Cross-Document Event-Keyed Summarization

Anonymous ACL submission

Abstract

Event-keyed summarization (EKS) requires generating a summary about a specific event described in a document, given the document and an event representation extracted from it. In this work, we extend EKS to the cross-document setting (CDEKS), in which summaries must synthesize information from accounts of the same event given by *multiple* sources. We introduce SEAMUS (**S**ummaries of **E**vents **A**cross **Mu**ltiple **S**ources), a high-quality dataset for CDEKS based on an expert reannotation of the FAMUS dataset for cross-document argument extraction. We present a suite of baselines on SEAMUS, covering both smaller, fine-tuned models, as well as zero- and few-shot prompted LLMs, along with detailed ablations of model performance, and a human evaluation study—showing SEAMUS to be a valuable benchmark for this new task.

1 Introduction

Delivering useful and actionable information about events requires the ability not only to extract relevant, user-specified information from documents, but also to surface that information in a readable form. Drawing on this observation, Gantt et al. (2024) recently proposed *event-keyed summarization* (EKS), a task that requires generating a summary of a *particular* event, given a document and an event representation extracted from it. EKS thus seeks to make good on both goals—reconciling the very specific information needs of information extraction (IE) end users with the much more generic outputs of traditional summarization models—in order to communicate *precise* information about a single event in a contextualized and *human-readable* form. EKS can thus be viewed as an event-centric form of controllable summarization (Fan et al., 2018), where the attributes to be controlled are the event and roles of interest.

However, the information available about an event from any *one* source is often incomplete; *full* understanding routinely requires finding and synthesizing information across multiple sources. Accordingly, this work extends EKS to the cross-document setting (CDEKS), drawing on, and enhancing, the FAMUS dataset for cross-document argument extraction (CDAE) to do so (Vashishtha et al., 2024). Our contributions are as follows:

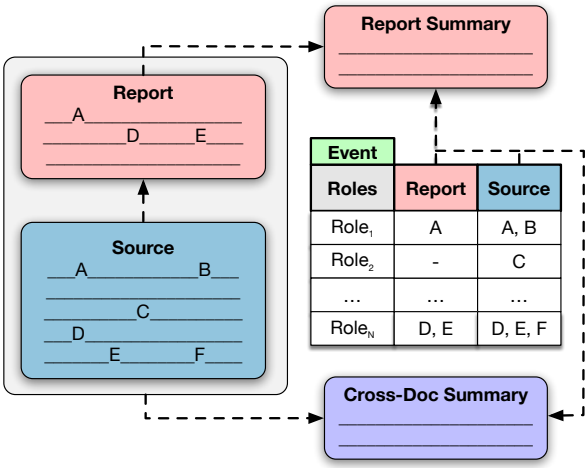

Figure 1: Illustration of the SEAMUS report and cross-document event-keyed summarization tasks. Letters represent event arguments.

1. We collect and release an expert reannotation of the FAMUS CDAE dataset, correcting the existing crowdsourced annotations.
2. Based on (1), we collect and release SEAMUS, an expert-annotated dataset of single- and cross-document event-keyed summaries, and the first ever dataset for CDEKS.
3. We present a suite of baselines on SEAMUS using both smaller, fine-tuned models and prompted LLMs, showing CDEKS to be challenging relative to single-document EKS.
4. We conduct detailed ablations and a human evaluation, giving a fine-grained picture of the task's demands and of models' capabilities.[1]

[1] Data is in the supplementary materials. Code will be made available on GitHub.

2 Background

FAMuS (Vashishtha et al., 2024) is a dataset of short (1- to 3-sentence) English Wikipedia passages (*reports*) paired with much longer, genre-diverse English *source* documents cited by those reports.[2] FAMuS supports two tasks: (1) *Source Validation* (SV), where the goal is to determine whether a candidate source document is *valid* for—i.e. describes the same event as—an event identified in a particular report; and (2) *Cross-Document Argument Extraction* (CDAE), which entails extracting arguments for an identified event from both the report and a valid source document. Our dataset builds on the FAMuS CDAE data, which contains 1,253 report-source document pairs (split 3:1:1 across train, dev, and test), and which annotates arguments of the same target event for each document in a pair. Events and arguments are annotated using a subset of the FrameNet ontology that is restricted to frames denoting events, states, or processes (Baker et al., 1998). A single, maximally "informative" mention is annotated for each argument, where proper names are considered more informative than nominal expressions, which in turn are more informative than pronouns (cf. Li et al., 2021). Since the account of the target event in either the report or source often spans multiple sentences, annotated arguments are often also distributed across sentences.

Event-Centric Summarization In introducing EKS, Gantt et al. (2024) released MUCSUM, a dataset for the task based on the classic MUC-4 template filling dataset (Sundheim, 1992). MUCSUM contains abstractive event-keyed summaries of at most three sentences for each event template in MUC-4, written so as to faithfully communicate the role of each template argument, plus any minimal additional context required for the summary to provide a standalone account of the event. Gantt et al. present baseline results on MUCSUM using fine-tuned and zero-shot prompted models, and also conduct a human evaluation of model outputs, which inspires our own ablation study (§4.5).

Beyond this, there has been relatively little work on event-centric summarization. S Hussain et al. (2022) use event-related keywords extracted from documents to condition summarization on CCN/DailyMail (Nallapati et al., 2016) and XSUM (Narayan et al., 2018), and integrate a novel event-oriented attention mechanism into BART (Lewis et al., 2020) to obtain summaries that aim to cover the *full* set of events each document reports on.

Second, Vallurupalli et al. (2022) introduce the POQue dataset, which features a wealth of annotations that aim to characterize complex events described in stories via their subevent structure and the changes undergone by individual participants. Among these annotations are "process summaries," which give high-level descriptions of a complex event as a whole, as well "change summaries," which describe the changes experienced by a participant as a result of the event.

Two key differences between these works and our own are (1) our focus on *single* events as opposed to multiple; and (2) our reliance on an event ontology—meant to represent an end user's information needs—to control summary contents.

Multi-Document Summarization CDEKS is an event-centric multi-document summarization (MDS) task. Work on MDS has pursued a wide variety of goals, including synthesizing reviews (Ganesan et al., 2010; Chu and Liu, 2019, *i.a.*), summarizing dialogues (Chen et al., 2021; Kraaij et al., 2005, *i.a.*), and distilling news articles (notably, via the Document Understanding[3] and Text Analysis Conferences[4]; cf. Fabbri et al., 2019).

Two lines of MDS work are especially relevant. First, the CDEKS task may be viewed as an instance of what Ma et al. (2022) call *query-oriented MDS*, in which a query expressing the kind of information to be summarized is provided as input alongside the documents.[5] Queries in prior work take the form of natural language statements—e.g. Wikipedia article titles (Liu and Lapata, 2019)—or questions—e.g. users' search queries (Pasunuru et al., 2021). In contrast, our queries are structured event representations, as may be generated by an event extraction system.

Second, among MDS datasets, our SEAMuS dataset is arguably most similar to WCEP (Gholipour Ghalandari et al., 2020) and WikiSum (Liu et al., 2018) in drawing on Wikipedia passages and their source articles, but differs from these in focusing on single events (WikiSum) and in using structured event representations to guide summary generation (both).[6]

[2] All documents are from MegaWika (Barham et al., 2023).

[3] https://duc.nist.gov/

[4] https://tac.nist.gov/publications/index.html

[5] This contrasts with most MDS tasks, in which a summary's focus is defined implicitly by the topical similarity of the input documents (e.g. as determined by clustering).

[6] Cf. Schuster et al. (2024) for a related QA task.

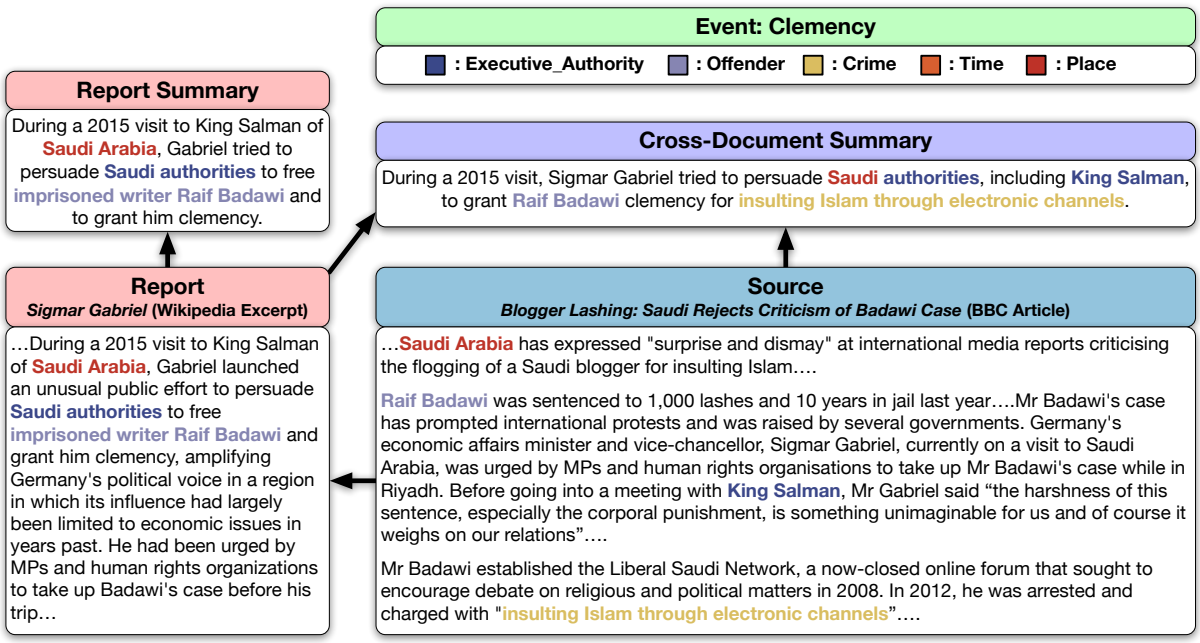

Figure 2: An example from our SEAMUS dataset. **Report** documents (bottom left) are Wikipedia passages that describe some event (top right) and that cite a longer (non-Wikipedia) **source** article (bottom right) as evidence, with event arguments annotated in both documents. SEAMUS features simple summaries of these events based on *only* the report (top left) as well as enriched, cross-document summaries based on *both* the report and its source, which are typically able to recover additional information about the event (here, the CRIME).

3 Annotation

Annotation of SEAMUS was divided into two phases. In the first phase, abstractive **report summaries** were written for each event in FAMUS based only on its *report* document, after which they were annotated for event arguments (§3.1). In the second phase, abstractive **cross-document summaries** were written for each event based *jointly* on its report and source documents, and were then annotated for event arguments as in the first phase (§3.2). In both phases, annotators were instructed to amend spurious, missing, or otherwise incorrect argument annotations in the report or source document before writing their summary. Thus, both phases involve (1) correcting existing FAMUS argument annotations; (2) writing a summary based on the corrected annotations; and (3) annotating arguments in the summary. The phases differ only in the documents on which the summaries are based (report only vs. report and source). All annotations were performed by authors of this work.[7]

3.1 Phase 1: Report Summaries

Similar to the summaries in MUCSUM (§2), the report summaries in SEAMUS are concise summaries of an event as recounted in a single document (a FAMUS report) that aim to faithfully represent the role of each participant and to provide the minimum additional context needed to produce an accurate, standalone account of the event. Although the FAMUS report documents are already relatively short (typically, 2-3 sentences), they often discuss multiple events. Thus, the report summaries offer distilled descriptions focused on just *one* event from the report.

Three authors completed the Phase 1 annotation, with each summary and its arguments singly annotated. Items from the train split were randomly and evenly divided among these three authors; items from the dev and test splits were similarly divided between two of them. All items were provided in JSON files containing the following information for each example: (1) a unique example ID, (2) the FAMUS report text; (3) the FAMUS-annotated frame, trigger, and arguments of the target event from the report; and (4) definitions of the annotated frame and roles as given in FrameNet. Annotators were provided with detailed instructions written by the first author and completed a 10-example practice task before beginning the main annotation. Consistent with FAMUS, both the corrected report arguments and the report summary arguments were annotated as single, maximally informative mentions (see §2). Annotators were encouraged to

[7] Additional annotation details are in Appendix E.

	Report		Cross-Doc	
	Train	Dev	Train	Dev
Examples	759	253	759	253
Avg. Words	21.8	24.6	30.5	34.5
Avg. Sentences	1.0	1.0	1.2	1.2
Avg. Arguments	3.1	3.5	4.1	4.6

Table 1: Summary statistics for the SEAMUS report (left) and cross-document summaries (right). Test statistics intentionally omitted.

use the same mentions in their summaries as were annotated in the (corrected) report arguments, but were permitted to alter them in the summary in order to preserve clarity or naturalness. Annotations were validated to ensure that (1) they were shorter than the report they summarized and (2) the number of arguments for a given role matched between each report and its summary. All initially invalid annotations were then corrected.

3.2 Phase 2: Cross-Document Summaries

The *cross-document summaries* are intended as enriched versions of the report summaries, synthesizing details about the target event from both the report and the target event's source document.

Five of the authors completed the Phase 2 annotation, with all summaries and arguments singly annotated as in Phase 1. Items from all three splits were randomly and evenly distributed to the five annotators. Given the complexity of the Phase 2 task, annotation was performed in two parts, using adapted versions of Vashishtha et al.'s (2024) interface for FAMUS CDAE annotation.[8]

In the first part, annotators corrected FAMUS argument annotations in the source documents and then wrote the cross-document summary based *jointly* on the report and source texts and their corrected arguments. Annotators were encouraged to use the most informative mention of an argument across *both* the report and source documents, but again were allowed to make alterations for clarity.

In the second part, annotators annotated arguments in the summaries from the first part. As in Phase 1, all annotators were provided with detailed instructions and completed a 10-example practice annotation before doing the main task. Summary argument annotations were again validated for length and to ensure that they featured as many arguments for a given role as the maximum number annotated for that role between the report and source texts.

Summary statistics for both the report and cross-document summaries can be found in Table 1 and an example is shown in Figure 2. Both types of summary average roughly a sentence in length, though cross-document summaries tend to be longer and to have more arguments—consistent with the richer information they provide.

4 Experiments

4.1 Overview

Tasks We present experiments on both the report (§4.2) and cross-document (§4.3) summarization tasks. In the report task (single-document EKS), both the report and its annotated event are provided as input. The cross-document task (CDEKS) is analogous, but additionally includes the corresponding source document and its event annotation as input. These initial experiments condition on gold event structures, but we then consider the impact of degraded extractions on summary quality in §4.4. Finally, in §4.5, we briefly discuss some ablations on the input inspired by similar ones from Gantt et al. (2024), with full results in Appendix F.

Models We benchmark SEAMUS using models of two types. First, we consider several classic pretrained encoder-decoder models widely used for summarization: BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020), and T5 (Raffel et al., 2020), fine-tuning the large versions of all three on the SEAMUS training data. Second, we consider some of the latest proprietary LLMs, evaluated in both the zero- and few-shot settings: GPT-4o[9], GPT-4o Mini (GPT-4O M in Table 2)[10], Claude 3 Haiku (CLAUDE H)[11], and Claude 3.5 Sonnet (CLAUDE S)[12]. For the few-shot examples, we use the three examples from the train split whose frame matches that of the target example. Finally, we also give results for a *report baseline* (RB) that treats the report text itself as the predicted summary.

Metrics We report several standard summarization metrics, including ROUGE-1 ($\mathbf{R}_1$), ROUGE-2 ($\mathbf{R}_2$), and ROUGE-LCS F_1 scores ($\mathbf{R}_L$; Lin, 2004), as well as BERTScore F_1 (**BS**; Zhang et al., 2019).

[8] Interface source code was obtained from Vashishtha et al.. Screenshots are shown in Appendix E.

[9] https://openai.com/index/hello-gpt-4o/

[10] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[11] https://www.anthropic.com/news/claude-3-haiku

[12] https://www.anthropic.com/news/claude-3-5-sonnet

Given EKS's focus on producing summaries that recover *specific* pieces of information—as encoded by an event's roles—we report a couple of other metrics that evaluate this. First, we report CEAFREE (**CR**; Du et al., 2021), a form of argument F_1 that allows us to compare arguments extracted from a *predicted* summary against those in a *reference* summary, aligning arguments based on exact match.[13] Following Gantt et al. (2024), we train the event extraction model of Xia et al. (2021)[14] on SEAMUS and use it to extract arguments from the predicted summaries, constraining extraction to arguments that fill roles of the target event only.

The summaries in SEAMUS naturally also make claims about these arguments that reflect their role in the target event. To evaluate these claims' fidelity to the text, we report AlignScore (**A**; Zha et al., 2023), a learned metric that provides a score in $[0, 1]$ that indicates how well a claim (here, a summary) is supported by a given context (the report for the report task, and the concatenated report and source for the cross-document task).

4.2 Report Summarization

Setup As input for BART, PEGASUS, and T5, we provide the full report text concatenated with a linearized representation of the annotated report event that contains the frame name, the event trigger, and the role names, each followed by a list of the arguments annotated for that role. We train each model against a standard NLL objective w.r.t. the gold report summaries for a maximum of 30 epochs, using a patience of 5 epochs, with dev $\mathbf{R}_1$ as the stopping criterion.[15] For inference, we use beam search decoding with a beam size of 5 and a max of 256 tokens.

For the Claude and GPT models, our system prompt tasks the model with analyzing and summarizing a specific event. The user prompt provides more specific task instructions, followed by the full report text, and a description of the target event that includes (1) the frame name and definition from FrameNet; (2) the trigger; and (3) a bulleted list, where each item includes a role, its definition, and the arguments annotated for that role. In the few-shot setting, we format the three few-shot examples (see §4) the same way, but with the target summary shown at the end of each. We set temperature to 0.7 and the max new tokens to 256, leaving other API defaults unchanged.[16]

Results are shown in the left half of Table 2. First, we find that T5 obtains the best performance across all metrics, followed by PEGASUS and BART, with T5 exhibiting particularly strong results for **CR**, indicating its ability to accurately recover event arguments in its summaries. Second, the LLMs almost universally outperform the report baseline (RB)—even in the zero-shot setting (ZS), where Claude Sonnet generally sees the best results. Third, adding just three few-shot examples (FS) yields major gains over the zero-shot setting for all LLMs on all metrics. Even here, however, few-shot results still trail the best fine-tuned results (T5) by sizable margins on most metrics.

4.3 Cross-Document Summarization

Setup The setup for the cross-document task is similar to that of the report task, but adds the source text and its annotated event to the input alongside the report text and its event. As the source texts are full web articles, most are quite long (e.g. dev texts average almost 62 sentences and over 1,500 words). While this is no obstacle for the LLMs, the smaller models do not support contexts of this size. Thus, to enable a fair comparison across models, we apply a sentence retriever to the source, using the report text as a query to select the top k most relevant sentences to use as context.[17] We consider $k \in \{3, \ldots, 10\}$ and selected the maximum value such that $\geq 95\%$ of the resulting dev set contexts would fit untruncated in the input, yielding $k = 7$. We experimented with the dense retrievers `all-mpnet-base-v2` (based on MPNet; Song et al., 2020) and `e5-large-v2` (Wang et al., 2022), but obtained our best results with BM25 (Robertson et al., 2009), which we use in all following experiments.[18]

We use the same training and inference settings from §4.2, and refer the reader to Appendices B and C for more details on input formatting.

Results are shown in the right half of Table 2 and are qualitatively similar to those for the report task, with the fine-tuned models generally showing

[13] Appendix F has results with a soft variant of this metric.

[14] https://hub.docker.com/r/hltcoe/lome

[15] Details on training and input formats are in Appendix B.

[16] Appendix C has further details on models and prompts.

[17] This approach can also be justified from the perspective of the task itself, as typically only a modest portion of the source text concerns the event of interest.

[18] Models were evaluated on recall of annotated arguments in the retrieved contexts for the dev set for fixed k. At $k = 7$, BM25 recovered $\sim 76\%$ of annotated source arguments.

Model	S	Report						Cross-Document					
		$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**
RB	-	56.2	46.1	48.4	91.6	52.6	99.1	48.5	33.3	39.3	89.6	31.0	99.3
GPT-4o M	ZS	62.2	42.3	51.3	93.2	58.5	86.0	51.8	29.9	39.0	91.3	39.0	81.5
	FS	72.0	55.4	61.0	94.3	66.8	94.1	57.5	36.9	45.7	92.1	39.8	88.5
GPT-4o	ZS	64.0	45.2	53.0	93.2	61.4*	83.9	58.0*	36.4	45.8	92.2*	41.3*	86.6
	FS	72.5†	56.6†	62.3†	94.4	69.6†	94.7	61.2†	40.7†	49.4†	**92.7**†	42.7†	90.6
Claude H	ZS	64.8	46.2	54.7	93.4	58.8	84.9	57.7	36.9*	46.5	92.1	36.2	90.4
	FS	71.7	55.9	61.1	94.3	63.2	94.8	59.4	39.5	48.6	92.1	37.2	91.0
Claude S	ZS	67.4*	48.1*	56.5*	93.8*	61.1	93.0*	56.7	34.8	45.3	91.9	35.2	93.4*
	FS	72.2	54.6	61.3	94.5†	65.7	95.9†	57.9	38.1	47.4	92.1	37.3	**95.1**†
BART	FT	74.5	61.7	66.4	94.6	69.9	91.6	63.8	45.5	53.0	92.6	**45.0**	85.6
PEGASUS	FT	75.2	62.5	67.0	94.7	70.0	96.1	63.7	46.2	**53.2**	92.5	43.7	93.9
T5	FT	**76.6**	**64.4**	**68.9**	**95.0**	**74.2**	**98.2**	**64.1**	**46.4**	52.8	92.6	44.7	92.5

Table 2: **Report** and **Cross-Document** summarization results on SEAMuS. Best overall results are **bolded**; * and † denote best zero- and few-shot results, respectively. **S**=setting; RB=report baseline; ZS=zero-shot; FS=few-shot; FT=fine-tuned. See §4.1 for an explanation of metrics; higher is better for all. See Table 7 for 95% CIs.

the best overall numbers ($\mathbf{R}_{1,2,L}$, **CR**) or nearly so (**BS**), although GPT-4o and Claude Sonnet obtain highest scores on **BS** and **A**, respectively. Once again, nearly all models outperform the report baseline across the board (ZS GPT-4o Mini excepted). Finally, we note that results on most metrics are much lower in absolute terms compared to the corresponding results from §4.2, testifying to the greater difficulty of the cross-document task.

4.4 Impact of Extraction Quality

Setup The preceding experiments conditioned summary generation on gold event annotations in order to focus model comparisons narrowly on summarization ability. In most practical settings, however, a (CD)EKS model would instead have to rely on the imperfect outputs of an event extraction model. To probe robustness to extraction errors in a controlled manner, we apply variable amounts of noise to the gold event annotations and evaluate model performance on the resulting examples. Concretely, for each role R of each event, we edit R's arguments with probability p. If a role is selected for editing, we then make *one* of the following edits with equal probability:

1. Insert: A new span from the text is added to the argument list for R.
2. Delete: An argument span is removed at random from the argument list.
3. Replace: A random argument in the list is replaced with a different span from the text.

For the cross-document task, we apply these edits to the event annotations for both the report and the source. We sample the edits to be made and then prompt an LLM (GPT-4o) to apply them by supplying in a prompt: (1) the text (report or source), (2) the (JSON-formatted) report or source event annotations, and (3) instructions for the edits to be made, generated automatically by populating templatic statements based on the edits sampled. The LLM is free to select an appropriate span to be used for the Insert and Delete operations. We consider $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, using the same sampled edits for all models for a given p.

We evaluate one fine-tuned model (T5) and one model each from the Claude (Claude Haiku) and GPT (GPT-4o Mini) families. For T5, we use the same checkpoint as presented in Table 2. For Claude Haiku and GPT-4o Mini, we use few-shot prompts similar to those used in the FS setting in Table 2, but with two key changes. First, we alter the task instructions to say that the event annotations for the target example *may contain errors*, and that the model must correct these errors when generating its summary by consulting the text(s). Second, we show the model how to do this by substituting noised versions of the event annotations in the few-shot examples while leaving their associated texts and summaries unchanged.

Results for both tasks are in Table 3. For all models, we observe (near-)monotonic drops in performance for most metrics as p increases. And while performance drops are sizable in some cases, they are arguably less radical than one might expect, given the destructiveness of the changes at $p = 0.5$, where roughly half of all roles contain

Model	p	Report						Cross-Document					
		$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**
T5	0.0	76.6	64.4	68.9	95.0	74.2	98.2	64.1	46.4	52.8	92.6	46.3	92.5
	0.1	75.6	62.8	67.8	93.9	71.4	97.6	62.8	45.3	51.8	91.5	47.2	92.0
	0.2	74.0	61.7	66.2	93.6	69.6	98.0	62.0	44.3	50.7	91.4	43.5	89.3
	0.3	72.1	60.0	64.7	93.3	67.5	98.2	60.0	42.8	49.3	91.0	43.3	87.3
	0.4	70.3	57.5	62.1	92.9	66.4	95.8	58.4	40.9	47.8	90.8	44.4	87.4
	0.5	68.3	55.2	60.6	92.6	63.2	96.3	56.6	39.1	46.3	90.4	43.1	87.3
CLAUDE H (FS)	0.0	71.7	55.9	61.0	94.3	63.2	94.8	57.7	36.9	45.7	92.1	36.2	91.0
	0.1	67.5	51.5	56.7	93.7	59.4	94.8	57.2	37.3	45.4	91.8	37.1	82.5
	0.2	65.6	48.8	55.1	93.5	55.1	94.7	56.2	37.0	45.1	91.7	37.8	79.4
	0.3	64.7	47.8	54.1	93.3	52.8	94.6	56.0	36.2	44.9	91.5	32.7	82.2
	0.4	64.1	47.2	54.1	93.3	52.2	95.0	54.5	34.3	43.1	91.3	31.4	85.0
	0.5	63.1	46.8	54.0	93.1	52.3	94.7	54.3	34.6	43.3	91.3	33.1	86.4
GPT-4O M (FS)	0.0	72.0	55.4	61.0	94.3	66.8	94.1	57.5	36.9	45.7	92.1	39.8	88.5
	0.1	69.2	52.8	59.5	94.0	64.0	94.5	58.8	38.2	46.2	92.1	42.2	74.5
	0.2	67.6	50.8	57.0	93.7	59.8	94.2	56.6	36.2	45.1	91.2	39.4	75.2
	0.3	66.9	50.1	57.0	93.7	59.3	94.9	56.4	36.2	44.5	91.8	37.8	77.2
	0.4	65.2	48.1	54.9	93.4	56.7	93.8	54.8	34.0	42.8	91.6	36.6	77.8
	0.5	65.1	47.5	54.8	93.4	55.2	95.4	54.2	33.2	42.6	91.4	34.4	80.9

Table 3: Performance of three models from Table 2 when the argument annotations for each role in the report event (**Report**) or additionally in the source event (**Cross-Document**) are corrupted with probability p (see §4.4).

extraction errors. This is especially evident in the results for Claude Haiku and GPT-4o Mini on the cross-document task, where (e.g.) $\mathbf{R}_{1,2,L}$ scores decrease by only about 3 points from $p = 0$ to $p = 0.5$ and **BS** by less than 1. Indeed, losses on **CR**—the most explicit measure of extraction ability—amount to only about 5 points for GPT-4o Mini and only 3 points for Claude Haiku.

p	Summary
0.0	**The gradual accumulation of partially decayed plant material in a bog functions as a carbon sink.**
0.1	**The gradual accumulation of decayed plant material in a bog functions as a carbon sink.**
0.2	**The gradual accumulation of decayed plant material in a bog acts as a carbon sink.**
0.3	**The gradual accumulation of decayed plant material in a bog functions as a carbon sink.**
0.4	**The gradual accumulation of decayed plant material, including peat, in bogs functions as a carbon sink.**
0.5	**The gradual accumulation of decayed plant material in a bog functions as a carbon sink.**

Table 4: Example outputs from GPT-4o Mini on the cross-document task as role annotations are corrupted with probability p. In many cases (as here), we find minimal degradation in quality from $p = 0$ to $p = 0.5$.

These findings are confirmed by manual inspection of model outputs (Table 4), where we often see minimal degradation in summary quality—suggesting an intriguing strength of this task relative to traditional event extraction: the ability to *counteract extraction errors post-hoc* by using imperfectly extracted event information to locate relevant passages in the input and then leveraging those passages to avoid analogous errors in the summary.

4.5 Input Ablations

Following Gantt et al. (2024), Appendix F considers ablations on the input for both tasks, in which we omit the annotated events (TEXT ONLY) or the texts (EVENT ONLY), and condition summary generation on the ablated inputs. We also present a novel third ablation that omits the annotated arguments only, leaving intact information about the target frame and roles (TEXT+SCHEMA). Consistent with Gantt et al.'s results, we find that both the text and the full event annotations are needed to obtain the best results (see Tables 5 and 6), indicating that SEAMUS is *not* reducible to standard summarization (TEXT ONLY), structure-to-text (EVENT ONLY), or even a (simpler) hybrid objective (TEXT+SCHEMA).

5 Human Evaluation

Setup Lastly, we conduct a human evaluation of the reference and model-generated summaries. We focus our evaluation on the combined task only, comparing the summaries generated by models presented in Table 2 (RB excluded). For the GPT and Claude models, we use the summaries generated in the few-shot setting only (FS), owing to their superior results above. We randomly sampled 30 test set examples and presented the model-generated summaries for these examples, along

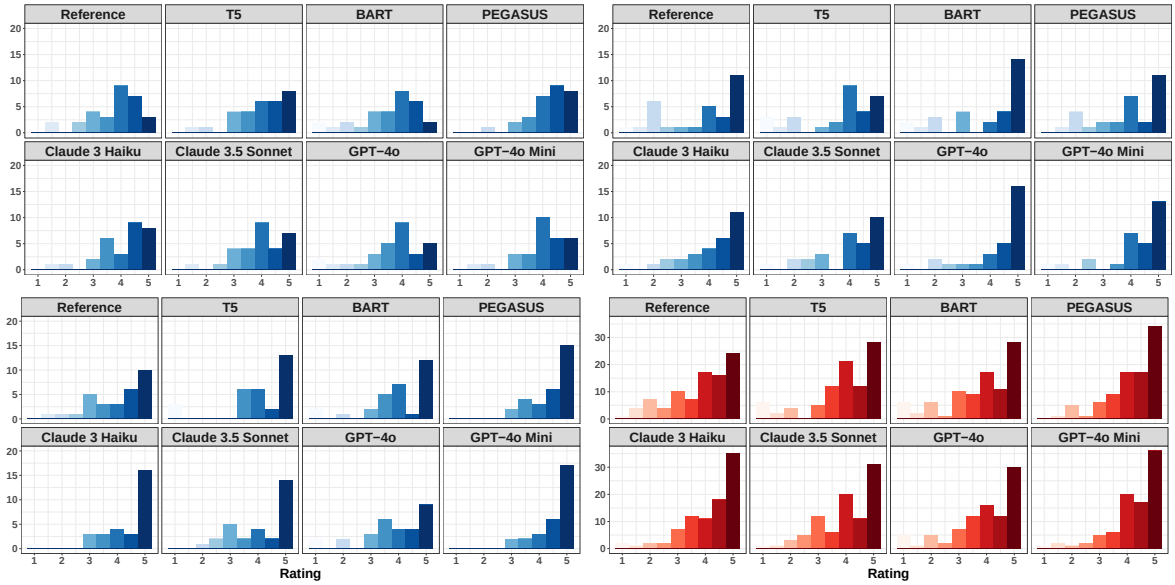

Figure 3: Histograms of summary quality scores (1-5, higher is better) from our human evaluation (§5). The bottom right plot (red) aggregates scores across all three raters; each of the other plots (blue) shows a single rater's scores.

with the references, to 3 human raters—all English-speaking NLP researchers who did not participate in other parts of this work. Each rater provided a single quality score for each summary based on the ranked list of attributes used by Gantt et al. (2024): *factuality*, *adequacy*, *coherence*, *relevancy*, and *fluency*. Scores were given from 1 (low) to 5 (high), with half points allowed. Each rater thus provided $30 \times (7+1) \times 1 = 240$ judgments. Raters were not shown which model produced which summary, and summary presentation order was randomized.[19]

Results Four sets of histograms of scores for each model (and the reference) are shown in Figure 3. The bottom right set (red) shows scores aggregated across annotators, while the other three (blue) each show scores of a single rater. For all raters, scores are consistently high across models and the reference, with modes of ≥ 4 for each. Comparing preferences across raters, however, we see significant variability: GPT-4o achieved the highest average score for one rater (top right, 4.28); GPT-4o Mini for the second (bottom left, 4.57); and PEGASUS for the third (top left, 4.23).

Looking at intra-rater distributions, however, it's unclear how robust these preferences are. Using Wilcoxon rank-sum tests to evaluate pairwise differences in each rater's scores for a given pair of models, we find that some of these preferences are reliable at $\alpha = 0.05$ (e.g. GPT-4o > T5 with $p = 0.016$ for the first rater), but none holds up when applying the Bonferroni correction to account for multiple comparisons. We take these results to indicate that our baselines are broadly effective at producing clear and accurate summaries, and that while they may somewhat differentiate themselves on individual metrics, the best summaries on a more holistic picture may come down to user preference. This plurality of solid modeling options is encouraging, and suggests flexibility in the application of CDEKS to a range of use cases.

[19] See Appendix D for further details.

6 Conclusion

This work has extended the task of *event-keyed summarization* (EKS) to the cross-document setting (CDEKS). To enable this, we provided an expert reannotation of the FAMUS CDAE dataset, yielding high-quality event argument annotations on all 1,253 examples. We then leveraged these improved annotations to construct SEAMUS—a collection of single- (*report*) and cross-document summaries on top of FAMUS, further annotating the summaries themselves for event arguments (§3). We benchmarked SEAMUS on a diverse set of baselines, including smaller fine-tuned models, as well as zero- and few-shot prompted LLMs (§4.2, §4.3). We then presented more detailed analysis, assessing the impact of degraded event extraction on summary quality (§4.4), and conducting a set of ablations on the input (§4.5)—concluding with a human evaluation of summary quality (§5). We release SEAMUS, along with our baseline results, to facilitate further work on EKS in both the single- and cross-document settings.

Limitations

One limitation of this work is the size of the resource. SEAMUS contains 1,253 examples, which, though sufficient for fine-tuning smaller models and for conducting prompting experiments with larger ones, is likely not sufficient for substantive fine-tuning of much larger models.

A second limitation is that the cross-document setting considers only two documents per example. This constraint was imposed by the choice of the FAMuS dataset as the basis for SEAMUS, as cross-document argument annotations in the former were provided only for *pairs* of report and source texts. Future work expanding the set of source texts would be valuable, and would allow both for richer summaries and for more robust evaluation of models' ability to accurately synthesize information across possibly differing accounts of events.

Addressing either limitation may require relaxing data quality goals—relying on either crowdsourcing or LLM-powered annotation techniques—as scaling our annotation procedure to substantially more examples or source documents would demand considerable resources.

Ethics

As the report and source texts in SEAMUS are the same as those in the FAMuS dataset, and as the summaries in SEAMUS are simply distillations of (parts of) these texts, we do not believe our dataset introduces any novel risks as a resource. Nonetheless, these texts do discuss real people, places, and institutions, and models trained on this data may thus be liable to make untrue claims about them or otherwise misrepresent them. We intend SEAMUS for academic use only, as a benchmark to evaluate systems for single- and cross-document event-keyed summarization.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Samuel Barham, Orion Weller, Michelle Yuan, Kenton Murray, Mahsa Yarmohammadi, Zhengping Jiang, Siddharth Vashishtha, Alexander Martin, Anqi Liu, Aaron Steven White, et al. 2023. Megawika: Millions of reports and their sources across 50 diverse languages. *arXiv preprint arXiv:2307.07049*.

Yanran Chen and Steffen Eger. 2023. MENLI: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Yunmo Chen, William Gantt, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023. A unified view of evaluation metrics for structured prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12868–12882, Singapore. Association for Computational Linguistics.

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International conference on machine learning*, pages 1223–1232. PMLR.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. 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.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

William Gantt, Alexander Martin, Pavlo Kuchmiichuk, and Aaron Steven White. 2024. Event-keyed summarization. *arXiv preprint arXiv:2402.06973*.

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.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Wessel Kraaij, Thomas Hain, Mike Lincoln, and Wilfried Post. 2005. The ami meeting corpus. In *Proc. International Conference on Methods and Techniques in Behavioral Research*, pages 1–4.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

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.

Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng. 2022. Multi-document summarization via deep learning techniques: A survey. *ACM Computing Surveys*, 55(5):1–37.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Guulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13666–13674.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from Abstract Meaning Representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25, Edinburgh, UK. 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.

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

Aafiya S Hussain, Talha Z Chafekar, Grishma Sharma, and Deepak H Sharma. 2022. Event oriented abstractive summarization. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 99–108, New Delhi, India. Association for Computational Linguistics.

Tal Schuster, Adam Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William Cohen, and Donald Metzler. 2024. SEMQA: Semi-extractive multi-source question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1363–1381, Mexico City, Mexico. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Sai Vallurupalli, Sayontan Ghosh, Katrin Erk, Niranjan Balasubramanian, and Francis Ferraro. 2022. POQue: Asking participant-specific outcome questions for a deeper understanding of complex events. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8674–8697, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Siddharth Vashishtha, Alexander Martin, William Gantt, Benjamin Van Durme, and Aaron White. 2024. FAMuS: Frames across multiple sources. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8250–8273, Mexico City, Mexico. Association for Computational Linguistics.

Jiarui Wang, Richong Zhang, Junfan Chen, Jaein Kim, and Yongyi Mao. 2022. Text style transferring via adversarial masking and styled filling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7654–7663, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large ontology multilingual extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 149–159, Online. Association for Computational Linguistics.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

A Additional Examples

Below, we show a few examples of the report summaries and their corresponding cross-document summaries to illustrate how the latter are able to provide greater detail about an event of interest relative to the former:

Example 1

- **Frame**: CAUSE TO RESUME
- **Report**: *Areva renewed a uranium deal with Niger in January 2008.*
- **Cross-Doc**: *On January 13, 2008, French state-controlled nuclear reactor maker Areva CEPFi said it had renewed a uranium mining deal with the state of Niger and would invest over 1 billion euros.*

Exmaple 2

- **Frame**: SMUGGLING
- **Report**: *A woman pled guilty to possession and attempting to smuggle 89 grams of heroin out of Thailand.*
- **Cross-Doc**: *Scot Sandra Gregory pled guilty to possession and attempting to smuggle 89 grams of heroin out of Thailand in 1993 and did her time in Thai jails.*

Exmaple 3

- **Frame**: HOSTILE ENCOUNTER
- **Report**: *The plot of Reign of Shadows involves players returning to the dark side of the moon of Luclin to face the snake-like Shissar race led by Emperor Ssraeshza.*
- **Cross-Doc**: *The plot of Reign of Shadows involves players returning to the heart of the dark side of the moon of Luclin to face the snake-like Emperor Ssraeshza and his unyielding throngs of insidious zealots and enslaved minions to take back the ancient citadel of Vex Thal and end their march.*

B Training and Evaluation

Models and Hardware The BART, T5, and PEGASUS models were all trained on a single NVIDIA A100 GPU using CUDA version 11.7. Results reported with these models are based on single runs with a fixed random seed. We fine-tune the following pretrained checkpoints available from HuggingFace:

- `t5-large`
- `facebook/bart-large`
- `google/pegasus-large`

Libraries Models were developed using Python 3.11.9. We used the following libraries for model training, inference, and evaluation:

- accelerate (0.34.2)
- bert-score (0.3.13)
- bm25s (0.2.1)
- datasets (3.0.1)
- deepspeed (0.15.1)
- editdistance (0.8.1)
- evaluate (0.4.3)
- metametric (0.1.2)
- numpy (1.26.4)
- rouge-score (0.1.2)
- sentence-transformers (3.1.1)
- spacy (3.7.5)
- torch (2.0.1+cu117)
- transformers (4.45.1)
- tokenizers (0.20.0)

Metrics We use the implementations of ROUGE ($\mathbf{R}_{1,2,L}$) and BERTScore (**BS**) provided by the HuggingFace `evaluate` library. We implement CEAF-REE (**CR**) and its soft-match variant (see Tables 5, 6) using the `metametric` package (Chen et al., 2023). Finally, we use the implementation of AlignScore released by the metric's authors (Zha et al., 2023).[20]

Hyperparameters BART, T5, and PEGASUS were all trained for a maximum of 30 epochs with a patience of 5 epochs, using ROUGE-1 ($\mathbf{R}_1$) F_1 score on the dev set as the evaluation criterion. We use the Adam optimizer (Kingma and Ba, 2014) with default hyperparameters ($\beta_1 = 0.9, \beta_2 = 0.99o$, $\epsilon = 1e^{-8}$, $\eta = 0.001$) for all models. For inference, we use beam search decoding with a beam size of 5 and set the maximum tokens to 256.

Input Formats Below, we show in greater detail the input format for BART, PEGASUS, and T5 for the report and cross-document results reported in Table 2 and Table 3. (Note: these input formats were also used to obtain the BART, PEGASUS, and T5 results in the TEXT+EVENT rows in Table 5 and Table 6.) Here, ⟨B⟩ and ⟨E⟩ denote the model's

[20] `https://github.com/yuh-zha/AlignScore`

start-of-sequence and end-of-sequence tokens, respectively (if applicable), and ⟨S⟩ denotes a special token used to delineate information pertaining to a particular event role. Other text set between angle brackets (⟨...⟩) denotes a variable placeholder. We add spaces between separators and adjacent text to improve readability below; they are not present in the actual input.

The input format for the **report** task is as follows:

⟨B⟩ Report: ⟨`Report Text`⟩ ⟨E⟩ ⟨B⟩ Frame ⟨S⟩ ⟨`Frame Name`⟩ ⟨S⟩ Trigger ⟨S⟩ ⟨`Trigger`⟩ ⟨S⟩ ⟨`Role 1 Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ ⟨`Role N Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ ⟨E⟩

The input format for the **combined** task is as follows:

⟨B⟩ Report: ⟨`Report Text`⟩ ⟨S⟩ Source: ⟨`Source Text`⟩ ⟨E⟩ ⟨B⟩ Report Event: Frame ⟨S⟩ ⟨`Frame Name`⟩ ⟨S⟩ Trigger ⟨S⟩ ⟨`Trigger`⟩ ⟨S⟩ ⟨`Role 1 Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ ⟨`Role N Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ Source Event: Frame ⟨S⟩ ⟨`Frame Name`⟩ ⟨S⟩ ⟨`Role 1 Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ ⟨`Role N Name`⟩ ⟨S⟩ ⟨`Arg 1`⟩; ⟨`Arg 2`⟩; ...⟨S⟩ ⟨E⟩

The ablation settings presented in Table 5 and Table 6 (TEXT ONLY, EVENT ONLY, TEXT+SCHEMA) do not fundamentally change this overall structure, but merely omit parts of it.

C LLMs

GPT All GPT models were accessed through the OpenAI Chat API[21], via the OpenAI Python SDK (`openai` 1.50.2). As noted in §4, we set temperature to 0.7 and set the maximum output tokens to 256 (consistent with the fine-tuned models) for all experiments reported in this paper and leave the other API defaults unchanged ($n = 1$, top_p is not set, and we use no frequency penalty, presence penalty, or logit bias). For GPT-4o, we used model version `gpt-4o-2024-08-06`. For GPT-4o Mini, we used model version `gpt-4o-mini-2024-07-18`. Results reported throughout the paper are based on a single generation per prompt.

Claude All Claude models were accessed through the Anthropic Messages API[22] via the Anthropic Python SDK (`anthropic` 0.34.2). As with the GPT models, we set temperature to 0.7 for all experiments in this paper and leave the other defaults unchanged (we do not set top_p or top_k, as recommended, and we do not set any stop sequences). For Claude 3.5 Sonnet, we used model version `claude-3-5-sonnet-20240620`. For Claude 3 Haiku, we used model version `claude-3-haiku-20240307`. Results reported throughout the paper are based on a single generation per prompt.

Prompts We use the same prompts for all LLMs. Complete prompts will be available in the public GitHub repository for this work. Here, we provide prompt templates used to obtain the results in Table 2 and Table 3, for both tasks (report or cross-document) and for both the zero- and few-shot settings. Text set between angle brackets ⟨...⟩ denote placeholders.

We use the same system prompt for both tasks:

> You are an expert intelligence briefer. Your task is to analyze a specific, important event based ONLY on certain information, and to compile a concise summary of that event to be presented to a high-ranked decision maker.

For the **report** task in the **zero-shot (ZS)** setting, the user prompt has the following structure:

> The Report text below describes a situation. The Report Template provides specific details about the same situation. Focus ONLY on information relevant to the Situation Type.
>
> Please write a short, accurate summary that is one sentence long and that is based ONLY on the provided information. DO NOT include any extraneous details. DO NOT use more than one sentence.
>
> Situation Type: ⟨`Frame Name`⟩ (⟨`Frame Def`⟩)
>
> Report: ⟨`Report Text`⟩
>
> Report Template:

[21] https://platform.openai.com/docs/api-reference/chat

[22] https://docs.anthropic.com/en/api/messages

- ⟨Role 1⟩ (⟨Role 1 Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

- ...

- ⟨Role N⟩ (⟨Role N Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

Summary:

The **few-shot (FS)** user prompt for the **report** task had the following structure:

The Report text below describes a situation. The Report Template provides specific details about the same situation. Focus ONLY on information relevant to the Situation Type.

Please write a short, accurate summary that is one sentence long and that is based ONLY on the provided information. DO NOT include any extraneous details. DO NOT use more than one sentence.

Here are a few examples to show you how to complete the task:

Example 1

————-

Situation Type: ⟨Frame Name⟩ (⟨Frame Def⟩)

Report: ⟨Report Text⟩

Report Template:

- ⟨Role 1⟩ (⟨Role 1 Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

- ...

- ⟨Role N⟩ (⟨Role N Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

Summary: ⟨summary text⟩

Example 2

————-

⟨ same format as above ⟩

Example 3

————-

⟨ same format as above ⟩

Now here is the target example for you to complete:

Target

———

⟨same format, but with summary text omitted⟩

The **zero-shot** user prompt for the **cross-document** task had the following structure:

The Report text below describes a situation, and the Report Template provides specific details about the same situation. The Source text provides additional context about this situation, and the Source Template provides additional details. Focus ONLY on information relevant to the Situation Type.

Please write a short, accurate summary that is preferably one sentence long (and no more than two sentences long) based ONLY on the provided information. DO NOT include any extraneous details. TRY to use one sentence and DO NOT use more than two.

Situation Type: ⟨Frame Name⟩ (⟨Frame Def⟩)

Report: ⟨Report Text⟩

Report Template:

- ⟨Role 1⟩ (⟨Role 1 Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

- ...

- ⟨Role N⟩ (⟨Role N Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

Situation Type: ⟨Frame Name⟩ (⟨Frame Def⟩)

Source: ⟨Source Text⟩

Source Template:

- ⟨Role 1⟩ (⟨Role 1 Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

- ...

- ⟨Role N⟩ (⟨Role N Def⟩): ⟨Arg 1⟩; ⟨Arg 2⟩; ...

Summary:

The **few-shot** user prompt for the **cross-document** task (not explicitly shown) follows exactly the same structure as the few-shot prompt for the report task, but naturally uses the cross-document example format in lieu of the report format.

D Human Evaluation

Full instructions for the human evaluation, along with a JSON file containing the items that were rated, are provided in the supplementary materials for this submission.

E Data & Annotation

E.1 License

SEAMUS will be released under an MIT License. As we note in the **Ethics** section, we intend SEAMUS for academic/research use only.

E.2 Annotation Interface

Here, we include screenshots of the annotation interface used to complete the Phase 2 annotation.[23] As noted in §3, the interface was adapted from Vashishtha et al.'s (2024) annotation interface for the FAMUS cross-document argument extraction task (cf. Figures 5 and 6 in Appendix A of their paper). Tasks were run via a locally hosted webapp.

In the first part of the Phase 2 annotation, the existing (crowdsourced) FAMUS argument annotations for the source text were reviewed and corrected, and the cross-document summaries were written jointly on the basis of these corrected annotations and the corrected report text argument annotations from Phase 1 (see Figure 4). The interface was pre-populated with (a) the corrected report text arguments from Phase 1 (in the "Report Text" tab, highlighted); the report summary from Phase 1 (in the "Report Summary" field); and (c) the uncorrected source text arguments (in the "Source Text" tab). The source text arguments were reviewed and corrected by toggling to the "Source Text" tab and making any necessary edits to the existing selections. The cross-document summaries were then written in the "Combined Summary" field. The UI for selecting, adding, and removing arguments was unchanged relative to Vashishtha et al.'s implementation. The major differences here are the addition of the "Report Summary" and "Combined Summary" fields, and the inability to alter the selected FrameNet frame for annotation.

In the second part, arguments were annotated on the summaries written in the first part (Figure 5). The interface is similar to the interface for the first part of the Phase 2 annotation, except that the "Report Summary" and "Combined Summary" fields have been removed, and a new tab ("Summary Text") containing the cross-document summary to be annotated was added. Summary arguments were annotated by toggling to this tab and making argument selections in the same way as before. Here,

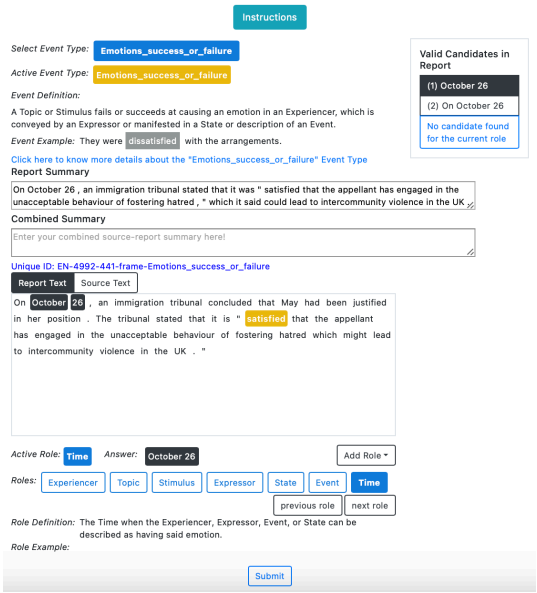

Figure 4: Interface for source text argument correction and cross-document summary writing (the first part of the Phase 2 annotation).

the corrected argument annotations for *both* the report text *and* for the source text were pre-populated for each task under their respective tabs, allowing annotators to toggle between these for reference in annotating the summary arguments.

As can be seen in both Figure 4 and Figure 5, details about the frame for the target event, including the frame name, its definition, as well as role names and their definitions, were provided as in the original FAMUS interface. Instructions were also accessible at any time via the dropdown shown at the top of the screen.

E.3 Annotation Instructions

Annotation instructions for both phases are in the supplementary materials for this submission.

E.4 Annotator Demographics

The full set of annotators consists of six students (five graduate and one undergraduate) pursuing degrees in Computer Science (3), Linguistics (2), and Cognitive Science (1), all of whom are fluent English speakers. Only one was financially compensated for the annotations (at a rate of $15 per hour), as this person initially became involved with the project through a university job board posting

[23] Recall that the Phase 1 annotation, which involved correcting the FAMUS report text argument annotations and writing the report summaries, was done in JSON files.

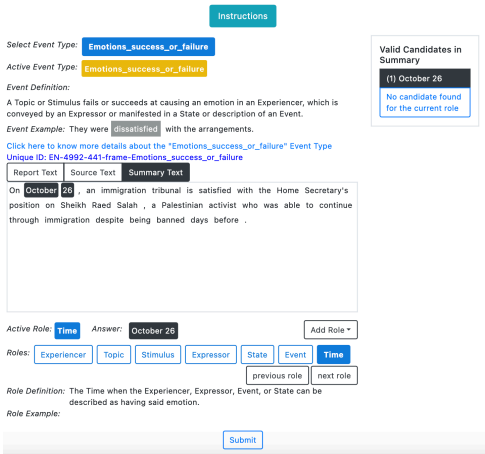

Figure 5: Interface for annotation of arguments on the cross-document summaries (the second part of the Phase 2 annotation).

for the task, whereas the others were members of the lab from which the project originated. The project, and the intended use of their annotations, was clearly explained to all participants in meetings before they began any annotation.

F Additional Results

F.1 Main Results

Table 7 contains test set scores and their 95% confidence intervals based on a non-parametric bootstrap ($n = 1,000$) of the results used to produce Table 2.

F.2 Input Ablations

Here, we include the full results of the ablations on the inputs introduced briefly in §4.5, which were inspired by similar ones conducted by Gantt et al. (2024). In the TEXT ONLY setting, we omit information about the target event entirely and include only the text in the input—either the report for the report task, or both the report and source for the cross-document task—effectively reducing the problem to standard summarization. In the EVENT ONLY setting, we omit the text(s) and include only information about the target event—either the report event annotations for the report task, or both the report and source event annotations for the cross-document task—making this ablation similar to structure-to-text tasks, such as AMR-to-text (Pourdamghani et al., 2016)). In the TEXT+SCHEMA setting, we omit the argument annotations, but leave in information about the frame and its roles. For the fine-tuned models, we include just the names of the frame and its roles. For the LLMs, we additionally include the definitions of the frame and roles as given in FrameNet. Finally, TEXT+EVENT is the name we assign to the *unablated* setting, used to obtain the results in Table 2 and Table 3, where both the text(s) and the full event annotations are present in the input. For all ablation settings, BART, PEGASUS, and T5 are fine-tuned on the ablated inputs using the same settings for training and inference as are described in §4. For the GPT and Claude models, the examples provided in the few-shot setting are also ablated in the way called for by each ablation.

Report Results for the report task are in Table 5. Here and in the cross-document results to follow (Table 6), we include a variant of CEAF-REE (**CR**) that we dub $\mathbf{CR}_{\text{soft}}$, which aligns and scores predicted arguments against reference arguments using normalized levenshtein distance rather than exact match—enabling a more nuanced comparison of different models' ability to recover event arguments in the summaries they produce.

Across all models and most metrics, we see significant drops in performance when ablating any component of the input. Notably, a number of models, especially the LLMs, fall to numbers near or below those of the report baseline (RB) on a variety of metrics.

There are, however, some unsurprising exceptions here. First, in many cases, results on **CR** and $\mathbf{CR}_{\text{soft}}$ in the EVENT ONLY ablation are markedly stronger than the report baseline, and are even competitive with the results in the unablated setting (TEXT+EVENT) for most of the zero-shot-evaluated LLMs. This echoes a similar finding by Gantt et al. (2024), who note that "the document [is not] needed to generate *some* string that contains all the [event] template's arguments." If this is correct, we would *expect* to see strong **CR** scores in the EVENT ONLY setting, even though the summaries may be poorer overall (as reflected in other metrics).

An intriguing, related observation is that whereas the fine-tuned models look dominant against the LLMs on **CR** in the unablated setting, this advantage sharply diminishes when we turn to $\mathbf{CR}_{\text{soft}}$. This is likely explained by the fact that the fine-tuned models are able to learn the conventions adopted by annotators in selecting argument spans, whereas the (prompted) LLMs do not—even though they may still be generating outputs with approximately correct spans that are nonetheless

harshly penalized by an exact match.

A second exception is the results on AlignScore (**A**) in the TEXT ONLY setting, which are competitive with—and in some cases superior to—the results in the unablated setting across models. Recall that **A** is a learned metric that (in this case) evaluates how well the report summary is supported by the report text. It is thus intuitively possible, and evidently quite feasible, to generate a summary that is adequately supported by the text without relying at all on the event annotations—which is exactly what is demanded by the TEXT ONLY setting. This is once again consistent with findings from Gantt et al. (2024) on the NLI-based family of metrics MENLI (Chen and Eger, 2023), which are broadly similar to AlignScore: "[event] templates are not needed to generate *some* summary that is entailed by the document."

We also note that, for the fine-tuned models, we obtain **A** scores in the TEXT+SCHEMA ablation that are comparable (T5) or higher than (BART, PEGASUS) those of the unablated setting. This makes sense, inasmuch as the TEXT+SCHEMA setting contains a superset of the inputs of the TEXT ONLY setting, though it is unclear why we do not find a similar pattern with the LLMs.

Finally, note that the report baseline, which treats the report text itself as the summary, should in theory achieve a perfect **A** score. The fact that it does not ($\mathbf{A} = 99.11$) likely merely reflects that this is a *learned* metric with modest flaws.

Cross-Document results on the cross-document task are shown in Table 6 and follow a pattern that is qualitatively very similar to that of the report results above. We consistently find that the best results are obtained in the unablated setting (TEXT+EVENT) for most metrics, with the same exception regarding $\mathbf{CR}/\mathbf{CR}_{\text{soft}}$ in the EVENT ONLY setting as we found for the report task. Curiously, however, the findings on **A** are more complicated here: whereas we continue to see the strongest results on this metric in the TEXT ONLY and TEXT+SCHEMA ablations for the fine-tuned models, with the LLMs, we instead see our best results in the unablated setting—following the trend of other metrics.

G Use of AI Assistants

GitHub Copilot was used as a coding assistant for parts of model development and data analysis, though its suggestions were carefully reviewed by the authors. AI assistants were **not** used for any other part of this work (writing, brainstorming, etc.).

Model	Ablation	Setting	R_1	R_2	R_L	BS	CR	CR_{soft}	A
Report Baseline	-	-	56.15	46.05	48.37	91.57	52.58	62.56	99.11
GPT-4o M	TEXT ONLY	ZS	49.96	28.18	39.23	91.31	34.59	53.13	95.74*
	EVENT ONLY	ZS	53.11	34.04	43.67	91.51	52.13	77.37	60.98
	TEXT+SCHEMA	ZS	53.29	31.60	42.91	91.28	38.24	56.92	79.07
	TEXT+EVENT	ZS	62.18	42.32	51.26	93.17	58.48	78.71	86.04
	TEXT+EVENT	FS	71.98	55.35	61.03	94.34	66.80	83.66	94.06
GPT-4o	TEXT ONLY	ZS	51.52	29.90	40.90	91.50	33.75	52.06	94.49
	EVENT ONLY	ZS	56.39	38.34	46.34	91.93	59.35	83.35	70.66
	TEXT+SCHEMA	ZS	56.57	37.19	47.08	92.00	42.37	61.50	81.66
	TEXT+EVENT	ZS	63.95	45.21	52.95	93.18	61.39*	82.60*	83.87
	TEXT+EVENT	FS	72.54†	56.59†	62.34†	94.40	69.61†	**87.27**†	94.72
CLAUDE H	TEXT ONLY	ZS	50.41	30.39	40.53	91.11	32.35	51.46	93.10
	EVENT ONLY	ZS	55.03	36.37	45.71	91.79	54.36	78.25	72.15
	TEXT+SCHEMA	ZS	57.67	38.51	47.68	92.08	41.36	59.10	83.24
	TEXT+EVENT	ZS	64.75	46.19	54.67	93.44	58.75	78.92	84.87
	TEXT+EVENT	FS	71.73	55.86	61.05	94.29	63.21	80.95	94.82
CLAUDE S	TEXT ONLY	ZS	46.98	22.83	36.24	90.78	25.68	45.88	91.31
	EVENT ONLY	ZS	55.66	36.89	46.21	92.13	56.38	78.54	72.15
	TEXT+SCHEMA	ZS	57.33	36.18	46.98	92.30	41.71	61.46	88.93
	TEXT+EVENT	ZS	67.38*	48.11*	56.52*	93.84*	61.07	81.35	92.96
	TEXT+EVENT	FS	72.16	54.64	61.29	94.54†	65.66	83.68	95.89†
BART	TEXT ONLY	FT	57.13	43.53	50.46	91.77	46.27	58.59	97.42
	EVENT ONLY	FT	58.34	40.96	48.51	91.83	59.82	75.34	51.17
	TEXT+SCHEMA	FT	62.23	49.43	55.55	92.59	52.92	65.83	95.01
	TEXT+EVENT	FT	74.46	61.68	66.42	94.57	69.88	82.72	91.59
PEGASUS	TEXT ONLY	FT	60.33	46.19	52.44	92.13	45.95	60.40	97.45
	EVENT ONLY	FT	59.69	41.97	49.46	91.90	57.14	74.34	53.93
	TEXT+SCHEMA	FT	63.28	49.79	55.91	92.71	53.69	66.28	96.94
	TEXT+EVENT	FT	75.18	62.53	66.96	94.70	70.00	82.68	96.08
T5	TEXT ONLY	FT	58.38	45.25	51.81	91.96	49.70	60.75	**98.88**
	EVENT ONLY	FT	63.14	45.62	52.47	92.67	64.00	80.08	68.42
	TEXT+SCHEMA	FT	65.82	51.90	58.46	93.11	56.18	68.42	97.92
	TEXT+EVENT	FT	**76.64**	**64.44**	**68.90**	**95.02**	**74.20**	85.22	98.15

Table 5: Input ablation results for the **report** summarization task. See §3.1 for an explanation of the settings.

Model	Ablation	Setting	R_1	R_2	R_L	BS	CR	CR_{soft}	A
Report Baseline	-	-	48.52	33.28	39.31	89.58	31.00	42.04	99.29
GPT-4o M	TEXT ONLY	ZS	37.56	16.93	26.97	88.98	21.86	40.48	73.58
	EVENT ONLY	ZS	52.45	31.15	40.04	91.17	37.48	66.51	69.97
	TEXT+SCHEMA	ZS	41.88	20.40	30.32	89.72	24.04	44.76	76.64
	TEXT+EVENT	ZS	51.87	29.90	39.10	91.31	38.99	64.13	81.46
	TEXT+EVENT	FS	57.48	36.99	45.74	92.08	39.78	62.93	88.48
GPT-4o	TEXT ONLY	ZS	41.59	19.28	30.70	89.48	21.60	42.04	69.09
	EVENT ONLY	ZS	54.03	33.98	42.13	91.51	41.75*	**69.63***	81.02
	TEXT+SCHEMA	ZS	49.87	27.04	37.76	90.86	25.80	48.53	85.44
	TEXT+EVENT	ZS	57.97	36.42	45.89	92.22*	41.34	68.04	86.61
	TEXT+EVENT	FS	61.17†	40.62†	49.38†	**92.67**†	42.72†	69.27†	90.62
CLAUDE H	TEXT ONLY	ZS	47.27	25.48	36.49	90.23	22.64	43.20	84.29
	EVENT ONLY	ZS	53.35	33.01	42.94	91.39	38.64	66.08	77.70
	TEXT+SCHEMA	ZS	51.79	30.45	41.04	90.87	26.38	48.02	87.10
	TEXT+EVENT	ZS	57.72*	36.88*	46.35*	92.05	36.22	60.03	90.37
	TEXT+EVENT	FS	59.42	39.40	48.56	92.13	37.20	59.70	90.99
CLAUDE S	TEXT ONLY	ZS	44.13	20.08	32.73	89.88	19.93	40.24	87.26
	EVENT ONLY	ZS	53.51	33.51	42.73	91.53	39.78	66.17	84.12
	TEXT+SCHEMA	ZS	51.37	29.33	40.06	90.94	28.05	49.07	88.64
	TEXT+EVENT	ZS	56.77	34.75	45.27	91.91	35.24	59.47	93.41*
	TEXT+EVENT	FS	57.95	38.05	47.53	92.09	37.32	59.31	95.09†
BART	TEXT ONLY	FT	48.57	30.30	39.70	89.99	27.12	44.43	90.06
	EVENT ONLY	FT	56.37	37.04	45.14	91.21	39.12	62.90	56.01
	TEXT+SCHEMA	FT	51.67	35.12	44.15	90.42	32.31	49.47	94.45
	TEXT+EVENT	FT	63.77	45.50	52.98	92.59	44.97	66.36	85.55
PEGASUS	TEXT ONLY	FT	50.85	33.44	42.51	90.29	30.22	47.46	97.63
	EVENT ONLY	FT	58.52	38.41	46.46	91.42	39.98	64.06	67.05
	TEXT+SCHEMA	FT	51.21	34.18	43.11	90.28	30.15	47.04	97.99
	TEXT+EVENT	FT	63.66	46.24	**53.18**	92.51	43.73	64.51	93.85
T5	TEXT ONLY	FT	49.18	33.15	41.39	89.94	30.98	46.58	**98.75**
	EVENT ONLY	FT	59.96	40.55	47.51	91.84	**45.30**	68.85	73.73
	TEXT+SCHEMA	FT	53.06	35.64	44.93	90.64	31.87	50.14	94.11
	TEXT+EVENT	FT	**64.14**	**46.36**	52.79	92.56	44.67	65.66	92.48

Table 6: Results on the **cross-document** summarization task. See §3.1 for an explanation of the settings.

		Report						Cross-Document					
Model	**S**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	**A**
GPT-4o M	ZS	$62.2^{64.5}_{60.0}$	$42.2^{45.2}_{39.3}$	$51.2^{54.0}_{48.7}$	$93.2^{93.6}_{92.8}$	$56.7^{60.5}_{52.8}$	$83.8^{86.9}_{80.5}$	$51.8^{53.7}_{49.9}$	$29.8^{32.0}_{27.8}$	$39.0^{41.0}_{37.0}$	$91.3^{91.6}_{91.0}$	$39.2^{42.2}_{35.9}$	$81.4^{84.1}_{78.5}$
	FS	$71.9^{74.0}_{69.8}$	$55.3^{58.1}_{52.5}$	$61.0^{63.5}_{58.5}$	$94.3^{94.7}_{94.0}$	$64.9^{68.8}_{60.7}$	$94.7^{95.8}_{93.4}$	$47.4^{59.7}_{55.2}$	$36.8^{39.5}_{34.3}$	$45.7^{47.8}_{43.4}$	$92.1^{92.4}_{91.7}$	$39.4^{42.9}_{35.6}$	$88.4^{90.6}_{86.0}$
GPT-4o	ZS	$63.9^{66.3}_{61.6}$	$45.1^{48.2}_{42.4}$	$52.9^{55.5}_{50.5}$	$93.2^{93.6}_{92.8}$	$58.4^{62.3}_{54.2}$	$86.0^{88.8}_{83.0}$	$57.9^{60.0}_{55.7}$	$36.2^{38.9}_{33.8}$	$45.7^{48.1}_{43.5}$	$92.2^{92.6}_{91.8}$	$39.9^{43.5}_{36.4}$	$86.5^{89.0}_{83.8}$
	FS	$72.5^{74.9}_{70.3}$	$56.6^{59.8}_{53.6}$	$62.3^{65.0}_{59.6}$	$94.4^{94.8}_{94.0}$	$66.8^{70.5}_{62.7}$	$94.0^{95.4}_{92.6}$	$61.1^{63.3}_{59.0}$	$40.6^{43.1}_{38.1}$	$49.3^{51.5}_{47.1}$	$92.7^{93.0}_{92.3}$	$41.4^{45.3}_{38.0}$	$90.6^{92.8}_{88.4}$
Claude H	ZS	$64.8^{67.2}_{62.7}$	$46.1^{49.1}_{43.5}$	$54.7^{57.3}_{52.4}$	$93.4^{93.9}_{93.1}$	$56.4^{60.4}_{52.5}$	$84.8^{87.7}_{81.5}$	$57.7^{59.7}_{55.6}$	$36.8^{39.2}_{34.3}$	$46.4^{48.7}_{44.0}$	$92.0^{92.4}_{91.7}$	$36.3^{39.9}_{32.8}$	$90.4^{92.3}_{88.1}$
	FS	$71.7^{73.9}_{69.4}$	$55.7^{58.7}_{52.7}$	$61.0^{63.6}_{58.5}$	$94.3^{94.7}_{93.9}$	$62.3^{66.5}_{58.1}$	$94.8^{96.1}_{93.3}$	$59.4^{61.5}_{57.0}$	$39.4^{42.1}_{36.7}$	$48.5^{50.9}_{46.0}$	$92.1^{92.5}_{91.8}$	$36.8^{40.4}_{33.4}$	$91.0^{92.9}_{89.0}$
Claude S	ZS	$67.3^{69.6}_{65.1}$	$48.0^{50.8}_{45.3}$	$56.5^{59.0}_{54.1}$	$93.8^{94.2}_{93.5}$	$59.2^{63.3}_{55.0}$	$92.9^{94.8}_{90.8}$	$56.7^{58.9}_{54.7}$	$34.7^{37.2}_{32.4}$	$45.3^{47.7}_{43.1}$	$91.9^{92.3}_{91.6}$	$34.5^{37.9}_{31.1}$	$93.4^{95.0}_{91.7}$
	FS	$72.0^{74.4}_{69.8}$	$54.5^{57.5}_{51.7}$	$61.3^{53.8}_{58.8}$	$94.5^{94.9}_{94.2}$	$64.6^{68.6}_{60.7}$	$95.9^{96.7}_{94.8}$	$57.9^{60.4}_{55.6}$	$38.0^{40.8}_{35.4}$	$47.5^{49.9}_{45.2}$	$92.1^{92.5}_{91.7}$	$37.6^{41.5}_{33.9}$	$95.1^{96.0}_{94.1}$
BART	FT	$74.3^{76.6}_{71.9}$	$61.7^{64.6}_{58.7}$	$66.4^{69.1}_{63.7}$	$93.7^{94.1}_{93.3}$	$68.2^{72.1}_{64.3}$	$91.6^{93.9}_{89.2}$	$63.8^{66.1}_{61.5}$	$45.5^{48.4}_{42.7}$	$53.0^{55.6}_{50.5}$	$91.9^{92.2}_{91.5}$	$45.1^{49.1}_{41.3}$	$85.6^{88.6}_{82.3}$
PEGASUS	FT	$75.1^{77.5}_{72.9}$	$62.4^{65.4}_{59.5}$	$66.8^{69.5}_{64.2}$	$93.7^{94.1}_{93.3}$	$69.0^{72.4}_{65.4}$	$96.1^{97.5}_{94.4}$	$63.6^{66.0}_{61.2}$	$46.0^{49.0}_{43.1}$	$53.1^{55.8}_{50.4}$	$91.7^{92.1}_{91.3}$	$44.7^{48.4}_{40.9}$	$93.8^{95.6}_{91.7}$
T5	FT	$76.6^{78.9}_{74.3}$	$64.3^{67.3}_{61.4}$	$68.8^{71.5}_{66.1}$	$94.0^{94.4}_{93.6}$	$73.3^{76.9}_{69.7}$	$98.2^{98.8}_{97.4}$	$64.0^{66.4}_{61.5}$	$46.2^{49.2}_{43.6}$	$52.8^{55.3}_{50.2}$	$91.8^{92.2}_{91.3}$	$44.5^{48.4}_{40.3}$	$92.4^{94.4}_{90.1}$

Table 7: Means and 95% confidence intervals ($\text{mean}^{\text{high}}_{\text{low}}$) from a non-parametric bootstrap ($n = 1000$) of the results reported in Table 2, which are based on single runs/prompts.