Exploring the Challenges of Open Domain Multi-Document Summarization

John Giorgi[1,2,3*] **Luca Soldaini**[4†] **Bo Wang**[1,2,3] **Gary Bader**[1,2]
Kyle Lo[4†] **Lucy Lu Wang**[4,5†] **Arman Cohan**[4†]

[1]University of Toronto [2]Terrence Donnelly Centre [3]Vector Institute for AI
[4]Allen Institute for AI [5]University of Washington

john.giorgi@utoronto.ca, {lucas, kylel, armanc}@allenai.org, lucylw@uw.edu

Abstract

Multi-document summarization (MDS) has traditionally been studied assuming a set of ground-truth topic-related input documents is provided. In practice, the input document set is unlikely to be available a priori and would need to be retrieved based on an information need, a setting we call open-domain MDS. We experiment with current state-of-the-art retrieval and summarization models on several popular MDS datasets extended to the open-domain setting. We find that existing summarizers suffer large reductions in performance when applied as-is to this more realistic task, though training summarizers with retrieved inputs can reduce their sensitivity retrieval errors. To further probe these findings, we conduct perturbation experiments on summarizer inputs to study the impact of different types of document retrieval errors. Based on our results, we provide practical guidelines to help facilitate a shift to open-domain MDS. We release our code and experimental results alongside all data or model artifacts created during our investigation.[1]

1 Introduction

Summarization is a popular task in natural language processing (NLP) that aims to automatically generate accurate and concise summaries for some given input text. Multi-document summarization (MDS) extends this task to provide multiple topic-related documents as input, where the goal is to summarize the salient information while avoiding redundancy. MDS has become a popular research objective with many proposed approaches (Yasunaga et al., 2017; Liao et al., 2018; Liu and Lapata, 2019; Li et al., 2020; Jin et al., 2020; Mao et al., 2020; Zhang et al., 2020a; Pasunuru et al., 2021b; Xiao et al., 2022) and has found important applications in summarizing clusters of news articles (Fabbri et al., 2019; Gholipour Ghalandari et al., 2020), scientific literature (Lu et al., 2020), medical studies (Wallace et al., 2021; DeYoung et al., 2021), and legal documents (Shen et al., 2022).

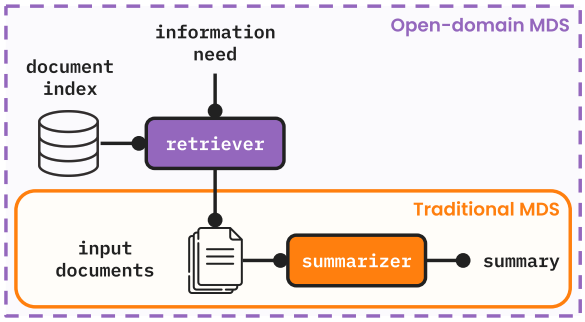

Figure 1: Prior work assumes a ground-truth input document set is given at train and test time ("traditional MDS"). We investigate how summarizers behave under the more realistic "open-domain" setting where documents must be retrieved given an information need.

Previous task definitions for MDS assume a ground-truth input document set is provided at both train and test time. However, this is an artifact of the dataset curation process; in practice, the input document set is unlikely to be available a priori. In many practical settings, the document set would be defined by an information need, expressed as a *query*. The query could be a question, e.g., "*Do vitamin D supplements improve the physical capabilities of elderly hospital patients?*" or topic statement, e.g. "*Report on vulnerabilities of the US power/electrical grid and efforts to change or improve it.*" Documents relevant to the query would then need to be retrieved from a larger collection (an *index*) and summarized (Figure 1). Because even state-of-the-art information retrieval (IR) methods are imperfect, errors, like the retrieval of irrelevant documents, will occur. It is an open

*Work performed during internship at AI2
†Core contributors. See authors contributions
[1]https://github.com/allenai/open-mds

question how existing summarizers behave under this more realistic "open-domain"[2] setting.

Here, we lay the groundwork for a shift in focus to open-domain MDS. We begin by providing a formal task definition and highlighting a particularly promising approach: "retrieve-then-summarize" (§2). Our major contributions are:

- We study open-domain MDS by bootstrapping the task with existing MDS datasets, state-of-the-art retrievers and summarizers (§4).
- We find that summarizers suffer large performance reductions in the open-domain setting, even when retrieval performance is high (§5); training with retrieved inputs can reduce this sensitivity to imperfect document retrieval (§6).
- We probe what drives this reduction in summarization performance by subjecting summarizers to an extensive suite of "perturbations" designed to simulate document retrieval errors (§7).

Based on our results, we provide detailed, practical guidelines for future work in MDS. We release our code and experimental results alongside all data or model artifacts created during our investigation.

2 Open-domain MDS

In the traditional MDS task definition, a model is given a set of topic-related input documents $D = \{d_1, ..., d_k\}$ and must generate a summary S that accurately and coherently summarizes the information in D. Such models are typically trained in a supervised fashion to minimize the difference between S and a (usually human-written) reference summary R. The goals (and evaluation) remain the same in open-domain MDS, but instead of D, the inputs are a query q and a large collection (*index*) of documents $\mathcal{D}_{\text{index}}$, where the size of $\mathcal{D}_{\text{index}}$ is $\gg k$ (Figure 1). The specifics of q, $\mathcal{D}_{\text{index}}$, and S will depend on the application. For example, in automatic literature review (Wallace et al., 2021; DeYoung et al., 2021), q would be a research question or statement, e.g. "*Is massage therapy effective for people with musculoskeletal disorders compared to any other treatment or no treatment?*", $\mathcal{D}_{\text{index}}$ would be a large corpus of scientific literature (e.g. PubMed), and S would be a literature review-style discussion, e.g. "*Massage therapy, as a stand-alone treatment, reduces pain and improves function compared to no treatment in some musculoskeletal conditions.*"[3]

There are multiple ways to approach open-domain MDS. One possibility is to think of q as a *prompt* to a large language model (LLM) capable of in-context learning (Brown et al., 2020) to generate S; in which case we can think of $\mathcal{D}_{\text{index}}$ as the LLMs training data and retrieval as happening implicitly during inference. However, because all information is stored in the model's weights, this approach requires extremely large models, cannot produce summaries for events that do not appear in the training data, and does not provide provenance for the model-generated summaries.

Another approach is to introduce an explicit retrieval step over an external knowledge source, which we refer to as "retrieve-then-summarize."[4] It works as follows: a *retriever* must rank all documents in $\mathcal{D}_{\text{index}}$ from most-to-least relevant given q. The top-k documents are then input to a summarizer, where k is a parameter that may be tuned for a particular use case. This approach has some desirable properties: (1) $\mathcal{D}_{\text{index}}$ can be updated with new documents (e.g. the latest scientific articles) without the need to re-train the retriever or summarizer, and (2) it naturally provides provenance for model-generated summaries: the top-k documents. In the remainder of the paper, we focus our investigation on the retrieve-then-summarize approach.

3 Research Questions

Our experiments are designed to probe the following research questions:

- **R1**: *How do state-of-the-art retrievers and summarizers perform in the open-MDS setting?*
- **R2**: *Does training in the open-domain setting make multi-document summarizers less sensitive to imperfect document retrieval?*
- **R3**: *What types of retrieval errors occur in the open-domain setting? How does each error type affect summarization performance?*

The rest of the paper is structured as follows. In §4, we describe the experimental design. The remaining sections present our experimental results as they relate to **R1** (§5), **R2** (§6), and **R3** (§7).

[2] We are borrowing this term from the question answering literature, which also uses it to denote the setting where only a query is provided as input

[3] Truncated example from MSˆ2 (DeYoung et al., 2021)

[4] Inspired by the "retrieve-then-generate" approach popular in knowledge-intensive NLP (Petroni et al., 2021)

Table 1: Dataset statistics, counting whitespace tokens and punctuation. *Following (DeYoung et al., 2021), we take the first 25 documents as input (full statistics in parentheses). †Multi-XScience and MSˆ2 each have inputs that are always provided (and never retrieved), the target articles abstract and the target reviews background section.

Dataset	Domain	Max Docs	Mean Docs	Avg. Tokens/Input Document	Avg. Tokens/Reference Summary
Multi-News	News Articles	10	2.7	788	267
WCEP-10	News Articles	10	9.1	494	33
Multi-XScience†	Scientific Literature	20	4.1	153	125
MSˆ2*†	Medical Studies	25 (401)	17 (23)	332	58
Cochrane*	Medical Studies	25 (537)	9 (11)	266	69

Table 2: Evaluated multi-document summarizers and the datasets for which a fine-tuned model is publicly available (or was trained by us).

Model	Fine-tuned on	Max Input Len.	Zero-shot?
LED	MSˆ2, Cochrane	16384	✗
PEGASUS	Multi-News	1024	✗
PRIMERA	Multi-News, WCEP-10, Multi-XScience	4096	✓
LSG-BART	Multi-News, WCEP-10	4096	✗

4 Bootstrapping Open-domain MDS

Since no large-scale annotated datasets or trained models exist for open-domain MDS, we bootstrap this task using existing datasets (§4.1) and models (§4.2, §4.3). We describe operationalization considerations in §4.4 and evaluation metrics in §4.5. We use MDS datasets for which high-performing summarizers exist and whose examples are annotated with ground-truth input documents (D) and human-written reference summaries (R).

4.1 Datasets

We investigate a representative selection of 5 MDS datasets comprised of news articles, medical studies, and scientific literature. Dataset statistics are listed in Table 1. The inputs of these datasets generally consist only of the documents to summarize. However, Multi-XScience and MSˆ2 each provide additional text as input — the target article's abstract and the target review's background section, respectively. In our experiments, we always provide this additional text (and do not retrieve it).[5]

4.2 Retrieval Models

Broadly speaking, retrievers are divided into two categories, *sparse* and *dense*. Sparse retrievers determine the relevance of a document to a query using counts of overlapping terms weighted by their frequencies. Dense retrievers embed documents and queries into a shared embedding space (typically using neural language models) and use proximity in this space to determine relevance. Retrievers from these families exhibit different characteristics and limitations (MacAvaney et al., 2022). We investigate two representative retrievers, one from each family: BM25 (sparse, Robertson et al. 1994) and Contriver (dense, Izacard et al. 2022a).[6] Both BM25 and Contriever achieve strong zero-shot performance for a variety of retrieval tasks,[7] making them particularly suitable for our purposes.

4.3 Multi-document Summarization Models

All multi-document summarizers we experiment with are transformer-based encoder-decoders (Vaswani et al., 2017) trained for abstractive summarization, representing the current state-of-the-art approach. The input is a string containing one or more documents concatenated with special tokens (e.g. `<doc-sep>`). Following Xiao et al. (2022), we truncate each document based on the maximum input length of the model divided by the total number of documents. The models are listed in Table 2. Where available, we use publicly available fine-tuned copies of these models; in the case of MSˆ2 and Cochrane, we fine-tune a model ourselves.[8]

4.4 Operationalizing Retrieve-then-Summarize

To extend these datasets and models to the open-domain setting and operationalize the retrieve-then-summarize approach, we address the following:

How to choose a query? In open-domain MDS, a query is anything that defines the documents to summarize (e.g. a question or topic statement). Ideally, a human-written query would be available for each example in our dataset. However, existing MDS datasets do not provide queries.[9] Therefore,

[5] See Appendix A for more details

[6] See Appendix B for more details

[7] See the BEIR (Thakur et al., 2021) zero-shot benchmark

[8] See Appendix C for more details

[9] Although query-focused MDS datasets exist, they are extremely small (on the scale of 10s of examples) and are there-

we use R, the human-written reference summaries, as a *pseudo*-query,[10] as it naturally describes the input documents of each example.

How to assemble the document index? For our purposes, we take the set of all documents in the train, validation, and test splits of each dataset to form $\mathcal{D}_{\text{index}}$. This guarantees that the correct documents for each example are present in the index while providing plenty of negative examples.

How many documents to summarize? The number of retrieved documents to summarize, k, is a parameter that can be tuned for different use cases. To determine its impact on summarization performance, we investigate three strategies:

- **Max**: Choose k as the *maximum* number of input documents for any example in a given dataset. Tends to select for *recall* at the cost of *precision*.
- **Mean**: Choose k as the *mean* number of input documents for all examples in a given dataset. Tends to select for *precision* at the cost of *recall*.
- **Oracle** Choose k as the *ground-truth* number of input documents for any example in a given dataset. This mimics the scenario where all documents with a relevance score (assigned by the retriever) above a certain optimal threshold (a hyperparameter) are retained.

We note that these decisions result in a highly idealized setting. Using R as query leaks information about the reference summary into the retrieval step and likely inflates summarization performance. In practice, $\mathcal{D}_{\text{index}}$ will be much larger (e.g. PubMed-, Wikipedia-, or even Web-scale), making retrieval more difficult. However, as we will show in §5, even this idealized setting often leads to large reductions in summarization performance.

4.5 Evaluation

We follow previous work by evaluating summarization with ROUGE-1, ROUGE-2, and ROUGE-L scores (Lin, 2004). To provide a single metric for comparison, we report ROUGE-Avg F1, the average F1-score of ROUGE-1/2/L. We also report BERTScore (Zhang et al., 2020b), which has been shown to better correlate with human judgment (Yuan et al., 2021; Fischer et al., 2022).[11] These automatic summarization metrics output a score by comparing a model-generated summary to a reference summary. For document retrieval performance, we report the precision and recall at k (abbreviated P@K and R@K). P@K is the fraction of the top-k retrieved documents considered relevant, and R@K is the fraction of known relevant documents appearing in the top-k retrieved results. These are suitable metrics when the input documents do not have an inherent order, as is usually the case in MDS and is true for the MDS datasets we investigate. We evaluate both retrievers and summarizers on the test splits of each dataset, except for MSˆ2 and Cochrane, where we evaluate on the validation set because the test split is blind.

5 Multi-Document Summarizers Do Not Generalize to the Open-Domain Setting

Here we present the results of our open-domain MDS experiments. In general, we find that existing summarizers suffer large reductions in performance when applied as-is to the open-domain setting, even when retrieval performance is high (Table 3).[12] Below, we provide key observations on how the individual components (summarizer and retriever) behave within a retrieve-then-summarize framework, particularly how their individual performances relate to overall system performance.

Strong summarizers are more sensitive to imperfect retrieval than weak summarizers We observe a relationship between a summarizer's (baseline) performance on a dataset and its sensitivity to imperfect document retrieval (Table 3). The largest reductions in summarization performance were observed for the most performant summarizers (and vice versa), despite retrieval performance being the highest in these cases. However, this relationship is confounded by differences in retrieval performance between datasets. To control for this, we conduct experiments comparing fine-tuned PRIMERA to PRIMERA evaluated *zero-shot* (Table 4).[13] This allows us to hold the dataset, model architecture, and retriever constant, isolating the relationship between summarization perfor-

fore not suitable for the large-scale analysis we are proposing

[10] Except for MSˆ2, where we found the provided "background" section to perform better as a query. See §4.1.

[11] BERTScore has many parameters which affect the final score. For reproducibility, a hashcode is produced. Our hashcode is: `microsoft/deberta-xlarge-mnli_L40_no-idf_version=0.3.11(hug_trans=4.22.0.dev0)-rescaled_fast-tokenize`.

[12] Results for sparse and dense retrievers were comparable and exhibited similar trends. We elect to show results for the sparse retriever. See Appendix D for dense retriever results.

[13] We choose PRIMERA as it is the only model we evaluate with demonstrated zero-shot capabilities

Table 3: Results of the open-domain MDS experiments. We observe: (1) retrieval performance ranges from high (Multi-News, WCEP-10) (**dark blue**) to low (Multi-XScience, MSˆ2, Cochrane), (2) when summarizers trained on these datasets are provided retrieved documents, they suffer from significant drops in performance (**dark red**); more severe performance drops were observed in cases where baseline summarization performance was relatively high (**dark green**). Experiments here used a sparse retriever (BM25); similar results were observed using a dense retriever (Contriever, see Table 8). Statistically significant results are underlined (paired t-test, p = 0.01).

		Retrieval			Summarization			
Dataset	Model	Top-k Strategy	P@K	R@K	ROUGE-Avg F1	Δ ROUGE-Avg F1	BERTScore F1	Δ BERTScore F1
Multi-News	PRIMERA	max (10)	0.22	0.82	31.66	-7.39	31.78	-10.33
		mean (3)	0.64	0.74	–	-2.82	–	-4.08
		oracle	0.75	0.75	–	-1.61	–	-2.36
	PEGASUS	max	–	–	31.23	-8.49	29.88	-10.87
		mean	–	–	–	-2.08	–	-2.93
		oracle	–	–	–	-1.15	–	-1.50
WCEP-10	PRIMERA	max (10)	0.63	0.67	35.50	-1.02	48.26	-0.76
		mean (9)	0.66	0.64	–	-0.90	–	-0.68
		oracle	0.67	0.67	–	-0.53	–	-0.32
	LSG-BART-base	max	–	–	35.76	-1.15	48.17	-0.85
		mean	–	–	–	-1.19	–	-0.84
		oracle	–	–	–	-0.88	–	-0.54
Multi-XScience	PRIMERA	max (20)	0.06	0.40	18.31	-0.57	10.57	-1.82
		mean (4)	0.16	0.27	–	-0.25	–	-1.27
		oracle	0.23	0.23	–	-0.06	–	-0.97
MSˆ2	LED-base	max (25)	0.16	0.22	19.66	-0.14	22.74	-0.47
		mean (17)	0.18	0.18	–	-0.10	–	-0.13
		oracle	0.18	0.18	–	-0.01	–	-0.21
Cochrane	LED-base	max (25)	0.17	0.57	17.39	-0.28	23.12	-2.11
		mean (9)	0.31	0.44	–	+0.34	–	-0.32
		oracle	0.40	0.40	–	+0.10	–	+0.00

Table 4: Results of the open-domain MDS experiments with zero-shot summarizers. Controls for differences in datasets & models, isolating the relationship between summarization performance in traditional and open-domain settings. Top-k strategy *mean* is used. Statistically significant results are underlined (paired t-test, p = 0.01).

		Retrieval			Summarization			
Dataset	Model	Retriever	P@K	R@K	ROUGE-Avg F1	Δ ROUGE-Avg F1	BERTScore F1	Δ BERTScore F1
Multi-News	PRIMERA	sparse (BM25)	0.64	0.74	31.66	-2.82	31.78	-4.08
		dense (Contriever)	0.59	0.70	–	-3.31	–	-4.60
	↪ *zero-shot*	sparse	–	–	23.58	-0.09	18.66	-0.39
		dense	–	–	–	-0.27	–	-0.44
WCEP-10	PRIMERA	sparse	0.66	0.64	35.50	-0.90	48.26	-0.68
		dense	0.66	0.63	–	-0.14	–	+0.68
	↪ *zero-shot*	sparse	–	–	21.43	+0.35	25.48	+0.72
		dense	–	–	–	+1.00	–	+2.19
Multi-XScience	PRIMERA	sparse	0.16	0.27	18.31	-0.25	10.57	-1.27
		dense	0.16	0.24	–	-0.81	–	-0.96
	↪ *zero-shot*	sparse	–	–	15.18	+0.69	6.02	-0.47
		dense	–	–	–	+0.46	–	+0.00

mance in the traditional and open-domain settings. Here, the trend is clear: stronger summarizers are more sensitive to imperfect retrieval than weaker summarizers.[14]

One explanation is that weak summarizers have less to lose from imperfect retrieval, perhaps because they are not adequately performing the task even when trained on ground-truth input documents. They may, to a greater degree than strong summarizers: hallucinate (generate coherent but irrelevant text), rely on shallow heuristics (Kryscinski et al., 2019), or use only a fraction of input documents (Wolhandler et al., 2022). To probe this hypothesis, we construct several heuristic baselines that mimic these behaviours and compare their performance to trained models (Table 5; see Appendix E for more details). We find that, for example, copying the provided background section of MSˆ2 performs comparably to the fine-tuned summarizer, suggesting that the observed insensitivity to retrieval errors could be due to the summarizer exploiting this heuristic. We observe a similar result for Multi-XScience by copying the document with the highest token overlap to the reference summary. Future work should carefully establish that summarizers are performing adequately — via extensive evaluation and comparison to base-

[14] We use "strong" and "weak" as shorthand to refer to cases where summarization performance is high (e.g. PRIMERA on Multi-News) and low (e.g. LED on Cochrane)

Table 5: Comparing ROUGE-Avg F1 scores of model-generated summaries to heuristic baselines. In some cases, the baselines perform surprisingly close to trained summarizers. All Lead is the concatenation of the first sentence from each input document. Oracle document is the document with the highest token overlap with the reference summary; oracle lead is the first sentence from this document. Background/Abstract is the additional input from MSˆ2 and Multi-XScience. The best baseline for each dataset is **bolded**.

		Baselines				
Dataset	Best Summarizer	Δ Random Summary	Δ All Lead	Δ Oracle Document	Δ Oracle Lead	Δ Background/Abstract
Multi-News	31.7	-18.3	-15.3	**-4.1**	-21.8	–
WCEP-10	35.8	-27.8	-24.4	-15.3	**-9.9**	–
Multi-XScience	18.3	-6.2	-5.0	**-0.8**	-9.3	-2.3
MSˆ2	19.7	-10.4	-11.0	-7.6	-4.0	**-0.2**
Cochrane	17.4	-5.0	-4.2	-3.9	**-3.5**	–

lines — before attempting the more challenging open-domain setting.

Better retrieval performance does not always correspond with better summarization performance The performance of the sparse (BM25) and dense (Contriever) retrievers was comparable (Table 3 & Table 8), with the sparse retriever performing better on some datasets (i.e. Multi-News, WCEP-10, Multi-XScience) and the dense retriever performing better on others (i.e. MSˆ2, Cochrane). Both retrievers made a comparable number of errors (see Appendix G). Interestingly, however, better retrieval performance does not always correspond with a smaller impact on summarization performance. For example, on WCEP-10, the sparse retriever performed slightly better, but the reduction in summarization performance was considerably larger. On MSˆ2 and Cochrane, the better-performing dense retriever leads to a larger reduction in summarization performance. This suggests that the two types of retrievers are making characteristically different errors, which has been noted in the literature (MacAvaney et al., 2022). Therefore, future work should not rely *solely* on IR metrics when optimizing retrieval pipelines for open-domain MDS but should also consider the impact on summarization performance directly.

The number of retrieved documents to summarize matters We observe clear differences in the strategy for choosing k, the number of retrieved documents to summarize. Unsurprisingly, the oracle strategy almost always leads to the smallest reduction in summarization performance. This strategy closely mimics the setting of retaining all documents with a relevance score (assigned by the retriever) over a certain threshold but assumes both a strong retriever and a well-calibrated threshold, both of which may be difficult to achieve in practice. Our results suggest that setting k as the *mean* number of relevant documents (if an accurate estimate can be produced) is a reasonable second-best strategy. We note that, relative to the max k strategy, mean k tends to select for precision over recall (see P@K vs. R@K scores in Table 3 & Table 8); future work should consider tuning k for precision to maximize summarization performance.

6 Training in the Open-domain Setting Reduces Sensitivity to Retrieval Errors

A natural question is whether a summarizer's robustness to document retrieval errors at *test* time might be improved by exposing the model to similar errors at *train* time. To explore this, we retrieve the documents for all examples in the train split of each dataset and fine-tune the best-performing summarizers on these examples. We then evaluate them on both the retrieved evaluation set and the original (ground-truth) evaluation set.[15] We find cases where summarization performance in the open-domain setting benefits from the additional training (e.g. Multi-XScience, Figure 2); however, this can come at the cost of performance on the ground-truth evaluation set (e.g. Multi-News). We note again that our retrieval setting is highly idealized. In practice, the document index would be much larger (making retrieval more difficult), and we would not have access to the reference summaries as queries. Nonetheless, our results suggest that existing summarizers can be adapted to the open-domain setting if query-annotated examples and appropriate document indices are available.

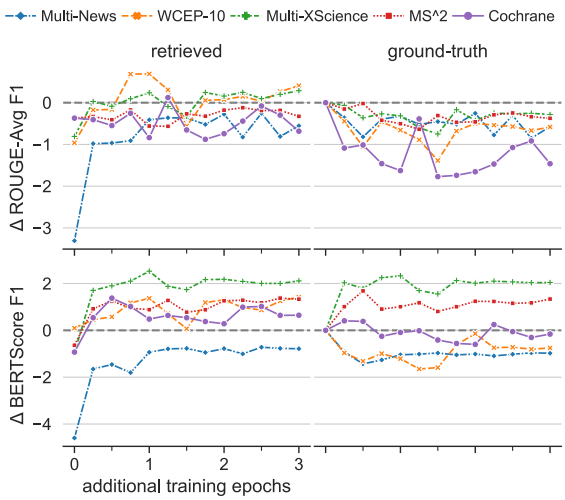

Figure 2: Fine-tuning existing summarizers in the open-domain setting. The additional training can reduce sensitivity to imperfect retrieval but often comes at the cost of performance on the ground-truth evaluation set. The dashed grey line represents no change in performance.

7 A Deep Dive into the Effects of Document Retrieval Errors on MDS

In this section, we investigate what is driving the reduction in summarization performance observed in §5. We begin by carefully categorizing the various retrieval errors that can occur in the open-domain setting. For example, we can erroneously retrieve documents *irrelevant* to the query or fail to retrieve *relevant* documents. For each error type, we design a corresponding "perturbation" that can be applied to the inputs of existing MDS datasets before they are fed to a summarizer. The perturbations are described below and depicted graphically in Figure 3:

- **Addition**: *Add one or more irrelevant documents*, i.e. if D_{actual} is the ground-truth input document set and k_{actual} its size and $D_{\text{perturbed}}$ is the perturbed document set and $k_{\text{perturbed}}$ its size, then $D_{\text{actual}} \subseteq D_{\text{perturbed}}$ and $k_{\text{actual}} < k_{\text{perturbed}}$. This could occur if we correctly retrieve all relevant documents but also retrieve irrelevant ones.
- **Deletion**: *Remove one or more documents*, i.e. $D_{\text{actual}} \supseteq D_{\text{perturbed}}$ and $k_{\text{actual}} > k_{\text{perturbed}}$. This could occur if we retrieve only a fraction of all relevant documents.
- **Replacement**: *Replace one or more relevant documents with irrelevant documents*, i.e. $D_{\text{actual}} \neq D_{\text{perturbed}}$ and $k_{\text{actual}} = k_{\text{perturbed}}$. This could occur if we retrieve the correct number of documents but substitute a relevant document for an irrelevant one.
- **Duplication**: *Duplicate one or more documents*, This could occur if duplicate (or, more likely, near-duplicate) documents exist in the index.[16]
- **Sorting**: *Shuffle the order of the input documents*. The input documents for MDS tasks are typically unordered. However, many summarizers concatenate documents before passing them as input to the model, and it is unknown if models are sensitive to this ordering. Different orderings could occur, for example, if documents are sorted by order of relevance before concatenating.

Token-level perturbations It is well known that NLP models are sensitive to minor *token*-level changes in their inputs (Prabhakaran et al., 2019; Niu et al., 2020; Ribeiro et al., 2020; Moradi and Samwald, 2021). To compare and contrast the *document*-level sensitivity we are investigating with this known token-level sensitivity, we include a token-level perturbation, **backtranslation**. In backtranslation (also called "round-trip translation"), we translate one or more input documents to another (high-resource) language and back again. This process causes small changes, e.g. paraphrasing and synonym substitution, allowing us to create many semantics-preserving, token-level changes to a document. See Appendix H for more details.

7.1 Selecting Documents to Perturb

Each perturbation requires selecting one or more documents to perturb, e.g., in addition and deletion, we need to choose which documents to add and remove. We investigate two strategies:

- **Random**: Select documents to perturb randomly, mimicking a (very) *weak* retriever.
- **Oracle**: Select documents in a way that mimics a *strong* retriever. For example, in deletion, we remove relevant documents in order of *least to most* similar to the reference summary R,[17] whereas in addition, we add irrelevant documents in order of *most to least* similar to R. We compute similarities using the Sentence Transformers library (Reimers and Gurevych, 2019).[18]

[15] See Appendix F for details

[16] Deduplication is non-trivial (Lee et al., 2022) and near-duplicates are not uncommon in large document collections like C4 (Dodge et al., 2021) or S2ORC (Lo et al., 2020)

[17] Similar to §5, this leverages R as a pseudo-query

[18] Specifically, we use all-MiniLM-L6-v2, which is a strong general-purpose model for sentence embeddings. Details here.

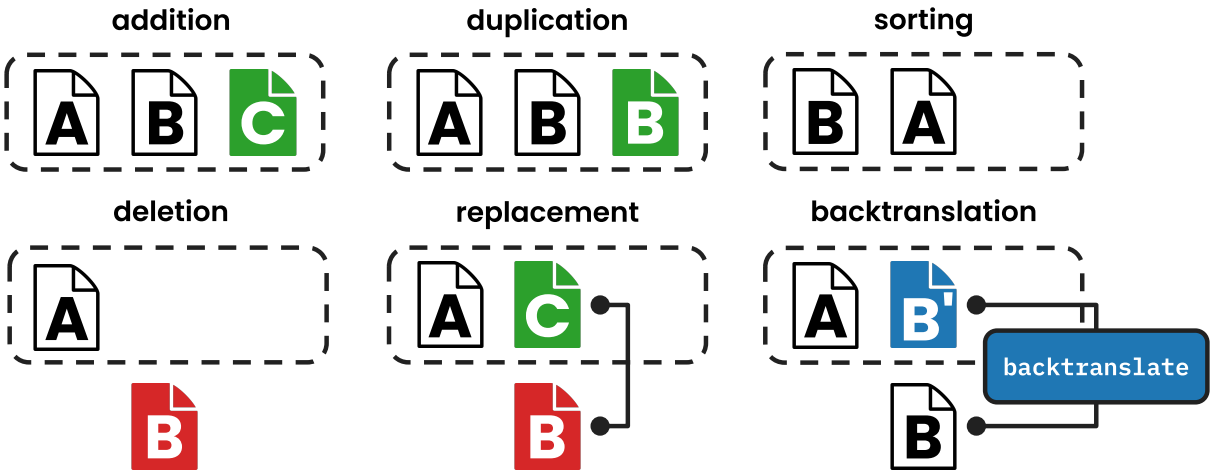

Figure 3: Graphical depiction of the perturbations. The dashed line indicates the input document set. Black and white documents are the ground-truth documents. **Green** documents have been added, **red** documents removed and **blue** documents modified. Unique documents are lettered.

For perturbations that require selecting irrelevant documents (e.g. addition and replacement), we select from the set of all documents in the train, validation, and test splits (excluding the documents from the example we are perturbing). For each document selection strategy, we evaluate summarizer performance under increasing amounts of perturbation, from 0% of documents perturbed up to 100%.

7.2 Results of Simulation Experiments

In Figure 4, we display the results of our experiments simulating document retrieval errors (besides sorting, see below) for two model-dataset pairs.[19] To better contextualize the results, we shade differences in ROUGE $\geq$0.5 — the average difference in summarization performance reported in *CL conferences (Deutsch et al., 2022) — in **red** and the rest in **green**. This serves as a rough yardstick to help identify large drops in performance. We also symlog (Webber, 2012) the y-axis to make small changes in performance more apparent. In general, the results are congruent with our open-domain MDS experiments (§5): large reductions in summarization performance are observed even in cases of few simulated errors and, strong summarizers (Figure 4, left) are more sensitive to retrieval errors than weak summarizers (Figure 4, right). Below, we discuss other notable trends in detail.

Models are insensitive to duplicates and small token-level changes A consistent trend across most models and datasets was an insensitivity to duplicate documents, even in the extreme case of 80-100% duplication, suggesting that deduplication efforts on the document index are unlikely to translate to improvements in summarization performance. However, this assumes that duplicate documents are included without replacing relevant documents, which is possible if k is chosen based on a relevance threshold.[20] Another trend is that models do not appear overly sensitive to minor token-level changes (exemplified by backtranslation) *relative* to the other perturbations, further motivating our focus on document-level errors.

Models may not consider all documents when summarizing In the random setting, even small amounts of deletion lead to large drops in summarization performance. Conversely, deletion has surprisingly little impact on performance in the oracle setting until a majority of documents (>60%) have been removed. These results have two non-mutually exclusive explanations. First, that models only consider some input documents when summarizing. Second, that reference summaries cover only a fraction of input document content, which is corroborated by recent work (Wolhandler et al., 2022). We note that this trend was less pronounced in cases of weaker summarization performance.

Document order does not generally impact summarization performance As far as we

[19] These results were chosen because they are exemplary of the main trends we observed across all model and dataset pairs. Please see Appendix I for the complete set of results.

[20] In this case, duplicate documents would obtain the same relevance score and would not take the place of other non-duplicate documents with the same or higher relevance score

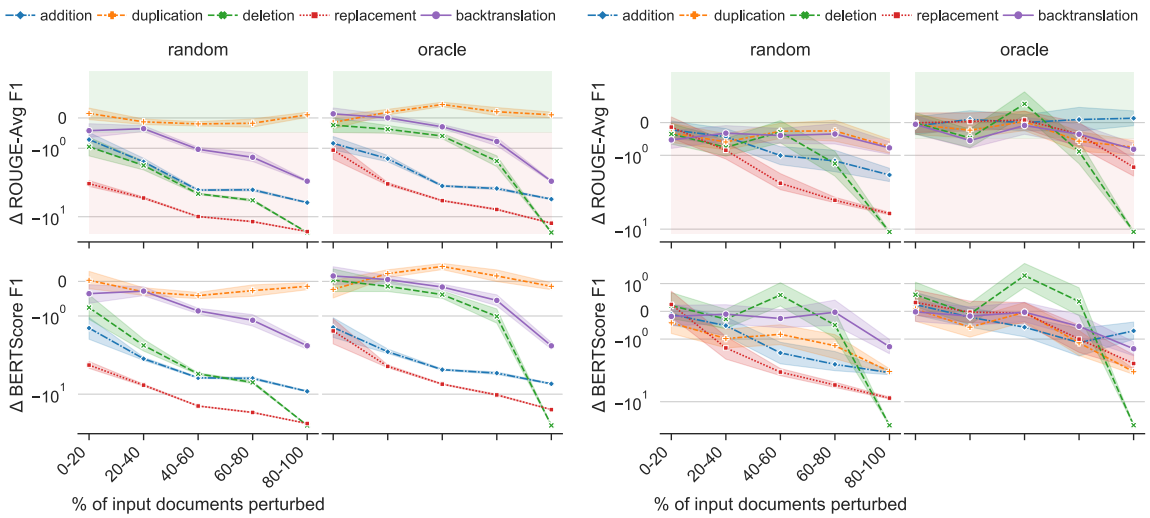

Figure 4: Results of the perturbation experiments on Multi-News (left) and Cochrane (right). Mean change in summarization performance plotted against the percent of perturbed input documents. Values above -0.49 ROUGE are shaded in **green**, and values below in **red**, the average difference in summarization performance reported in *CL conferences. Y-axis is displayed in symlog scale. 68% confidence intervals (CI) are plotted as error bands.

know, prior work has not investigated whether multi-document summarizers are sensitive to input document order. Although input documents are generally considered unordered, they are usually concatenated before passing them to a summarizer. To determine if models are sensitive to this order, we sorted the input documents of each dataset *before* concatenation and re-evaluated the summarizers. We investigate two ordering strategies:

- **Random**: Shuffle documents randomly.
- **Oracle**: Sort documents by similarity to the reference summary. This is motivated from two perspectives: (1) prior work has found that transformers are biased toward earlier tokens in their inputs (Hofstätter et al., 2021), so we might expect improved performance by placing the most similar content to the reference summary first, (2) from an IR perspective, a strong retriever would assign a higher rank to the most relevant documents, and we might choose to input documents to our summarizer based on this order.

In our experiments, we find no significant difference (paired t-test, p = 0.01) in summarization performance for any model-dataset pair, *except* in the case of WCEP-10 (see Appendix J). Here we find that both models we evaluate (PRIMERA and LSG-BART) are negatively affected by random sorting. One possible explanation is that, due to how WCEP-10 was constructed, the documents are (partially) sorted in order of relevance (see Appendix A). Models trained on this dataset may have learned to exploit this, e.g., by assigning more weight to earlier documents in the input. After randomly shuffling input documents, this learned heuristic would no longer hold, and summarization performance might drop accordingly.

8 Related Work

Query-focused MDS In query-focused MDS (QMDS) (Wang et al., 2013; Feigenblat et al., 2017; Xu and Lapata, 2020; Pasunuru et al., 2021a), a query or topic statement is provided alongside the input documents and used to guide summarization. For example, extractive QMDS methods use query relevance to select the sentences that will form the summary. However, ground-truth input documents are still provided, and no retrieval from a document index is performed. In this work, we propose and investigate a more realistic scenario where, given only the query, the input documents must be retrieved from a large document index. We also note that existing query-focused MDS datasets (e.g. DUC 2005, 2006 & 2007) are extremely small (on the scale of 10s of examples) and are therefore not suitable for the large-scale analysis we conduct.

Open-domain QA Our proposal for open-domain MDS mirrors a similar trend in the question answering (QA) literature. While earlier research focused on answering a question provided a text passage (Rajpurkar et al., 2016, 2018), the now

predominant approach (open-domain QA) is to answer a question *without* providing this passage, usually by referencing an external knowledge source (e.g. Wikipedia). Even broader is the class of knowledge-intensive (KI) language tasks (Petroni et al., 2021), which include open-domain QA but also fact-checking and entity-linking. These tasks are defined by their requirement to access large, external knowledge sources and are commonly approached with a retrieve-then-*generate* framework using "retrieval-augmented" generation architectures (Guu et al., 2020; Lewis et al., 2020b; Borgeaud et al., 2022; Izacard et al., 2022b). Our retrieve-then-*summarize* approach is very similar (§2), except that the outputs are, on average, much longer than existing KI language tasks. Therefore, we think a particularly promising approach for future work is adapting retrieval-augmented generation architectures for open-domain MDS.

Previous attempts at open-domain MDS In Zhang et al. (2021), a method similar to our retrieve-then-summarize approach is proposed, using a pretrained dense passage retriever (DPR, Karpukhin et al. 2020) and $T5_{base}$ (Raffel et al., 2020) as summarizer.[21] The model is trained and evaluated on a dataset constructed from existing QMDS datasets. This dataset is small, with ∼90 training examples and ∼45 test examples, and does not appear to be publicly available. Here, we conduct a much larger-scale analysis on multiple datasets from different domains (each consisting of thousands or 10s of thousands of examples) and evaluate several of the top-performing multi-document summarizers currently available. We also extensively simulated document retrieval errors to probe their impact on summarization performance. Together, this allows us to draw much broader conclusions about open-domain MDS and to provide detailed practical advice for future work.

9 Limitations

Automated evaluation metrics may not correlate with human judgment Though established metrics such as ROUGE and BERTScore are imperfect (Deutsch et al., 2022), they are frequently used in the summarization literature, do correlate with aspects of summary quality, and are useful for comparing system-level performance, especially in scenarios such as ours where performance differences due to retrieval errors can be several points below the baseline. To validate our findings, we intend to conduct a human evaluation to better understand qualitative differences in summaries generated in the open-domain setting. The investigation of better automated MDS evaluation metrics is also an active field of research, and we hope to integrate novel and performant metrics in future work.

Results conflate dataset features and model performance Our evaluation conflates several issues beyond the relative performance of retrievers and summarizers. Dataset quality, the "multi-document-ness" of each dataset, and the lack of suitable evaluation metrics all contribute to noise in our results. For example, a dataset that may require only a subset of its input documents (as characterized by Wolhandler et al. 2022) would not be expected to respond to retrieval errors in the same way as a dataset that needs more of its input documents. By experimenting with multiple datasets, retrievers, and summarizers, as well as in the synthetic perturbation setting, we hope our results are more resilient to these confounders.

Fine-tuning retrievers may lead to better performance We experiment with retrievers in the zero-shot setting without fine-tuning on MDS datasets. Fine-tuning might improve performance and lead to smaller reductions in summarization performance. We leave the investigation of fine-tuning retrieval for open-domain MDS to future work.

10 Conclusion

This paper introduces open-domain MDS, a new task definition for multi-document summarization. This reformulation is more realistic and potentially more useful, enabling users to specify their intent with only a query. Open-domain MDS shares similarities with recent work in knowledge-intensive NLP, which homogenizes many tasks into a unified "retrieve-then-generate" approach. To enable further progress in the open-domain setting, creating high-quality MDS datasets annotated with both queries and reference summaries is necessary.

Acknowledgements

This research was enabled in part by support provided by the Digital Research Alliance of Canada (alliancecan.ca) and Compute Ontario (www.computeontario.ca).

[21] At the time of writing, this work is unpublished

Author Contributions

John Giorgi made most of the technical contributions, including dataset collection and processing, model implementation, and running experiments. John also contributed to project scoping and ideation, wrote the paper with feedback from everyone, and led the project in general. Luca, Kyle, Lucy, and Arman were project mentors, contributing equally to project scoping and experimental design and providing the core ideas and direction throughout the course of the project and paper writing. Additionally, Luca made technical contributions to model implementation. Bo and Gary provided high-level feedback and advice.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. Re-examining system-level correlations of automatic summarization evaluation metrics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6038–6052, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. MSˆ2: Multi-document summarization of medical studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7494–7513, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Guy Feigenblat, Haggai Roitman, Odellia Boni, and David Konopnicki. 2017. Unsupervised query-focused multi-document summarization using the cross entropy method. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 961–964. ACM.

Tim Fischer, Steffen Remus, and Chris Biemann. 2022. Measuring faithfulness of abstractive summaries. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 63–73, Potsdam, Germany. KONVENS 2022 Organizers.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the*

Association for Computational Linguistics, pages 1302–1308, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Sebastian Hofstätter, Aldo Lipani, Sophia Althammer, Markus Zlabinger, and Allan Hanbury. 2021. Mitigating the position bias of transformer models in passage re-ranking. *ArXiv*, abs/2101.06980.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane A. Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Few-shot learning with retrieval augmented language models. *ArXiv*, abs/2208.03299.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254, Online. Association for Computational Linguistics.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60:493–502.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summariza-*

tion Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Edward Ma. 2019. NLP Augmentation.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*.

Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1737–1751, Online. Association for Computational Linguistics.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online. Association for Computational Linguistics.

Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021a. Data augmentation for abstractive query-focused multi-document summarization. In *AAAI*.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021b. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *TREC*.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multilexsum: Real-world summaries of civil rights lawsuits at multiple granularities.

Luca Soldaini. 2022. PyTerrier Sentence Transformers.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Byron C Wallace, Sayantan Saha, Frank Soboczenski, and Iain J Marshall. 2021. Generating (factual?) narrative summaries of RCTs: Experiments with neural multi-document summarization. *AMIA Summits Transl. Sci. Proc.*, 2021:605–614.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1384–1394, Sofia, Bulgaria. Association for Computational Linguistics.

J Beau W Webber. 2012. A bi-symmetric log transformation for wide-range data. *Measurement Science and Technology*, 24(2):027001.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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, Online. Association for Computational Linguistics.

Ruben Wolhandler, Arie Cattan, Ori Ernst, and Ido Dagan. 2022. How "multi" is multi-document summarization?

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Linzi Xing, Wen Xiao, and Giuseppe Carenini. 2021. Demoting the lead bias in news summarization via alternating adversarial learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 948–954, Online. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3632–3645, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *ArXiv*, abs/2106.11520.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Weijia Zhang, Svitlana Vakulenko, Thilina Rajapakse, and Evangelos Kanoulas. 2021. Scaling up query-focused summarization to meet open-domain question answering. *ArXiv preprint*, abs/2112.07536.

A Dataset Details

All datasets were managed in the HuggingFace Datasets library (Lhoest et al., 2021). The examples of each dataset consist of an input document set, D and a human-written reference summary, S. Multi-XScience and MSˆ2 each have an additional input that is always provided (and never retrieved or perturbed), the target articles abstract and the target reviews background section. Below, we provide detailed descriptions of each dataset:

- **Multi-News** (Fabbri et al., 2019): Consists of news articles and summaries collected from `www.newser.com`. There are 44,972 examples in the train set and 5622 examples in the test set. Each example contains between 1 and 10 documents, with a mean of ∼2.7.
- **WCEP-10**: Consists of news articles and summaries collected from the Wikipedia Current Events Portal (WCEP[22]). WCEP-10[23] sub-samples the top 10 most relevant documents from the original WCEP dataset (Gholipour Ghalandari et al., 2020). There are 8158 examples in the train set and 1022 examples in the test set. Each example contains between 1 and 10 documents, with a mean of ∼9.1.
- **Multi-XScience** (Lu et al., 2020): The target summary of each example is the related works section of a scientific article, and the input documents are the abstracts of the articles this section cites. Also included is the target article's abstract. There are 30,369 examples in the train set and 5093 examples in the test set. Each example contains between 1 and 20 documents, with a mean of ∼4.1.
- **MSˆ2** (DeYoung et al., 2021): The target summary is a few sentences from a biomedical systemic review which summarize the main findings. The input documents are the included studies for that review. Also included is the target reviews background section. There are 14,188 examples in the train set and 2021 examples in the validation set. Each example contains between 1 and 401 documents, with a mean of ∼23.2.
- **Cochrane** (Wallace et al., 2021): Similar to MSˆ2, except a background statement is not included as input. There are 3752 examples in the train set and 470 examples in the validation set. Each example contains between 1 and 537 documents, with a mean of ∼10.9.

B Retrieval Details

Document retrieval and evaluation are conducted in the PyTerrier library (Macdonald and Tonellotto, 2020). In Table 6, we present the retrieval performance on the train, validation and test split for each dataset, retriever, and top-k strategy. Below, we provided detailed descriptions of all retrievers:

- **BM25** (Robertson et al., 1994): Like other sparse retrievers, BM25 represents queries and documents as sparse vectors, where each element of a vector corresponds to a term in the vocabulary. BM25 is a widely used weighting scheme that extends TF-IDF (Jones, 2004) to account for document length and term-frequency saturation. We use BM25 via PyTerrier with the default settings.
- **Contriever** (Izacard et al., 2022a): Contriever is an unsupervised dense retriever that uses a bi-encoder architecture. Documents and queries are encoded independently using the same BERT model (Devlin et al., 2019), and the final embedding is obtained by mean-pooling over the hidden representations of the model's last layer. The relevance score between a query and a document is the dot product of their embeddings. Specifically, we use contriever-msmarco,[24] which has been fine-tuned on the MS MARCO dataset (Campos et al., 2016). We use Contriever via the PyTerrier Sentence Transformers plugin (Soldaini, 2022) with the default settings.

C Model Details

All models are implemented in PyTorch, and pretrained weights are obtained from the HuggingFace Transformers library (Wolf et al., 2020). Below, we provide detailed descriptions of all models:

- **LED** (Beltagy et al., 2020): LED replaces full self-attention with local windowed attention and global attention mechanisms that scale

[22] `https://en.wikipedia.org/wiki/Portal:Current_events`

[23] `https://huggingface.co/datasets/ccdv/WCEP-10`

[24] `https://huggingface.co/facebook/contriever-msmarco`

Table 6: Retrieval performance. The precision and recall at k for each retriever and top-k strategy is reported. The index for each dataset is the set of all documents in the train, validation and test sets; the reference summaries are used as queries, except for MSˆ2, where we use the provided background section. The Cochrane test set is blind, so we do not have access to the reference summaries to use as queries and therefore do not evaluate on the test set.

				Train		Validation		Test	
Dataset	Retriever	Retriever Type	Top-k Strategy	P@K	R@K	P@K	R@K	P@K	R@K
Multi-News	BM25	sparse	max(10)	0.22	0.83	0.22	0.82	0.22	0.82
			mean (3)	0.64	0.74	0.64	0.74	0.64	0.74
			oracle	0.75	0.75	0.75	0.75	0.75	0.75
	Contriever	dense	max	0.21	0.80	0.21	0.79	0.21	0.80
			mean	0.59	0.69	0.59	0.69	0.59	0.70
			oracle	0.69	0.69	0.69	0.69	0.69	0.69
WCEP-10	BM25	sparse	max (10)	0.59	0.66	0.60	0.63	0.63	0.67
			mean (9)	0.62	0.62	0.63	0.60	0.66	0.64
			oracle	0.64	0.64	0.63	0.63	0.67	0.67
	Contriever	dense	max	0.60	0.66	0.60	0.64	0.63	0.67
			mean	0.62	0.63	0.63	0.60	0.66	0.63
			oracle	0.65	0.65	0.63	0.63	0.66	0.66
Multi-XScience	BM25	sparse	max (20)	0.05	0.41	0.06	0.40	0.06	0.40
			mean (4)	0.16	0.27	0.16	0.26	0.16	0.27
			oracle	0.22	0.22	0.22	0.22	0.23	0.23
	Contriever	dense	max	0.06	0.38	0.06	0.38	0.06	0.38
			mean	0.16	0.24	0.16	0.24	0.16	0.24
			oracle	0.20	0.20	0.20	0.20	0.21	0.21
MSˆ2	BM25	sparse	max (25)	0.17	0.26	0.16	0.22	0.17	0.22
			mean (17)	0.21	0.22	0.18	0.18	0.20	0.18
			oracle	0.22	0.22	0.18	0.18	0.19	0.19
	Contriever	dense	max	0.19	0.29	0.18	0.25	0.19	0.26
			mean	0.23	0.24	0.21	0.21	0.23	0.21
			oracle	0.24	0.24	0.21	0.21	0.22	0.22
Cochrane	BM25	sparse	max (25)	0.17	0.55	0.17	0.57	–	–
			mean (9)	0.30	0.42	0.31	0.44	–	–
			oracle	0.38	0.38	0.40	0.40	–	–
	Contriever	dense	max	0.20	0.63	0.20	0.64	–	–
			mean	0.34	0.48	0.35	0.49	–	–
			oracle	0.45	0.45	0.44	0.44	–	–

linearly with input sequence length, allowing for efficient processing of inputs up to 16K tokens. Its parameters are initialized with the pretrained parameters of BART (Lewis et al., 2020a), its positional embeddings with 16 copies of BART's 1K position embeddings. The model is fine-tuned on MDS datasets in a supervised fashion.

- **PEGASUS** (Zhang et al., 2020a): PEGASUS is pretrained using a novel Gap Sentences Generation (GSG) objective, where whole sentences from each document are masked, and concatenated to form a pseudo-summary. The model is then fine-tuned on MDS datasets in a supervised fashion.
- **PRIMERA** (Xiao et al., 2022): Extends the GSG objective with a novel masking strategy explicitly designed for multi-document inputs and pre-trains on a corpus of multi-document examples. The model is then fine-tuned on MDS datasets in a supervised fashion or used in a zero-shot setting.
- **LSG-BART**[25]: Like LED, LSG-BART replaces full self-attention with a sparsified version, dubbed Local-Sparse-Global (LSG) attention, to allow for efficient processing of long inputs. It is initialized with the pretrained parameters of BART and then fine-tuned on

[25] https://huggingface.co/ccdv/lsg-bart-base-4096

Table 7: Reported versus reproduced ROUGE-1/2/L scores for each model-dataset pair evaluated in the main paper. We also report zero-shot performance on select datasets for PRIMERA. *Fine-tuned by us.

Dataset \ Model	Reported				Reproduced			
	PRIMERA	PEGASUS	LED-base	LSG-BART-base	PRIMERA	PEGASUS	LED-base	LSG-BART-base
Multi-News	49.9/21.1/25.9	47.5/18.7/24.9	–	–	49.3/20.3/25.4	48.2/20.1/25.4	–	–
↪ *zero-shot*	42.0/13.6/20.8	–	–	–	39.7/11.9/19.2	–	–	–
WCEP	46.1/25.2/37.9	–	–	46.0/24.2/37.4	45.1/24.7/36.7	–	–	45.9/24.1/37.2
↪ *zero-shot*	28.0/10.3/20.9	–	–	–	31.3/10.7/22.2	–	–	–
Multi-XScience	31.9/7.4/18.0	–	–	–	31.7/6.1/17.1	–	–	–
↪ *zero-shot*	29.1/4.6/15.7	–	–	–	27.0/3.9/14.6	–	–	–
MSˆ2*	–	–	26.4/8.0/19.6	–	–	–	28.5/9.5/20.9	–
Cochrane*	–	–	23.9/6.6/17.6	–	–	–	26.9/6.9/18.4	–

MDS datasets in a supervised fashion.

C.1 Reproducing Reported Scores

Before experimentation, we attempt to reproduce the reported scores of each MDS model. The results are provided in Table 7. In general, we can reproduce the reported ROUGE scores (and sometimes even improve upon them); however, in a few cases, there are differences as large as ∼3 ROUGE, with the largest differences being observed for PRIMERA, particularly in the zero-shot setting.

D Extended Results from: section 5

In §5, we presented the results from our open-domain MDS experiments for the sparse retriever (BM25) only. The results from the dense retriever (Contriever) were comparable and exhibited the same general trends. We present the complete dense retriever results in Table 8.

E Summarization Baselines

In Table 5, we presented the scores of several simple heuristic baselines. Detailed descriptions of each baseline follow:

- **Random (length-matched) summary**: For each example, take the summary to be the reference summary of *another* example from the same dataset that is the same (or similar) length as the examples reference summary. This provides us with coherent (but likely irrelevant) summaries of approximately the correct length from the same domain.

- **All lead**: For each example, take the summary to be the concatenation of the first sentence from each input document. This is motivated by the notion of a *lead bias*, namely that in many summarization datasets (particularly those comprised of news articles), sentences at the beginning of a document are more likely to contain information that appears in the reference summary (Nenkova et al., 2011; Hong and Nenkova, 2014; Xing et al., 2021).

- **Oracle document**: For each example, take the summary to be the input document with the highest ROUGE-1 F1 score with that example's reference summary. This provides us with relevant (but likely incomplete) summaries with high token overlap. A high score may indicate that a dataset is less "multi" (Wolhandler et al., 2022).

- **Oracle lead**: The first sentence of the oracle document (see above). In the case of MSˆ2 and Cochrane, this is the title of the oracle document.

- **Background/abstract**: Applies only to MSˆ2 and Multi-XScience. For each example, take the summary to be the additional input from MSˆ2 (target reviews background section) and Multi-XScience (target articles abstract).

F Training with Retrieval

In Figure 2, we presented the results of our experiments fine-tuning existing summarizers in the open-domain setting. We fine-tuned and evaluated the best-performing model from each dataset: PRIMERA for Multi-News and Multi-XScience, LSG-BART-Base for WCEP-10 and LED-Base for MSˆ2 and Cochrane. Each model was fine-tuned for an additional 3 epochs (we found that additional training made little difference) using the original training hyperparameters. All models were fine-tuned with the AdamW optimizer (Loshchilov and Hutter, 2019) in the HuggingFace Transformers library. The learning rate was linearly increased for the first 10% of training steps and linearly decayed to zero afterward.

Table 8: Results of the open-domain MDS experiments using a dense retriever (Contriever). Difference between a summarizers performance on the ground-truth input documents and performance when the documents were retrieved is shown. Statistically significant results are underlined (paired t-test, p = 0.01).

		Retrieval			Summarization			
Dataset	Model	Top-k Strategy	P@K	R@K	ROUGE-Avg F1	Δ ROUGE-Avg F1	BERTScore F1	Δ BERTScore F1
Multi-News	PRIMERA	max (10)	0.21	0.80	31.66	-7.77	31.78	-10.47
		mean (3)	0.59	0.70	–	-3.31	–	-4.60
		oracle	0.69	0.69	–	-2.20	–	-3.07
	PEGASUS	max	–	–	31.23	-8.69	29.88	-10.88
		mean	–	–	–	-2.65	–	-3.45
		oracle	–	–	–	-1.76	–	-2.28
WCEP-10	PRIMERA	max (10)	0.63	0.67	35.50	+0.10	48.26	+0.90
		mean (9)	0.66	0.63	–	-0.14	–	+0.68
		oracle	0.66	0.66	–	+0.29	–	+0.86
	LSG-BART-base	max	–	–	35.76	-0.56	48.17	+0.26
		mean	–	–	–	-0.96	–	+0.10
		oracle	–	–	–	-0.15	–	+0.66
Multi-XScience	PRIMERA	max (20)	0.06	0.38	18.31	-0.45	10.57	-0.96
		mean (4)	0.16	0.24	–	-0.81	–	-0.96
		oracle	0.21	0.21	–	-0.28	–	-0.37
MSˆ2	LED-base	max (25)	0.18	0.25	19.66	-0.43	22.74	-0.70
		mean (17)	0.21	0.21	–	-0.37	–	-0.64
		oracle	0.21	0.21	–	-0.32	–	-0.38
Cochrane	LED-base	max (25)	0.20	0.64	17.39	-0.94	23.12	-2.77
		mean (9)	0.35	0.49	–	-0.37	–	-0.93
		oracle	0.44	0.44	–	+0.25	–	+0.71

G Document Retrieval Error Analysis

In Figure 5, we tally the total number of errors made by the sparse (BM25) and dense (Contriever) retrievers on each dataset we investigated. For each example, we count an *addition* (i.e. erroneous inclusion) each time a document not in the ground-truth input document set is retrieved, a *deletion* (i.e. erroneous exclusion) each time a ground-truth document is not retrieved and a *replacement* (i.e. an erroneous swap of a relevant document for an irrelevant one) each time both one addition and one deletion occur.

H Backtranslation

In the main paper, we use backtranslation to create token-level perturbations. The procedure involves selecting one or more documents from the input set and translating them to another language and back again, often creating small, token-level changes like paraphrasing and synonym substitution (this is sometimes called "round-trip translation", or RTT). We choose to translate documents to and from Danish, as there exists freely available and high-performing EN→DA and DA→EN machine translation (MT) models. In particular, we use the models provided by the Language Technology Research Group at the University of Helsinki (Tiedemann and Thottingal, 2020). We implement backtranslation using the `nlpaug` library (Ma, 2019). In Figure 6, we provide an example of a backtranslated document demonstrating synonym substitution (e.g. "highly"→"very"), paraphrasing (e.g. "said the surviving ones"→"said that the survivors") and grammatical errors (e.g. "14 critically endangered black rhinoceros has died").

I Extended Results from: section 7

In §7, we presented the results from our experiments simulating document retrieval errors for two model-dataset pairs that exemplified the main trends in the rest of the results. In figures 7-13, we present the complete results for all model-dataset pairs.

J Sorting Perturbation Results

In Table 9, we present the tabulated results from the sorting perturbation experiments. See §7 for more details on the experimental procedure and §7.2 for an analysis of the results.

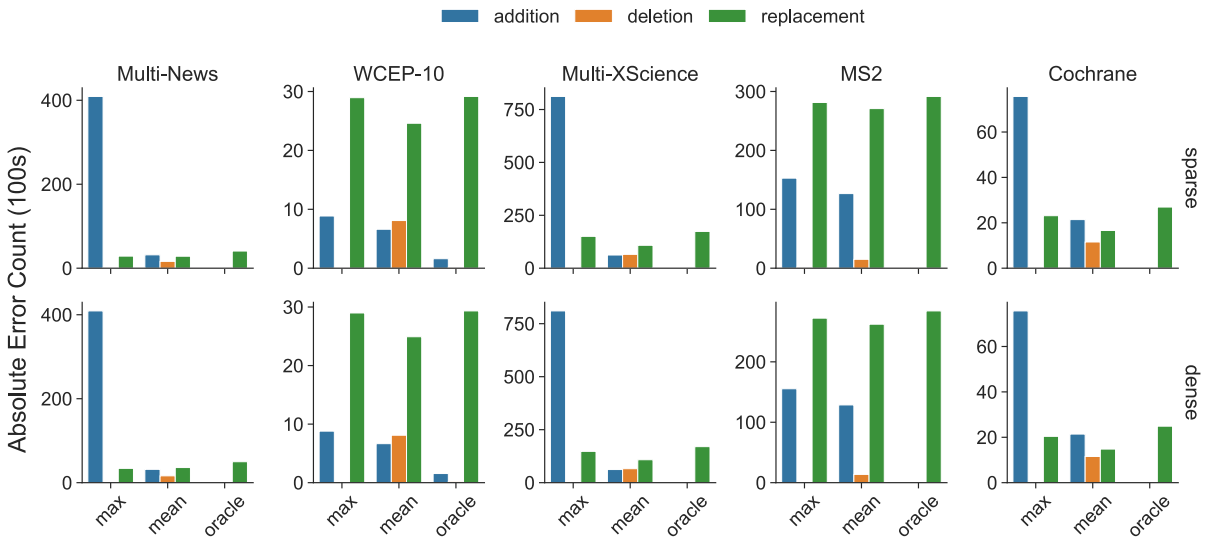

Figure 5: Absolute error counts for different retrieval systems (sparse and dense) and top-k selection strategies (max, mean, oracle). For each example in a given dataset, a retrieved document that does not exist in the ground-truth input document set is counted as an *addition* and a ground-truth document that was not retrieved as a *deletion*. Instances of one addition and one deletion are counted as a *replacement*.

Original Document

Relocation of endangered animals carries risks but loss of half of them is highly unusual. Eight out of 14 critically endangered black rhinos have died after being moved to a reserve in southern Kenya, wildlife officials have revealed, in what one conservationist described as "a complete disaster". Preliminary investigations pointed to salt poisoning as the rhinos tried to adapt to saltier water in their new home, the Kenyan Ministry of Tourism and Wildlife said in a statement. It suspended the moving of other rhinos and said the surviving ones were being closely monitored.

Backtranslated Document

Movement of endangered animals carries risks, but the loss of half of them is very unusual. Eight out of 14 critically endangered black rhinoceros has died after being moved to a reserve in southern Kenya, wildlife officials have revealed in what a conservation expert described as a complete disaster. Preliminary studies pointed to salt poisoning when the rhinoceroses tried to adapt to salt water in their new home, the Kenyan Ministry of Tourism and Animal Health said in a statement. It suspended the movement of other rhinoceros and said that the survivors were being closely monitored.

Figure 6: Graphical depiction of the backtranslation perturbation. A truncated document from the Multi-News (Fabbri et al., 2019) dataset is shown, and changes after backtranslation are highlighted.

Table 9: Results of the sorting perturbation experiments. Difference between a summarizers performance on the ground-truth input documents and performance when the documents were perturbed is shown. Statistically significant results are underlined (paired t-test, p = 0.01).

				Δ ROUGE-Avg F1		Δ BERTScore F1	
Dataset	Model	ROUGE-Avg F1	BERTScore F1	Random	Oracle	Random	Oracle
Multi-News	PRIMERA	31.66	31.78	+0.06	+0.00	+0.02	+0.02
	PEGASUS	31.23	29.88	-0.05	+0.04	-0.05	+0.16
WCEP-10	PRIMERA	35.50	48.26	-0.86	+0.11	-0.55	+0.57
	LSG-BART-base	35.76	48.17	-0.98	-0.18	-0.62	+0.38
Multi-XScience	PRIMERA	18.31	10.57	+0.07	-0.04	+0.13	-0.03
MS2	LED-base	19.66	22.74	+0.09	+0.24	+0.00	-0.01
Cochrane	LED-base	17.39	23.12	-0.41	-0.32	-0.42	+0.06

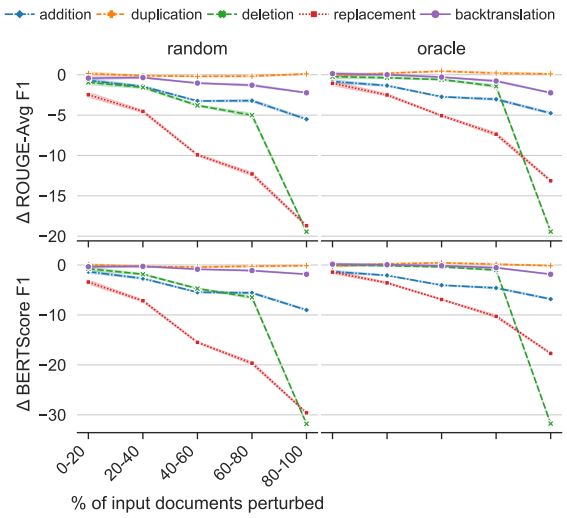

Figure 7: Results of the perturbation experiments on the Multi-News test set using PRIMERA. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

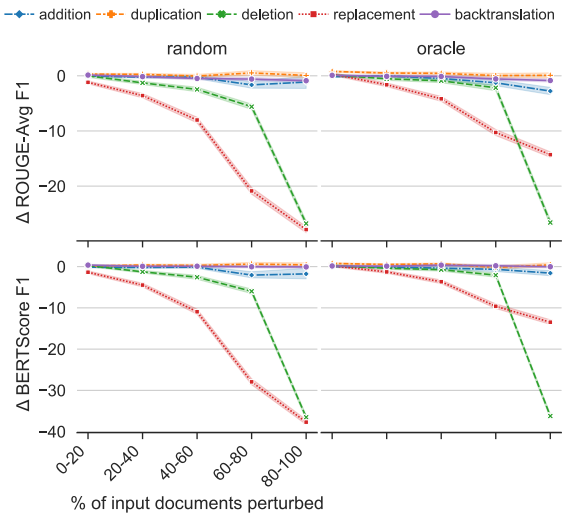

Figure 9: Results of the perturbation experiments on the WCEP-10 test set using PRIMERA. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

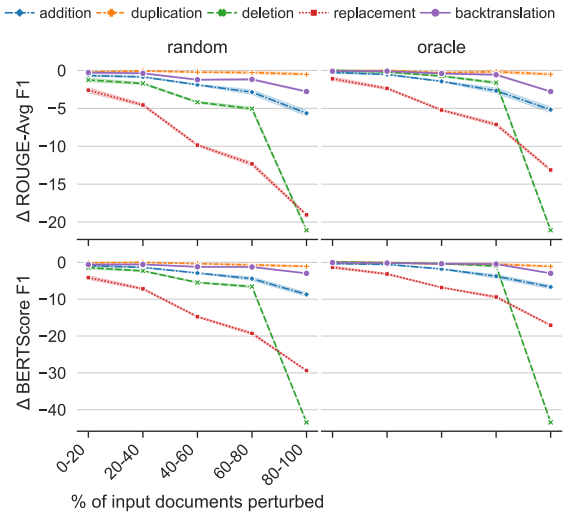

Figure 8: Results of the perturbation experiments on the Multi-News test set using PEGASUS. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

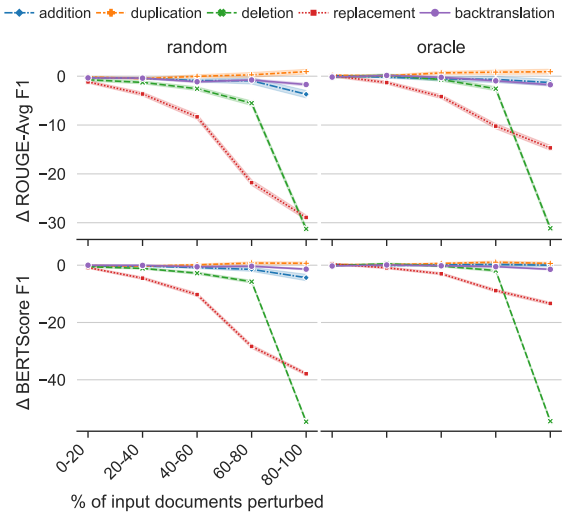

Figure 10: Results of the perturbation experiments on the WCEP-10 test set using LSG-BART-base. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

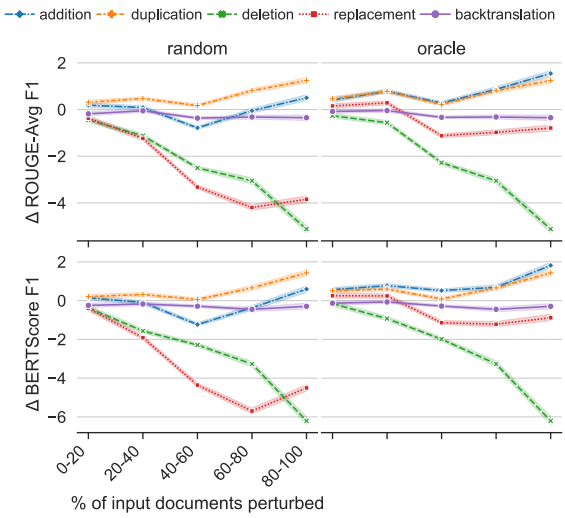

Figure 11: Results of the perturbation experiments on the Multi-XScience test set using PRIMERA. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

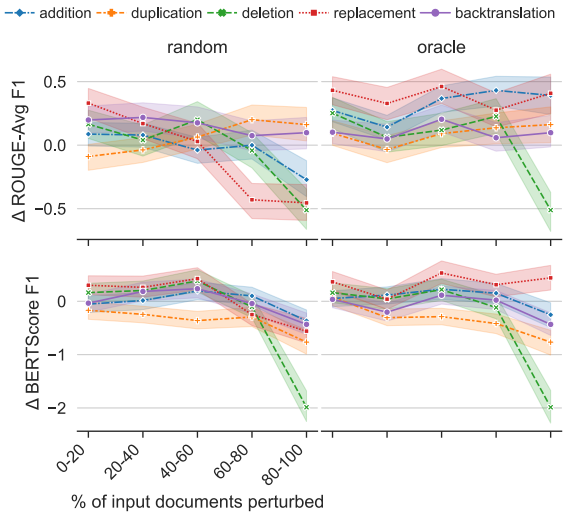

Figure 12: Results of the perturbation experiments on the MSˆ2 validation set using LED-base. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.

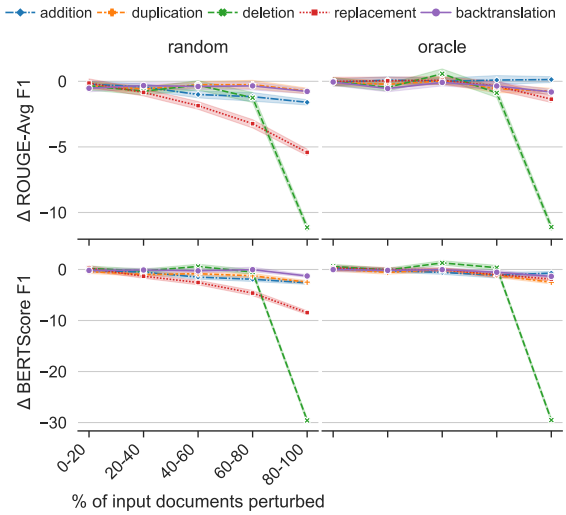

Figure 13: Results of the perturbation experiments on the Cochrane validation set using LED-base. Mean change in summarization performance plotted against percent of perturbed input documents. 68% confidence intervals (CI) are plotted as error bands.