Event-Keyed Summarization

William Gantt[1] **Alexander Martin**[1] **Pavlo Kuchmiichuk**[2] **Aaron Steven White**[2]
[1]Johns Hopkins University [2]University of Rochester
{wwalden1, amart233}@jh.edu {pkuchmii@ur., aaron.white@}rochester.edu

Abstract

We introduce *event-keyed summarization* (EKS), a novel task that marries traditional summarization and document-level event extraction, with the goal of generating a contextualized summary for a *specific* event, given a document and an extracted event structure. We introduce a dataset for this task, MUCSUM, consisting of summaries of all events in the classic MUC-4 dataset, along with a set of baselines that comprises both pretrained LM standards in the summarization literature, as well as larger frontier models. We show that ablations that reduce EKS to traditional summarization or structure-to-text yield inferior summaries of target events and that MUCSUM is a robust benchmark for this task. Lastly, we conduct a human evaluation of both reference and model summaries, and provide some detailed analysis of the results.[1]

1 Introduction

Traditional event extraction (EE) aims to produce structured event representations from unstructured text. As early as the Message Understanding Conferences of the 1990s (Grishman and Sundheim, 1996), the motivation for EE was fundamentally human-centric: a desire for adaptive systems that could "respond to a user's information need" (Okurowski, 1993; Grishman, 2019). Yet, the majority of EE research focuses intensively on improving metrics on major benchmarks without due consideration for how (or whether) those improvements translate into better results for users.

Arguably, the most human-centric way to convey information about complex events is with summaries. Prominent summarization datasets such as CNN/Daily Mail (Nallapati et al., 2016), XSUM (Narayan et al., 2018), and GigaWord (Rush et al., 2015) rely on some conception of what is most salient to the *average* reader. While this is appropriate for many use cases, it is much less so when a user has a *specific* information need, as in EE.

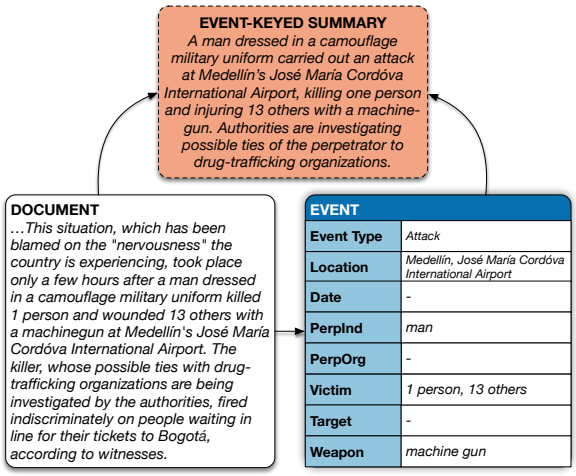

Figure 1: An illustration of the event-keyed summarization (EKS) task on a document and event template from the MUCSUM training split. Given a document and event template, a system must generate a contextualized summary of that *specific* event.

Research on controllable summarization tries to mediate this one-size-fits-all regime by giving users more command over particular summary attributes, such as length, style, or specificity (Fan et al., 2018; Liu et al., 2018; He et al., 2022; Zhang et al., 2023, *i.a.*), but very little work in this vein has explicitly focused on events. Notable exceptions include S Hussain et al. (2022), who use sets of extracted event keywords to encourage generated summaries to cover *all* events mentioned in a document, and Vallurupalli et al. (2022), whose POQue dataset includes both *process summaries*, which give high-level descriptions of a complex event, and *change summaries*, which give more granular descriptions of the changes undergone by a single participant.

Our work is related to these efforts, but differs critically in focusing *only* on types of events and roles a user has deemed relevant to their interests. We seek to marry EE's focus on event-centric infor-

[1]Code and data are available at https://github.com/wgantt/eks. This work was completed while the first and second authors were students at University of Rochester.

Findings of the Association for Computational Linguistics: EACL 2024, pages 7333–7345

mation needs with the readability of summarization to produce *event-keyed summaries* (EKS): short, targeted summaries of a *particular* event based on a document and an event structure extracted from it against a target ontology (Figure 1). In conditioning summary generation on an event representation, EKS also draws inspiration from prior work on query-driven summarization (Xu and Lapata, 2021; Shapira et al., 2022).

To study EKS, we introduce MUCSUM, a benchmark for this task based on the classic MUC-4 dataset (Sundheim, 1992). We further present fine-tuned and zero-shot baselines on MUCSUM, as well as ablations to show that our benchmark is *not* readily reducible to either traditional summarization or structure-to-text formulations. Finally, we conclude with a human evaluation of both human- and model-generated summary quality.

2 Task definition

We define an *event ontology* as a tuple $\langle \mathcal{E}, \mathcal{R}, \mathcal{S} \rangle$, consisting of a set of event types $\mathcal{E}$, a set of role types $\mathcal{R}$, and an assignment $\mathcal{S} : \mathcal{E} \rightarrow 2^{\mathcal{R}}$ of event types to sets of role types. We define an *event* as a pair $\langle E, R \rangle$, consisting of an event type paired with a (possibly empty) set of *event triggers* $E \in \mathcal{E} \times 2^{\Sigma^*}$ and a mapping from the roles associated with that event type to a (possibly empty) set of *role fillers* $R : \mathcal{S}(E) \rightarrow 2^{\Sigma^*}$.

We define *event-keyed summarization* (EKS) as the task of mapping an input document $D \in \Sigma^*$ and a query event $\langle E, R \rangle$ to a summary $S \in \Sigma^*$ (with $|S| \ll |D|$) that conveys all and only the *relevant* information in D about $\langle E, R \rangle$—with relevance determined by the role set $\mathcal{S}(E)$.

3 Data

MUC-4 We focus on the classic MUC-4 template extraction dataset as a case study (muc, 1992; Sundheim, 1992). In template extraction (contrasting with general event extraction), the set of event triggers is always empty, and so, in this case study, all information to be summarized comes from the event type and mapping from roles to role fillers.

MUC-4 annotates 1,700 documents concerning political conflict in Latin American countries, with terrorism-focused event types $\mathcal{E} = \{$`arson`, `attack`, `bombing`, `kidnapping`, `robbery`, `forced work stoppage`$\}$. All event types are associated with the same set of 24 roles—i.e. $\mathcal{S}[\mathcal{E}] = \mathcal{R}$, and each document may be associated with zero or more events of each type.

	Train	Dev	Test
Documents	1,300	200	200
Events (summaries)	1,114	191	-
Avg. words/doc	328.5	354.1	-
Avg. sents/doc	12.7	14.0	-
Avg. words/summary	44.1	51.1	-
Avg. sents/summary	1.7	1.8	-

Table 1: MUCSUM dataset statistics. Detailed test set statistics are deliberately omitted.

Since MUC-4, it has become standard to focus on a five-role subset, consisting of the individual perpetrator(s) (`PerpInd`), the organization(s) they are affiliated with (`PerpOrg`), the weapons they use (`Weapon`), victims of the incident (`Victim`), and damaged physical infrastructure (`Target`) (Chambers and Jurafsky, 2011; Du et al., 2021a; Chen et al., 2023; Gantt et al., 2023, 2024, *i.a.*). We follow this practice here, but with two additions. First, we include the `StageOfExecution` role, which conveys whether the event actually occurred, was (unsuccessfully) attempted, or was merely threatened. Second, we include the `Location` and `Date` roles in cases where this information can actually be extracted from the text.[2] These properties (time, location, and reality status) contain essential details about an event, and are necessary for a complete summary when provided.

MUCSUM Given the gold template annotations in MUC-4, we (the first three authors) wrote one *abstractive* summary per document-template pair $\langle D, \langle E, R \rangle \rangle$ that aims to convey all relevant information about $\langle E, R \rangle$ provided in D, given the roles listed above. To reduce the burden of writing summaries for so many events, we adopted a generate-then-edit approach: we first prompted ChatGPT[3] to produce a succinct ($\leq$ 3-sentence) candidate summary conditioned on $\langle D, \langle E, R \rangle \rangle$, then manually edited the result to ensure that it contained all $\mathcal{R}$-relevant information represented in D about $\langle E, R \rangle$. Additional information was included in the summaries if it provided important context or was otherwise necessary to ground the situation being described. In the course of writing summaries, we also re-annotated the `Date` and `Location`

[2] In MUC-4, arguments for `Location` and `Date` act more like document metadata, as in most cases their values are not actually extractable from the text itself. We (re-)annotate them only when they can be extracted.

[3] `https://openai.com/blog/chatgpt`.

Model	Setting	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	$S_r \rightarrow S_p$	$S_p \rightarrow S_r$	$S_r \leftrightarrow S_p$	$D \rightarrow S_p$
3-Sent Baseline	-	46.0	28.7	33.6	89.3	37.9	1.7	5.9	3.8	1.3
ChatGPT	temp+doc	47.0	30.4	35.6	88.6	60.2	30.6	**43.0**	36.8	40.8
GPT-4	temp+doc	48.7	30.0	35.9	88.9	67.6	**45.0**	40.3	**42.7**	38.8
BART	temp+doc	66.7	47.9	52.7	93.4	71.8	39.9	30.6	35.2	37.4
	temp only	51.9	30.5	37.9	91.2	74.6	15.4	8.8	12.1	12.2
	doc only	46.1	27.5	35.7	89.6	41.6	18.3	11.8	15.1	41.0
PEGASUS	temp+doc	63.9	44.9	50.4	93.0	67.6	36.8	28.5	32.6	40.6
	temp only	54.4	34.1	41.4	91.8	**75.7**	30.6	7.9	19.3	19.8
	doc only	47.0	28.2	36.2	89.8	41.2	18.8	13.0	15.9	42.2
T5	temp+doc	**67.0**	**48.6**	**53.4**	**93.5**	70.9	43.1	30.5	36.8	40.6
	temp only	54.4	33.6	40.6	91.7	75.1	27.4	7.6	17.5	16.5
	doc only	47.2	29.0	37.0	90.0	42.5	18.2	12.5	15.3	**42.4**

Table 2: **R**OUGE-$\{1, 2, \text{L}\}$, **B**ERT**S**core, **C**EAF-**R**EE F_1 scores, and NLI metrics (see §4) on the MUCSUM test set. ChatGPT and GPT-4 results are zero-shot and reflect averages across three prompts. BART, PEGASUS, and T5 are fine-tuned and reflect averages across three training runs. Ablation results are in gray.

roles (see footnote 2). Each $\langle D, \langle E, R \rangle \rangle$ pair was singly annotated, though we redundantly annotated a random subset of 30 test set examples and include agreement measures for these in Appendix B. We release the resulting dataset, **MUCSUM**, under an MIT License. Summary statistics are in Table 1.[4]

4 Evaluation

Apart from our human evaluation (§6), we report several standard summarization metrics, including ROUGE-$\{1, 2, \text{LCS}\}$ F_1 (Lin, 2004; Lin and Och, 2004) and BERTScore F_1 (Zhang et al., 2019). Since EKS summaries focus on event participants, we also report the CEAF-REE F_1 metric of Du et al. (2021a), a form of argument F_1 for string-fill roles. This provides a direct measure of how well a summary recovers the arguments from the input template. We train the span extractor of Xia et al. (2021) to extract and type arguments of the five entity-valued roles from the MUCSUM summaries. We then use the extractor to extract arguments from each summary and report CEAF-REE F_1 scores relative to the gold templates.

Finally, recent metrics based on Natural Language Inference (NLI) entailment ($\rightarrow$) probabilities have been shown to exhibit higher correlation with human summary quality judgments than prior metrics (Chen and Eger, 2023). Letting $\langle S_p, S_r \rangle$ denote a predicted-reference summary pair for document D, we report the following probabilities: $S_p \rightarrow S_r$, $S_r \rightarrow S_p$, $S_p \leftrightarrow S_r$[5], and $D \rightarrow S_p$.[6]

[4]Data and annotation instructions can be found on our GitHub repo. Details on the ChatGPT summarization prompt and hyperparameters are included in Appendix A.

[5]$S_p \leftrightarrow S_r = [(S_p \rightarrow S_r) + (S_r \rightarrow S_p)]/2$

[6]Entailment classification is 3-way, so probabilities $> \frac{1}{3}$ indicate predicted entailment. Further details in Appendix A.

5 Experiments

5.1 Fine-Tuning

Setup We fine-tune several standard pretrained language models (LMs) for EKS: BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020), and T5 (Raffel et al., 2020), using the large versions of each. We provide as input the document concatenated with a linearized representation of the template and its arguments. We train on the gold summaries for 30 epochs, using ROUGE-1 F_1 scores on the dev split to select the best checkpoint. For inference, we use beam search decoding with a beam size of 5 and constrain the summary length to be no longer than the length of the longest summary in the training data (256 tokens). For inputs that exceed the context window size, we right-truncate the document text. Additional details on models and inputs are provided in Appendix A.

Results Average test set metrics across three training runs for each model are given in the bottom three `temp+doc` rows of Table 2. T5 maintains a slight-to-moderate edge over the other two models on most metrics, with BART exhibiting particularly competitive (and, for **CR** and $S_p \rightarrow S_r$, superior) performance. Prior work has noted the tendency of PEGASUS to produce more extractive summaries (Ladhak et al., 2023), which may in part explain why it underperforms the other two on our abstractive benchmark. All three models tend to produce summaries that are entailed both by the reference ($S_r \rightarrow S_p$) and the document ($D \rightarrow S_p$).[7]

[7]We also calculated $D \rightarrow S_r = 0.410$ on the test set, suggesting that BART and PEGASUS are approaching human-level scores on $D \rightarrow S_p$.

5.2 Ablations

Setup One might wonder how much the summaries in MUCSUM actually synthesize information from *both* the document and query event. Most structure-to-text tasks, such as AMR-to-text (Pourdamghani et al., 2016; Flanigan et al., 2016, *i.a.*) and SQL-to-text (Koutrika et al., 2010; Iyer et al., 2016, *i.a.*) condition generation *only* on the relevant structured representation (the AMR graph or the SQL query). Conversely, traditional summarization conditions only on the input document. These two setups provide natural baselines against which to compare the results discussed so far, which condition on both the document and query event. As such, we consider an ablation of the fine-tuned models in which we provide as input either *only* the event template or *only* the document.

Results The results of these experiments are in the `temp only` and `doc only` rows of Table 2. Across most metrics, we observe degradations when ablating either the document or the template from the input, strongly indicating that MUCSUM summaries *do* generally leverage both the document and the event template. In most cases, this degradation is more severe when ablating the document, which makes intuitive sense, as the summaries are deliberately written to be targeted to the event represented by the *template*. The superior performance of `doc only` on $D \rightarrow S_p$ and of `temp only` on **CR** are intelligible when considering that templates are not needed to generate *some* summary that is entailed by the document, nor is the document needed to generate *some* string that contains all the template's arguments. Yet, both are necessary for a maximally informative, contextualized summary ($\mathbf{R}_{1,2,L}$, **BS**, $S_r \leftrightarrow S_p$).

5.3 Zero-Shot Prompting

Setup Finally, we present zero-shot prompted results using ChatGPT and GPT-4 (OpenAI, 2023). To avoid inflating scores, we use three *different* prompts from the one used to generate candidate summaries for MUCSUM annotation, and report average results across the three, using the same prompts for both models.[8]

Results Results are in the second and third rows of Table 2. While $\mathbf{R}_{1,2,L}$, **BS**, and **CR** scores trail

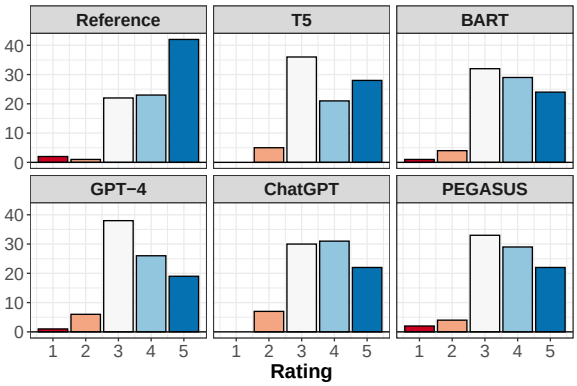

Figure 2: Distribution of ratings for models' summaries across 30 documents in quality evaluation.

those of the fine-tuned models, the NLI metrics are actually superior in a number of cases. This may indicate that these models are capable of generating decent summaries, but ones with much less lexical overlap with the references than the summaries the from fine-tuned models exhibit.

6 Analysis

Lastly, we consider how the summaries generated by the above models actually fare under human evaluation. We solicited summary quality judgments from three fluent English speakers on 30 randomly selected $\langle D, \langle E, R \rangle\rangle$ pairs from the test split. For each pair, annotators provided a single, five-point Likert-scale quality judgment for the summary generated for that pair by each of the five models in Table 2, plus the reference summary.[9] Annotators were given information about the event ontology and were asked to consider the following attributes (in order of importance) in making their judgments: *factuality*, *adequacy*, *coherence*, *relevancy*, and *fluency*. The source of each summary (whether a model or the reference) was not revealed to annotators, and summary presentation order was randomized across examples.[10]

Figure 2 shows the distribution of responses for each model's summaries. To compare these ratings, we conducted paired Wilcoxon rank sum tests for each pair of models, computing the difference between the rating an annotator gave for a particular model's summary for a particular $\langle D, \langle E, R \rangle\rangle$ against each other model's summary for that $\langle D, \langle E, R \rangle\rangle$.

We find that all models produce summaries that are reliably worse than the reference ($ps < 0.01$):

[8] A modest effort was invested in manually identifying effective prompts using several training examples, but we leave a thorough prompt engineering study for future work.

[9] We use the summaries associated with the checkpoint/prompt that obtained the highest dev ROUGE-1 score.

[10] See Appendix Appendix C for further details.

the best model outputs are rated 0.33 points worse than the reference on average (BART), with some models yielding an average difference of as much as 0.51 (GPT-4). The differences among models are generally much smaller: all are less than 0.2, with GPT-4 and PEGASUS tending to perform worse than BART and T5 (consistent with the results in §5), though no differences are reliable ($ps > 0.1$). Appendix C contains further discussion.

7 Conclusion

We have introduced the task of *event-keyed summarization* (EKS), in which the goal is to generate a summary of a *specific* target event described in a document, given an underlying event ontology. We have introduced a robust benchmark for EKS, MUCSUM, and presented a suite of fine-tuned and zero-shot baseline results across a diverse array of metrics. Our ablations reveal that MUCSUM effectively synthesizes targeted event information with its document context. Lastly, our human evaluation testifies to the quality of the reference summaries, while showing that our baselines also yield summaries of reasonable quality.

Limitations

Owing to its tightly focused ontology and its long and productive history in IE (Sundheim, 1992; Patwardhan and Riloff, 2009; Chambers and Jurafsky, 2011; Du et al., 2021a,b; Chen et al., 2023; Das et al., 2022; Gantt et al., 2023, *i.a.*), MUC-4 offers an excellent initial testbed for event-keyed summarization, and we chose it as the basis for our MUCSUM dataset for these reasons. However, the MUC-4 ontology is small and other document-level EE datasets with more diverse or sophisticated ones, such as FAMuS (Vashishtha et al., 2024) or MAVEN (Wang et al., 2024), may require more detailed summaries, and performance on EKS datasets derived from these resources may be lower than what we observe here.

Additionally, in the interest of controlling for extraction quality and focusing specifically on summarization performance, we generate summaries exclusively from the reference templates in this work. However, applications leveraging event-keyed summaries may rely on predicted templates, which could yield a degradation in summary quality. Experiments that consider the full extraction-to-generation pipeline would thus be an intriguing avenue for follow-up work.

Ethics

While the MUC-4 dataset has a long history in the NLP and IE communities, the documents it contains—and our MUCSUM summaries, by extension—do concern historical incidents of terrorism and use the names of real persons involved in them. As such, caution is clearly warranted in using this data in the training, development, or deployment of models for EKS or any other task. Given the fallibility of summarization models, it is possible, and even likely, that models trained on this data will make inaccurate statements concerning these historical incidents and others. We intend MUCSUM to be used for academic purposes only.

Acknowledgements

This work was supported by NSF-BCS (2040831). Alexander Martin was also supported in part by the River Campus Libraries and the Goergen Institute for Data Science at the University of Rochester. The authors thank Jin Dou, Pranay Mundra, and David Gantt for their assistance with the human evaluation, as well as members of the Human Language Technology Center of Excellence and participants at the PEER 2024 workshop for helpful feedback on this work.

References

1992. *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA. Association for Computational Linguistics.

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

Yunmo Chen, William Gantt, Weiwei Gu, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023. Iterative document-level information extraction via imita-

tion learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1858–1874, Dubrovnik, Croatia. Association for Computational Linguistics.

Aliva Das, Xinya Du, Barry Wang, Kejian Shi, Jiayuan Gu, Thomas Porter, and Claire Cardie. 2022. Automatic error analysis for document-level information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3960–3975, Dublin, Ireland. Association for Computational Linguistics.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Xinya Du, Alexander Rush, and Claire Cardie. 2021a. 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.

Xinya Du, Alexander Rush, and Claire Cardie. 2021b. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, 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.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

William Gantt, Shabnam Behzad, Hannah YoungEun An, Yunmo Chen, Aaron Steven White, Benjamin Van Durme, and Mahsa Yarmohammadi. 2024. Multimuc: Multilingual template filling on muc-4. *arXiv preprint arXiv:2401.16209*.

William Gantt, Reno Kriz, Yunmo Chen, Siddharth Vashishtha, and Aaron White. 2023. On event individuation for document-level information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12938–12958, Singapore. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166.

Ralph Grishman. 2019. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. CTRLsum: Towards generic controllable text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2073–2083, Berlin, Germany. Association for Computational Linguistics.

Georgia Koutrika, Alkis Simitsis, and Yannis E Ioannidis. 2010. Explaining structured queries in natural language. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 333–344. IEEE.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. 2023. When do pre-training biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3206–3219, Dubrovnik, Croatia. Association for Computational Linguistics.

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.

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.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium. Association for Computational Linguistics.

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.

Mary Ellen Okurowski. 1993. Information extraction overview. In *TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19-23, 1993*, pages 117–121, Fredericksburg, Virginia, USA. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160, Singapore. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21(1).

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

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.

Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Ido Dagan, and Yael Amsterdamer. 2022. Interactive query-assisted summarization via deep reinforcement learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2551–2568, Seattle, United States. Association for Computational Linguistics.

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.

Xiaozhi Wang, Hao Peng, Yong Guan, Kaisheng Zeng, Jianhui Chen, Lei Hou, Xu Han, Yankai Lin, Zhiyuan Liu, Ruobing Xie, Jie Zhou, and Juanzi Li. 2024. MAVEN-ARG: Completing the puzzle of all-in-one event understanding dataset with event argument annotation. In *Proceedings of the 62nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4072–4091, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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.

Yumo Xu and Mirella Lapata. 2021. Generating query focused summaries from query-free resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6096–6109, Online. Association for Computational Linguistics.

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

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

Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2023. MACSum: Controllable summarization with mixed attributes. *Transactions of the Association for Computational Linguistics*, 11:787–803.

A Model and Implementation Details

A.1 OpenAI Models

Candidate summaries were generated using the `gpt-3.5-turbo` model, accessed via the OpenAI Chat API on November 25, 2023.[11] Default Chat API hyperparameters were used, with the following exceptions: (1) temperature was set to 0.8; (2) the maximum number of new tokens (i.e. tokens in the summary) was set to 256. The OpenAI Chat API allows users to specify both *system prompts*, which provide high-level instructions about the task to be performed, as well as *user prompts*, which generally provide the data to be operated on. For summary creation, we supply the following as the system prompt for all examples:

> I will give you a document and a bulleted list of information about an event that the document describes. Using AT MOST 3 sentences, I want you to generate a short, accurate summary that includes ALL the information I provide you in the list. Additionally, please include information about the time and location of the attack if it is given in the document. You absolutely CANNOT include any other information that is not provided in the list. DO NOT include any extraneous details. DO NOT use more than 3 sentences.

For each example, the user prompt has the format shown below. Text between angle brackets (⟨`text`⟩) is a placeholder, populated with the relevant value for each target example. Text between square brackets ([`text`]) is included only if a template has a non-null value for that slot:

> Document: ⟨`document text`⟩
>
> Event Information:
>
> - Event Type: ⟨`event type`⟩
> - Stage of Execution: ⟨`StageOfExecution`⟩
> - [Individual Perpetrators: ⟨`perp1,...,perpN`⟩]
> - [Organizations Responsible: ⟨`org1,...,orgN`⟩]
> - [Weapons: ⟨`weapon1,...,weaponN`⟩]
> - [Victims: ⟨`victim1,...,victimN`⟩]
> - [Physical Targets: ⟨`target1,...,targetN`⟩]
>
> Summary:

The three user prompts used for the zero-shot results in §5 are the same as above, but we vary the system prompt, using three different paraphrases of the system prompt above. For ChatGPT, we again use the `gpt-3.5-turbo` model and for GPT-4, we use the `gpt4` model, both with the same API query parameters as listed above. These experiments were run on December 8, 2023.

[11] https://platform.openai.com/docs/guides/text-generation/chat-completions-api

A.2 Fine-Tuned Models

Training and inference of our fine-tuned models was done with the HuggingFace Transformers (v4.35.2) and Tokenizers (v0.15.0) Python (v3.10.13) libraries (Wolf et al., 2019). We use the `t5-large` (770M params), `facebook/bart-large` (406M params), and `google/pegasus-large` (568M params) pretrained checkpoints available on the HuggingFace Hub (v0.19.4). We did not perform any hyperparameter search on these models, relying on the reasonable defaults provided by the HuggingFace API. We additionally rely on the default inference ("generation") configuration for each model, with the exception of a uniform beam size (5) across all three models and the constraints on the minimum (15) and maximum (256) number of generated tokens. We used 1337, 1338, and 1339 as the random seeds for the training runs for each of the three models. We train each model on a single NVIDIA RTX 6000 GPU.

For the `temp+doc` setting, the input for each example consists of the document text, followed by a linearized representation of the template:

$$\langle \text{document} \rangle [\text{SEP}] \langle \text{template} \rangle$$

with BOS and EOS tokens inserted as required by the specific model. Drawing inspiration from Du et al. (2021a), the ⟨template⟩ representation uses a special role-delimiter token, [RSEP], chosen from each model's additional special token vocabulary, to delineate fillers for different roles, and also includes a description of the role:

$$[\text{RSEP}] \langle \text{role 1 description} \rangle : \langle \text{role 1 value(s)} \rangle$$
$$\ldots$$
$$[\text{RSEP}] \langle \text{role } N \text{ description} \rangle : \langle \text{role } N \text{ value(s)} \rangle$$

Where ⟨role i value(s)⟩ is a comma-separated list of role fillers and where the role descriptions are (in order), *event type*, *completion*, *date*, *location*, *individual perpetrators*, *organizations responsible*, *physical targets*, *victims*, *weapons*. For the entity-valued roles, a single mention is used to represent each entity. For the `doc only` ablation, only ⟨document⟩ is used. For the `temp only` ablation,

only ⟨template⟩ is used. When the input exceeds the context window length ($W = 1,024$ for all models), only the document text is truncated, and it is truncated right-to-left.

A.3 NLI Metrics

For our NLI metrics, we use the `menli` Python package released by Chen and Eger (2023).[12] We use the entailment probability (e) alone, following the authors' observation that it generally yields reasonable results (in lieu of the other formulas they consider that incorporate the neutral (n) and contradiction (c) probabilities), and we do not mix the NLI metrics with any others (i.e. we set `nli_weight`=1.0). We use NLI-R as the underlying model, which is a RoBERTa-large model (Liu et al., 2019) fine-tuned on several NLI datasets. The metrics we report in the main text cover both reference-based ($S_p \rightarrow S_r$, $S_r \rightarrow S_p$, $S_p \leftrightarrow S_r$) and reference-free ($D \rightarrow S_p$) settings. $S_p \leftrightarrow S_r$ is simply the mean of $S_p \rightarrow S_r$ and $S_r \rightarrow S_p$.

A.4 Other Metrics

We use the implementations of ROUGE-{1,2,LCS} and BERTScore provided by the HuggingFace Evaluate (v0.4.1) Python library.[13] For CEAF-REE (**CR**), we use a lightly adapted version of the implementation provided by Du et al. (2021b) that excludes the event type from the micro-average scores.[14] We train the span extraction system of Xia et al. (2021) on MUCSUM, using RoBERTa-large (Liu et al., 2019) as the encoder.[15] We report exact span match (matching span boundaries and matching slot type) and partial span match (matching span boundaries, ignoring slot type) metrics in Table 3. For both, we obtain F_1 scores in the low-to-mid 70s. While these are strong scores, they are not perfect and our **CR** results should be interpreted cautiously—as those for any model-based metric should be. As a final note, **CR** for the gold test set summaries is 78.2, which puts the best models on this metric in Table 2 (the `temp only` ablations) within several points of human-level performance.

A.5 Preprocessing

As the MUC-4 data does not have canonical sentence splits, we use the SpaCy (v3.7.2) sentence tokenizer to obtain sentence boundaries and their default word-level tokenizer for English to obtain the statistics used in Table 1.[16]

	P	R	F_1
Exact Span Match	72.5	75.2	73.8
Partial Span Match	73.8	76.6	75.2

Table 3: Exact and partial span match P/R/F_1 of our span extraction system on the MUCSUM test split.

B Annotation Agreement

Instructions for the MUCSUM summary annotation are included in the GitHub repository. As we note in §3, a single annotator wrote the summary for each $\langle D, \langle E, R \rangle\rangle$ pair. However, in the interest of providing some measure of inter-annotator agreement, all annotators annotated the same random sample of 30 documents from the test split. In Table 4, we report a subset of the metrics from Table 2 on these annotations—alternately treating the summaries of one annotator as the "reference" and those of the other two as "predictions."

Perhaps the most important observation is that, across metrics, there are numerous cases (i.e. metric-annotator pair combinations) in which one can find a superior result from one of the models in Table 2—though with the important caveat that these scores are not calculated on the same items. This offers some testament to the strength of our baselines (and of the fine-tuned models in particular). At the same time, it suggests that "human-level" summarization performance perhaps sits lower on the scales of these metrics than we may reflexively be inclined to think, and that numbers higher than these should not automatically be read as better. Across the NLI metrics, for instance, the highest entailment score we observe is 53.0 (between A1 and A3 on $S_p \rightarrow S_r$) which—though better than any NLI result in Table 2—is still far from 100. We thus echo the many calls from this literature to be wary of any individual summarization metric (Bhandari et al., 2020; Deutsch et al., 2021; Gehrmann et al., 2023, *i.a.*), but we do not think this warrants their dismissal (see Appendix C).

C Human Evaluation

The participants in our human evaluation study (§6) comprised three fluent English-speaking volunteers

[12] https://github.com/cyr19/MENLI

[13] https://huggingface.co/docs/evaluate

[14] https://github.com/xinyadu/gtt. The `Location` and `Date` roles are also excluded.

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

[16] https://spacy.io/

S_r	S_p	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	$S_r \to S_p$	$S_p \to S_r$	$S_r \leftrightarrow S_p$	$D \to S_r$
A1	A2	57.0	40.8	49.3	92.1	3.4	46.8	25.1	48.9
	A3	53.3	36.4	41.8	91.3	13.2	53.0	33.1	
A2	A1	57.0	40.8	49.3	92.1	46.8	3.4	25.1	50.2
	A3	71.1	53.1	59.0	94.0	36.4	49.7	43.1	
A3	A1	53.3	36.4	41.8	91.3	53.1	13.2	33.1	46.1
	A2	71.1	53.0	59.0	94.0	49.7	36.4	43.1	

Table 4: Agreement among the authors (annotators of MUCSUM) on 30 test set examples, as measured by the metrics reported in the main text. The annotator in the S_r column is treated as the "reference" and the annotator in the S_p column is treated as the "prediction" (but note that the distinction is moot for all metrics except $S_r \to S_p$ and $S_p \to S_r$). Also note that these are *not* the same annotators as in Table 5. See discussion in Appendix B.

(unpaid), two of whom are students at the authors' home institution and one of whom is a personal acquaintance of one of the authors. The task was designed to be doable in under two hours. Annotators were permitted to ask clarificatory questions about the task, but the authors provided no input on particular judgments. Full data from the study, along with instructions, is included in the supplementary materials.

C.1 Example

Below, we include an example document from the evaluation study, along with the candidate summaries and their annotated quality scores (in consistent annotator order). MUC-4 documents are uncased, but we have added casing to the document and summaries below for readability.

Document:

> The Government of El Salvador strongly condemns the terrorist attacks carried out with explosives today against the Committee of Mothers of Political Prisoners and Missing in El Salvador and against the Salvadoran Workers National Union Federation. As a result of these attacks, several persons were wounded and others died. The government offers its condolences. The government repudiates these irrational actions and the terrorist attack carried out on 30 October by the FMLN against the Armed Forces staff installations, where 1 civilian died and 14 others were wounded. It is evident that these actions are part of a plot whose objective is to block the dialogue initiated by President Alfredo Cristiani. The government has ordered a careful investigation to determine what happened and to file charges against criminal groups that have unleashed this increased wave of terrorist violence to create a climate of unrest and to block the road to peace. Finally, the Government of El Salvador reaffirms its determination to continue the search for peace through the dialogue, and it states that in spite of these bloody events, Salvadorans must remain calm and not give up hopes to find a solution to the conflict through civilized means.
>
> 31 October 1989, National Secretariat of Communications.

Template:

- `Event Type`: Attack
- `StageOfCompletion`: Accomplished
- `Date`: 30 October
- `Location`: El Salvador
- `PerpInd`: criminal groups
- `PerpOrg`: FMLN
- `Target`: Armed Forces staff installations
- `Victim`: 1 civilian, 14 others
- `Weapon`: -

Summaries:

Reference On 30 October, the FMLN, one of the criminal groups in El Salvador, carried out a terrorist attack on Armed Forces staff installations in El Salvador, resulting in the death of 1 civilian and injuries to 14 others. **Scores**: 5, 5, 3

BART The Government of El Salvador strongly condemns the terrorist attack carried out on 30 October by the FMLN against the Armed Forces staff installations, where one civilian died and 14 others were wounded. It is evident that these actions

are part of a plot to block the dialogue initiated by President Alfredo Cristiani. **Scores**: 4, 3, 4

PEGASUS On 30 October, the FMLN carried out a terrorist attack on the Armed Forces staff installations in El Salvador, where one civilian died and 14 others were wounded. **Scores**: 4, 3, 3

T5 The Government of El Salvador strongly condemns the terrorist attacks carried out with explosives on 30 october by criminal groups affiliated with the FMLN against the Armed Forces staff installations, where one civilian died and 14 others were wounded. **Scores**: 2, 4, 3

ChatGPT On October 30, 1989, in El Salvador, a terrorist attack was carried out by the FMLN against the Armed Forces staff installations, resulting in the death of one civilian and the injury of 14 others. The Government strongly condemned these actions, which were part of a plot to block the dialogue initiated by President Alfredo Cristiani. A careful investigation has been ordered to determine what happened and file charges against the criminal groups responsible for this increased wave of terrorist violence. **Scores**: 5, 5, 4

GPT-4 On October 30, 1989, the FMLN, along with associated criminal groups, carried out a terrorist attack on Armed Forces staff installations in El Salvador. the attack resulted in the death and injury of several civilians. This event, which was part of a broader scheme to disrupt President Alfredo Cristiani's initiated dialogue, has been condemned by the Salvadoran Government who are conducting an investigation. **Scores**: 2, 5, 2

C.2 Judgment-Metric Correlations

Here, we consider the extent to which the judgments from the human evaluation of the model and reference summaries (§6) correlate with automatic metrics. In Table 5, we report Kendall's rank correlation coefficient (two-sided τ-c) between each evaluator's quality judgments on the 180 summaries (30 examples $\times$ 6 candidate summaries/example) and the corresponding metric value for that item, for each metric in Table 2.

Intriguingly, the only reliably positive correlations ($ps < 0.05$) we observe are for annotators B1 ($\mathbf{R}_{1,2,L}$, **BS**, **CR**) and B2 ($\mathbf{R}_{1,2,L}$, **BS**, $D \rightarrow S_p$). Of particular interest is that, *contra* the findings of Chen and Eger (2023), we observe almost no reliable positive correlations among the NLI metrics (B2 on $D \rightarrow S_p$ excepted). One part of the explanation very likely lies in the difference between our dataset and the ones they study, which include SummEval (Fabbri et al., 2021), RealSum (Bhandari et al., 2020), and Rank19 (Falke et al., 2019)—all of which have corpora focused on very different topics from MUC-4/MUCSUM, and which relied on somewhat different (and differently prioritized) evaluation criteria for their judgments.

Another part of the explanation may lie in the fact that Chen and Eger used different entailment-based "formulas" in their summarization results, depending on which performed best on a particular dataset and setting (reference-based vs. reference-free evaluation). We use *one* of these formulas in this work (the entailment probability, e), whereas they further consider others that incorporate the neutral (n) and contradiction (c) probabilities, such as $-c$, $e - n$, and $e - c$.

It is also worth noting that BERTScore (**BS**) is at least not obviously superior in our study to ROUGE (**R**), *contra* findings from Zhang et al. (2019)—though again, differences in the data and judgment task may help account for this.

Even so, we do not think this is cause for a wholesale dismissal of automatic metrics. Each of the metrics considered here *does* convey information about a candidate summary that is likely to be useful in real-world contexts (e.g. degree of lexical overlap is very informative for plagiarism detection)—it is simply *different* information from what is captured by a human judgment. Moreover, the fact that individual *human* judgments can clearly exhibit such variability also suggests that, while they may (must) remain the gold standard for summarization, any *particular* judgment ought to be understood as being nothing more than that.

Annotator	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	**BS**	**CR**	$S_r \rightarrow S_p$	$S_p \rightarrow S_r$	$S_r \leftrightarrow S_p$	$D \rightarrow S_p$
B1	$.136^*_{(.04)}$	$.156^*_{(.02)}$	$.169^*_{(.01)}$	$.127^*_{(.05)}$	$.057_{(.37)}$	$.003_{(.97)}$	$-.030_{(.65)}$	$.004_{(.95)}$	$.085_{(.19)}$
B2	$.208^*_{(.00)}$	$.211^*_{(.00)}$	$.182^*_{(.01)}$	$.232^*_{(.00)}$	$.218^*_{(.00)}$	$.009_{(.90)}$	$.123_{(.07)}$	$.056_{(.41)}$	$.152^*_{(.02)}$
B3	$-.014_{(.81)}$	$-.053_{(.34)}$	$-.064_{(.25)}$	$.001_{(.98)}$	$-.030_{(.58)}$	$-.023_{(.68)}$	$-.033_{(.55)}$	$-.021_{(.70)}$	$.084_{(.13)}$
Avg	$.114_{(.21)}$	$.118_{(.09)}$	$.113_{(.08)}$	$.102_{(.26)}$	$.130_{(.25)}$	$.088_{(.84)}$	$-.006_{(.48)}$	$.021_{(.72)}$	$.119_{(.09)}$

Table 5: Kendall's rank correlation coefficient ($\tau_{\text{(p-val)}}$) between each human evaluator's judgments and the corresponding automatic metric across the 180 judgments (30 examples $\times$ 6 candidate summaries) from the human evaluation study in §6. "Avg" indicates the macro-average correlation across evaluators. "*" denotes significance at $p = 0.05$. Note that these are *not* the same annotators as in Table 4. See discussion in Appendix C.