/1.36	0.00/0.30/2.44	0.00/0.63/3.95
	BART-B	0.00/1.30/2.55	0.00/0.68/2.69	0.00/1.03/5.13
CNN/Daily Mail ← Random domain	T5-L	0.00/0.06/0.88	0.00/0.03/0.42	0.00/0.02/0.89
	T5-B	0.00/2.47/1.01	0.00/0.28/1.32	0.00/0.17/3.63
	BART-L	0.00/0.95/0.98	0.00/0.26/1.64	0.00/0.29/3.32
	BART-B	0.00/0.00/1.32	0.00/0.02/0.36	0.00/0.01/0.42
CNN/Daily Mail ← SAMSum	T5-L	0.00/5.92/1.50	0.00/0.56/0.77	0.00/0.49/1.55
	T5-B	0.00/1.62/1.78	0.00/0.48/3.97	0.00/0.82/6.94
	BART-L	0.00/0.65/1.07	0.00/0.90/1.78	0.00/0.24/1.53
	BART-B	0.00/3.65/2.04	0.00/1.71/3.34	0.00/0.80/4.85

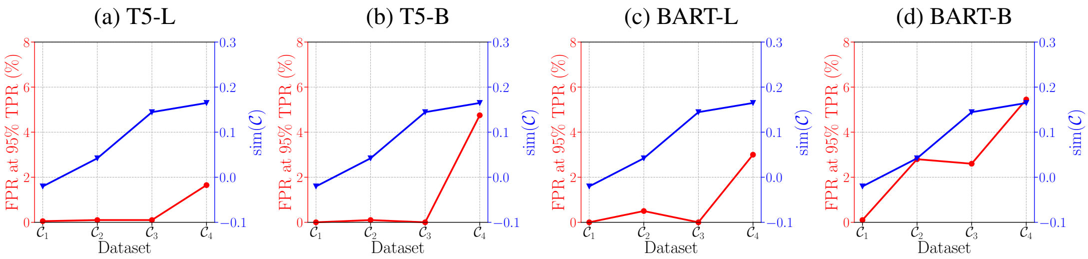

Figure 16: FPR at 95% TPR vs. sim($\mathcal{C}$); The relevant documents sourced from the Delve (1K) domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing SAMSum, CNN/Daily Mail, Random Domain, and S2orc.

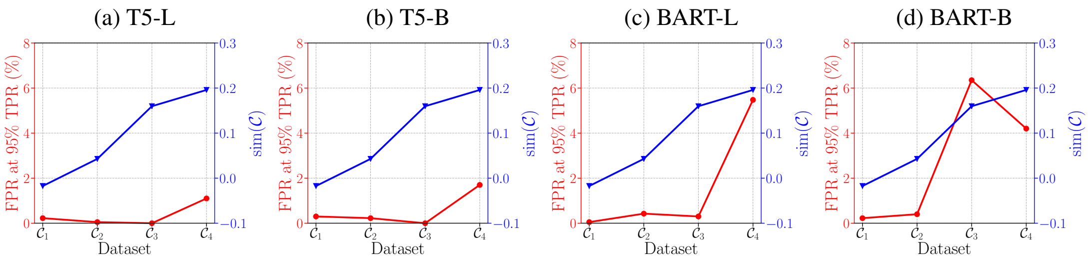

Figure 17: FPR at 95% TPR vs. sim($\mathcal{C}$); The relevant documents sourced from the S2orc domain, and varying irrelevant domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing SAMSum, CNN/Daily Mail, Random Domain, and Delve.

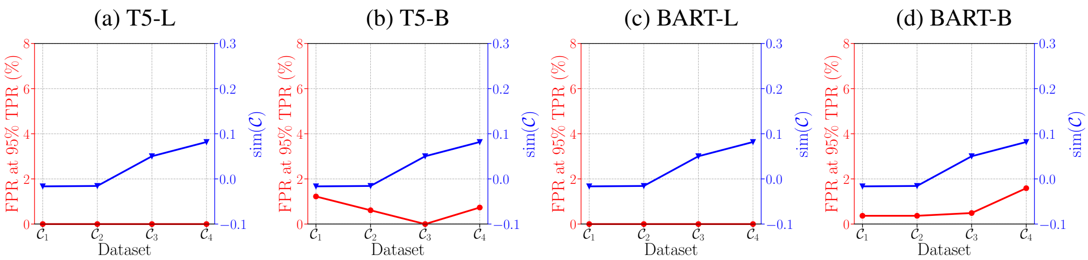

Figure 18: FPR at 95% TPR vs. sim($\mathcal{C}$); The relevant documents sourced from the SAMSum domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing Delve, S2orc, Random Domain, and CNN/Daily Mail.

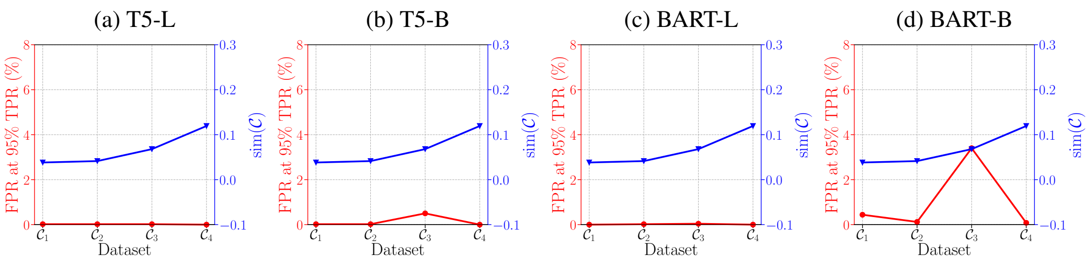

Figure 19: FPR at 95% TPR vs. sim($\mathcal{C}$); The relevant documents sourced from the CNN/Daily Mail domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing Delve, S2orc, SAMSum, and Random Domain.

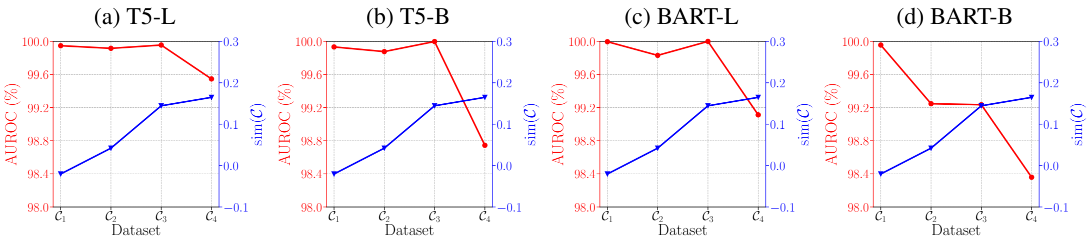

Figure 20: AUROC vs. sim($\mathcal{C}$); The relevant documents sourced from the Delve (1K) domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing SAMSum, CNN/Daily Mail, Random Domain, and S2orc.

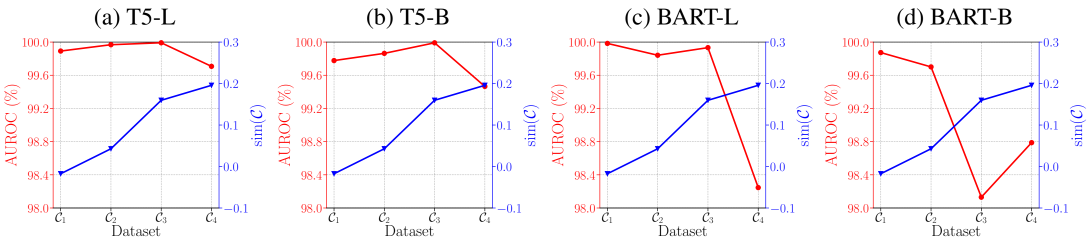

Figure 21: AUROC vs. sim($\mathcal{C}$); The relevant documents sourced from the S2orc domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing SAMSum, CNN/Daily Mail, Random Domain, and Delve.

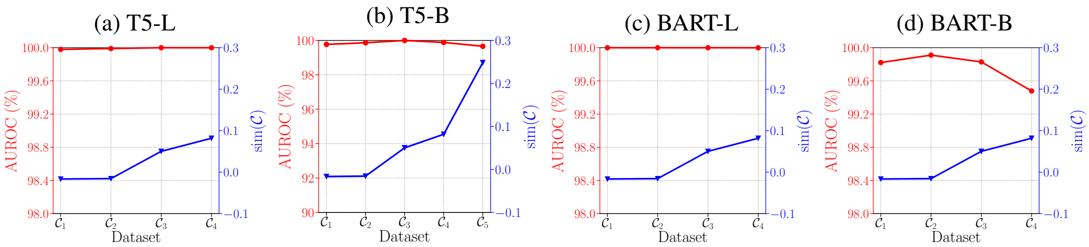

Figure 22: AUROC vs. sim($\mathcal{C}$); The relevant documents sourced from the SAMSum domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing Delve, S2orc, Random Domain, and CNN/Daily Mail.

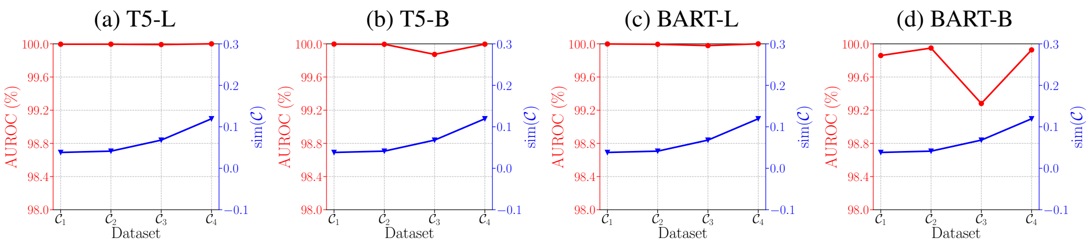

Figure 23: AUROC vs. sim($\mathcal{C}$); The relevant documents sourced from the CNN/Daily Mail domain, and varying irrelevant document domains represented as $\mathcal{C}_1$ through $\mathcal{C}_4$, encompassing Delve, S2orc, SAMSum, and Random Domain.

A.8 Hyper-parameter Sensitivity

In this section, we show how different choice of the hyper-parameter α and β affects the in-domain irrelevant document detection performance of our method. Specifically, we present the relationship between the selection of α and β and irrelevant document detection performance. Each figure in this section displays FPR at 95% TPR or AUROC of our method on each dataset and model when selecting different combinations of α and β. The details of hyper-parameters can be found in Table 12 in A.3.

We observe that the best performance occurs

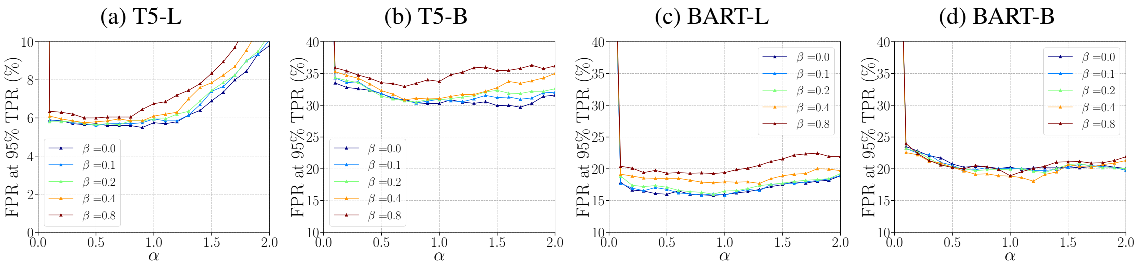

Figure 24: FPR at 95% TPR vs. Hyper-parameter on Delve-ID (1K)

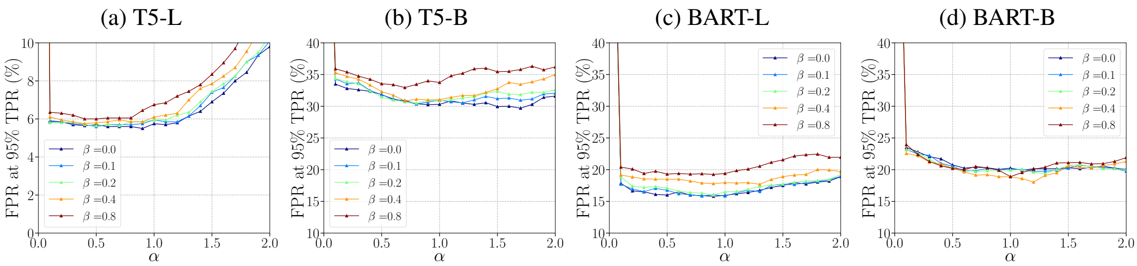

Figure 25: FPR at 95% TPR vs. Hyper-parameter on Delve-ID (8K)

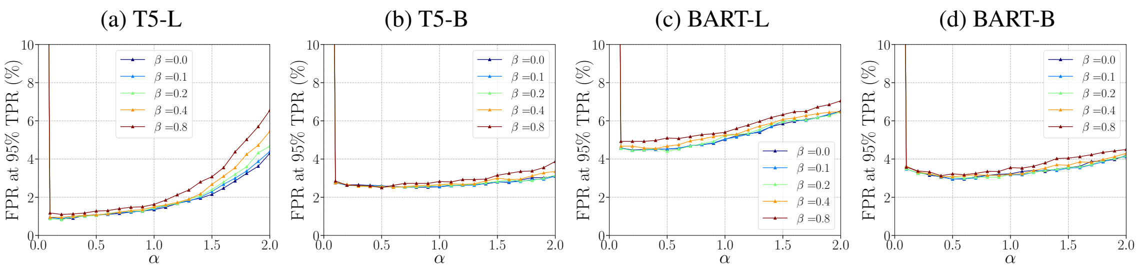

Figure 26: FPR at 95% TPR vs. Hyper-parameter on S2orc-ID

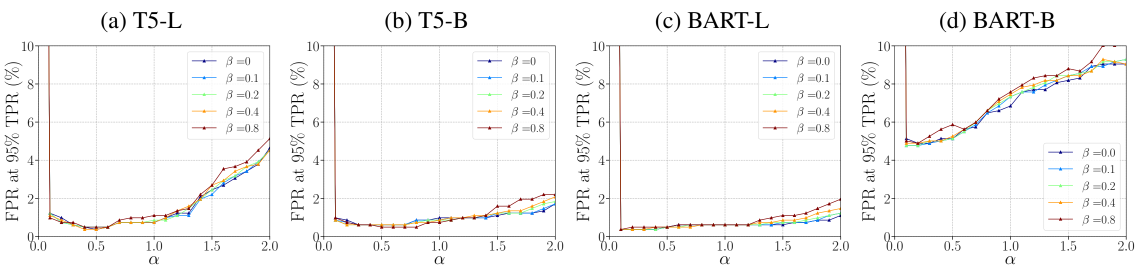

Figure 27: FPR at 95% TPR vs. Hyper-parameter on SAMSum-ID

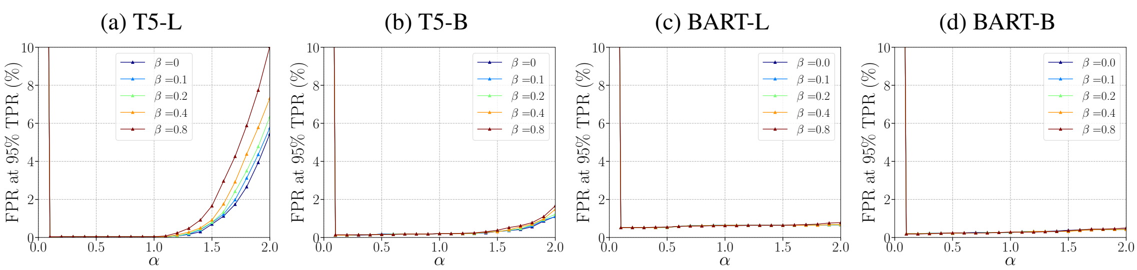

Figure 28: FPR at 95% TPR vs. Hyper-parameter on CNN/Daily Mail-ID

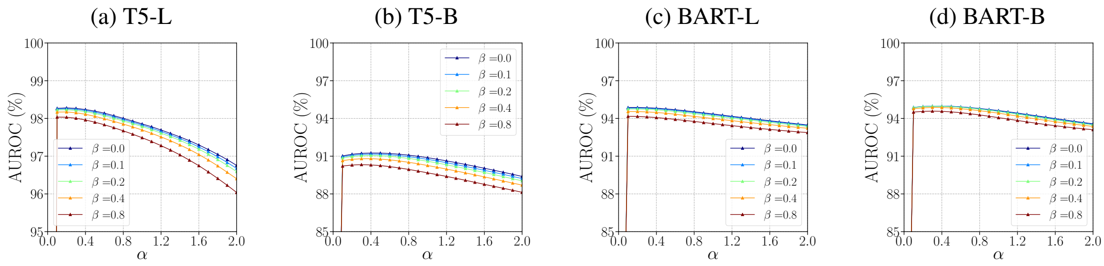

Figure 29: AUROC vs. Hyper-parameter on Delve-ID (1K)

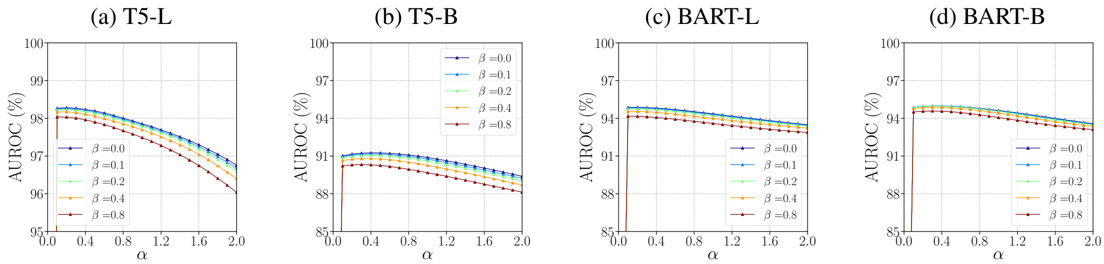

Figure 30: AUROC vs. Hyper-parameter on Delve-ID (8K)

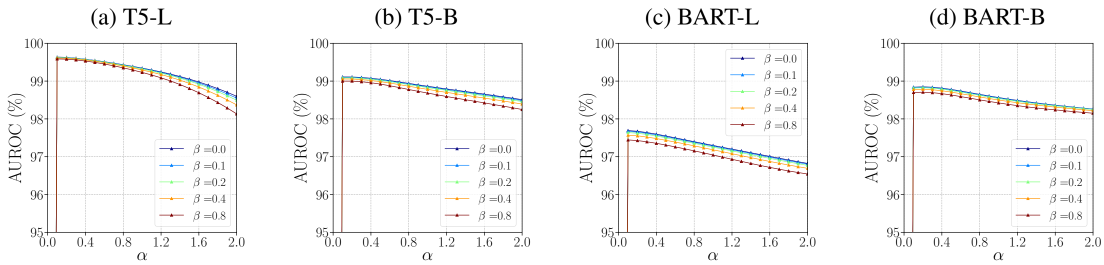

Figure 31: AUROC vs. Hyper-parameter on S2orc-ID

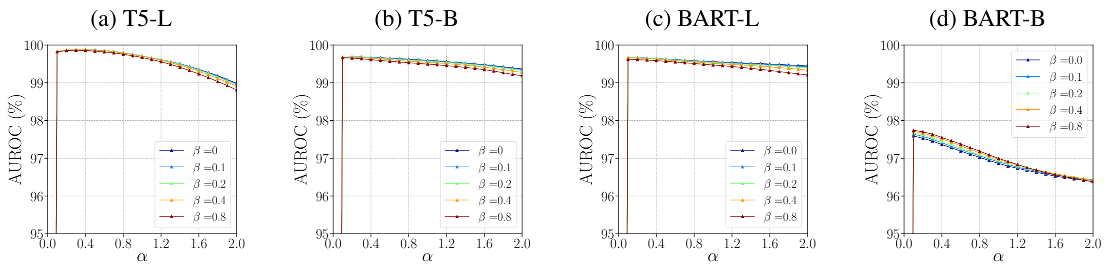

Figure 32: AUROC vs. Hyper-parameter on SAMSum-ID

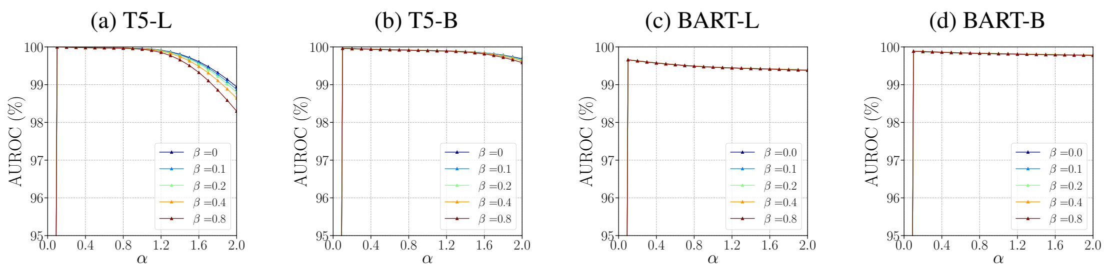

Figure 33: AUROC vs. Hyper-parameter on CNN/Daily Mail-ID

near $\alpha = 0.6$ for most choices of β and the best performance occurs near $\beta = 0.2$ for most choices of α. We also observe that the performance does not change much when α varies from 0 to 1. Similarly, the performance also changes slightly when β varies from 0 to 0.4. We observed that the performance of CODE on both types of pretrained models is more sensitive to α compared to β.

The correspondence between the figures and the setting is as follows:

- Figure 24: FPR at 95% TPR on Delve-ID (1K) dataset and each model.
- Figure 25: FPR at 95% TPR on Delve-ID (8K) dataset and each model.
- Figure 26: FPR at 95% TPR on S2orc-ID dataset and each model.
- Figure 27: FPR at 95% TPR on SAMSum-ID dataset and each model.
- Figure 28: FPR at 95% TPR on CNN/Daily Mail-ID dataset and each model.
- Figure 29: AUROC on Delve-ID (1K) dataset and each model.
- Figure 30: AUROC on Delve-ID (8K) dataset and each model.
- Figure 31: AUROC on S2orc-ID dataset and each model.
- Figure 32: AUROC on SAMSum-ID dataset and each model.
- Figure 33: AUROC on CNN/Daily Mail-ID dataset and each model.

A.9 Supplementary Material for Effectiveness of In-domain Irrelevant Documents in Pretraining

A.9.1 Pretraining with Irrelevant Documents vs. Without Irrelevant Documents

In this subsection, we study how the irrelevant documents in the pretraining affect the performance. Specifically, we pretrained the T5-Large model using only relevant documents from the Delve dataset.

We evaluate the pretrained models with three metrics for text summarization, and Table 17 presents the results. We observe that irrelevant

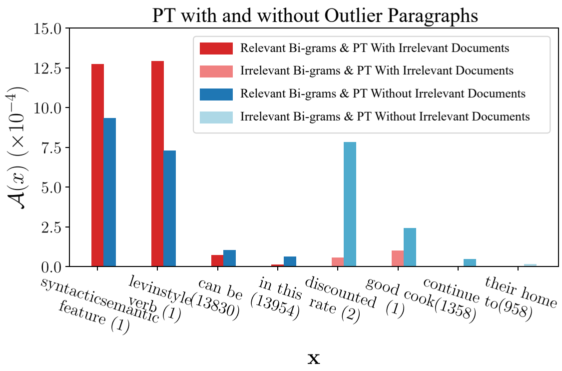

Figure 34: Cross-attention scores on eight bi-grams when T5-Large is pretrained with and without irrelevant documents. Bi-gram occurrences are in the parenthesis.

documents can slightly improve the generation performance. This may be due to the fact that irrelevant documents may help enrich the corpus in that domain, therefore enhancing the summarization performance.

Table 18 presents three metrics of irrelevant document detection under the case where T5-Large is pretrained with and without irrelevant documents. We observe that irrelevant documents plays an important role for irrelevant document detection task.

A.9.2 Case Study

To provide more insights, we spotlight eight bi-gram phrases, of which half originate from relevant documents and the remainder from irrelevant documents. Furthermore, half of these bi-grams frequently appear, as indicated by their occurrence counts in parenthesis. Comparing the cross-attention scores when the T5-Large model is pretrained with (i.e., red bars) and without (i.e., blue bars) irrelevant documents, we observed that including irrelevant documents enhances the attention scores of less frequent bi-grams in relevant documents, simultaneously depressing scores for the less frequent irrelevant bi-grams. For instance, after incorporating irrelevant documents in pretraining, the relevant bi-gram “levinstyle verb" with a single occurrence nearly doubles its attention score, whereas the irrelevant bi-gram “discounted rate" with two occurrences sees an 80% attention reduction. Moreover, we observed that the attention scores of domain-agnostic phrases also wane, potentially bolstering irrelevant document detection capabilities. For example, after incorporating irrelevant documents in pretraining, we observe notable reductions in attention scores for the domain-agnostic phrases “can be" in relevant documents and “continue to" in irrelevant documents.

Table 17: Performance of pretrained model vs. irrelevant documents

		ROUGE-1	ROUGE-2	ROUGE-L
irrelevant documents	With	19.34	3.38	14.42
	Without	17.00	2.45	12.87

Table 18: Performance vs. irrelevant documents (%)

		FPR at 95% TPR	AUROC	AUPR
irrelevant documents	With	5.80	98.08	97.03
	Without	80.45	62.92	66.99

Table 19: The performance of the baseline Frozen under different hidden layer dimensions.

Models	FPR (95%) TPR $\downarrow$	AUROC $\uparrow$	AUPR $\uparrow$
		Frozen	
(24N, 8N, N)	28.98 ± 0.74	93.75 ± 0.14	93.08 ± 0.15
(16N, 4N, N)	29.08 ± 1.00	93.82 ± 0.11	93.12 ± 0.09
(4N, 2N, N)	30.30 ± 0.94	92.87 ± 0.21	93.57 ± 0.16

A.10 Effect of FNN size on the detection performance of baseline algorithms

We test the impact of different sizes of FNN on the detection performance of Frozen on T5-Large and Delve-ID (1K). The results are shown in Table 19. We find that as the hidden layer dimension of FNN increases, the detection performance of Frozen shows a slight improvement, but the overall improvement is not significant.

A.11 Time consumption of CODE and baselines.

We compare the time computation of CODE and baselines. The time complexity of CODE is $O(|X| \times |\hat{Y}|)$, where $|X|$ represents the length of a single document, and $|\hat{Y}|$ represents the length of the generated summary. We test the time consumption of CODE and baseline algorithms on T5-Large and Delve-ID (1K) during the hyper-parameter tuning and testing phases. The batch size is uniformly set to 1 for testing CODE and the baseline algorithms. During the hyper-parameter tuning phase, for CODE, we measure the time consumption required to complete a hyper-parameter search for a single hyper-parameter combination; for the baseline algorithms, we measure the time consumption required to complete one epoch of training. The test results are shown in Table 20, indicating that CODE has higher time efficiency than the two baseline algorithms during both the hyper-parameter tuning and testing phases.

Table 20: Time consumption of CODE and the baselines.

	Tuning (s)	Testing (s)
CODE	51	72
Frozen	504	157
FT-ALL	1,352	155