Prompted Opinion Summarization with GPT-3.5

Adithya Bhaskar[1]
IIT Bombay

Alexander R. Fabbri[2]
Salesforce AI

Greg Durrett[3]
UT Austin

[1]adithyabhaskar@cse.iitb.ac.in
[2]afabbri@salesforce.com
[3]gdurrett@cs.utexas.edu

Abstract

Large language models have shown impressive performance across a wide variety of tasks, including text summarization. In this paper, we show that this strong performance extends to opinion summarization. We explore several pipeline methods for applying GPT-3.5 to summarize a large collection of user reviews in a prompted fashion. To handle arbitrarily large numbers of user reviews, we explore recursive summarization as well as methods for selecting salient content to summarize through supervised clustering or extraction. On two datasets, an aspect-oriented summarization dataset of hotel reviews (SPACE) and a generic summarization dataset of Amazon and Yelp reviews (FewSum), we show that GPT-3.5 models achieve very strong performance in human evaluation. We argue that standard evaluation metrics do not reflect this, and introduce three new metrics targeting faithfulness, factuality, and genericity to contrast these different methods.

1 Introduction

Recent years have seen several shifts in summarization research, from primarily extractive models (Erkan and Radev, 2004; Gu et al., 2022; Kwon et al., 2021; Jia et al., 2020; Zhong et al., 2020) to abstractive models with copy mechanisms (See et al., 2017; Song et al., 2018; Gehrmann et al., 2018) to pre-trained models (Devlin et al., 2019; Isonuma et al., 2021; Lewis et al., 2020; Zhang et al., 2020a; He et al., 2020). GPT-3 (Brown et al., 2020; Wu et al., 2021; Saunders et al., 2022; Goyal et al., 2022) and GPT-4 represent another shift: they show excellent zero- and few-shot performance across a variety of text generation tasks. However, their capabilities have not been extensively benchmarked for opinion summarization. Unlike news, where extractive lead baselines are often highly effective, opinion summarization requires balancing contradictory opinions and a higher degree of abstraction to convey all of the viewpoints faithfully.

In this paper, we apply GPT-3.5, specifically the `text-davinci-002` model,[1] to the task of opinion summarization, focusing on reviews of products, hotels, and businesses. Applying GPT-3.5 in this setting is not straightforward, as the combined length of the reviews or posts may exceed the model's maximum input length. Furthermore, we find that certain styles of inputs can lead to GPT-3.5 simply echoing back an extract of the inputs. To mitigate these issues, we explore a family of pipelined approaches, specifically (1) filtering a subset of sentences with an extractive summarization model, (2) chunking with repeated summarization, and (3) review-score-based stratification. In the context of aspect-oriented summarization, we also explore the inclusion of a sentence-wise topic prediction and clustering step.

We show that our approaches yield high-quality summaries according to human evaluation. The errors of the systems consist of subtle issues of balancing contradictory viewpoints and erroneous generalization of specific claims, which are not captured by metrics like ROUGE (Lin, 2004) or BERTScore (Zhang et al., 2020b). This result corroborates work calling for a re-examination of current metrics (Fabbri et al., 2021; Tang et al., 2023) and the need for fine-grained evaluation (Gehrmann et al., 2022). We therefore introduce a set of metrics, using entailment as a proxy for support, to measure the *factuality*, *faithfulness*, and *genericity* of produced summaries. These metrics measure the extent of over-generalization of claims and misrepresentation of viewpoints while ensuring that summaries are not overly generic.

Our results show that basic prompted GPT-3.5 produces reasonably faithful and factual summaries when the input reviews are short (fewer than 1000 words); more sophisticated techniques do not show much improvement. However, as the input size

[1]The most advanced model available at the time this work was being conducted.

Findings of the Association for Computational Linguistics: ACL 2023, pages 9282–9300
July 9-14, 2023

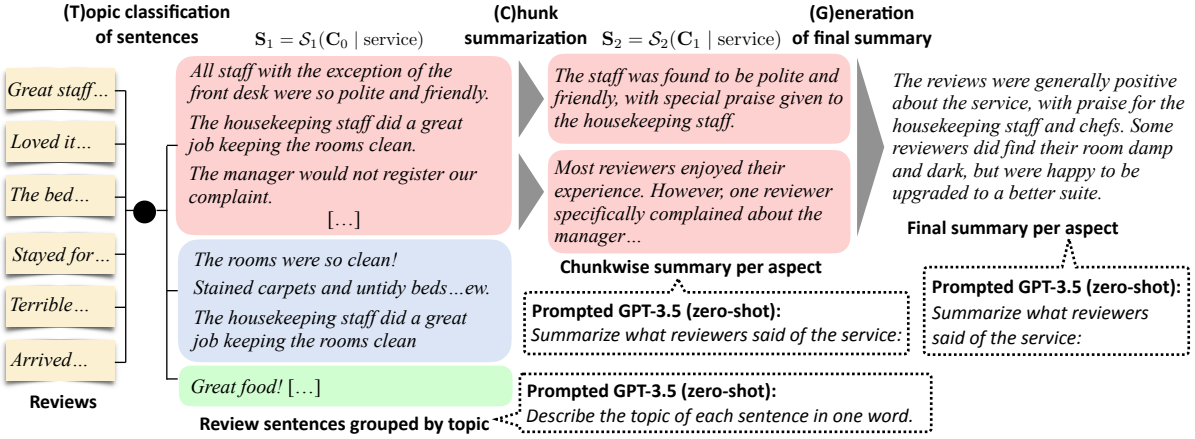

Figure 1: Illustration of the TCG pipeline. Sentences are clustered based on the aspects closest to their topic (T step); examples are shown for rooms, food and service. The relevant cluster is then repeatedly chunked and summarized until the combined length falls below 35 sentences (C step). A final round of GPT-3.5 summarization follows (G step).

grows larger, repeated summarization leads GPT-3.5 to produce generalized and unfaithful selections of viewpoints relative to the first round. We demonstrate that using QFSumm (Ahuja et al., 2022), an extractive summarization model, to filter out sentences prior to GPT-3.5 (instead of multi-level summarization) can slightly help with factuality and faithfulness. The resulting summaries also present a more specific selection of viewpoints but are generally shorter and use a higher proportion of common words. A topicwise clustering and filtering step pre-pended to the pipeline alleviates these issues while relinquishing a portion of the gains on factuality and faithfulness.

Our main contributions are: (1) We introduce two approaches to long-form opinion summarization with GPT-3.5, namely, hierarchical GPT-3.5 summarization with chunking, and pre-extraction with an extractive summarization model. (2) We establish the strength of these approaches with a human study and demonstrate the need for objective and automatic means of evaluation. (3) We develop three entailment-based metrics for factuality, faithfulness, and genericity that are better suited to evaluate extremely fluent summaries as compared to metrics based on n-gram matching. The relevant artifacts and code for this work are publicly available and can be found at https://github.com/testzer0/ZS-Summ-GPT3/.

2 Motivation and Problem Setting

Review summarization involves the summarization of the text of multiple reviews of a given product or service into a coherent synopsis. More formally, given a set of reviews $\mathcal{R} = \{R_i\}_{i=1}^{n}$ with the review R_i consisting of l_i sentences $\{r_{ij}\}_{j=1}^{l_i}$, we define a *summarization system* $\mathcal{S}$ to be a function that takes as input the combined reviews C and then produces k output sentences $S = \{s_i\}_{i=1}^{k}$, written as $S = \mathcal{S}(C)$, where $C \equiv \texttt{combine}(\mathcal{R})$ is typically obtained by concatenating the review sentences. We use the notation `combine` to refer to the combination of both sentences and reviews.

We can also instantiate this pipeline for *aspect-oriented review summarization*, which involves the summarization of multiple reviews conditioned on an aspect a (such as *'cleanliness'*). In particular, the summarization is written as $S = \mathcal{S}(C \mid a)$. We consider aspect-agnostic review summarization as a special case of aspect-oriented review summarization with the aspect *'none'* for notational simplicity.

2.1 Desiderata

Opinion summaries should demonstrate three key characteristics.

First, the summaries should also be **faithful**, i.e., select the most subjectively important viewpoints with the largest consensus. For instance, if five reviews raised the issue of small rooms while eight complained about dusty carpets, the choice (due to a limited output size) to discuss the latter over the former would be considered faithful. Thus, faithfulness is about careful management of the word budget given constrained output length.

The summaries should also be **factual**, i.e., report information grounded in statements that actu-

Pipeline	Constituents
SPACE	
Q	QFSumm
A	AceSum
TCG	Topicwise-Clustering + Recursive GPT-3-Chunking
First-TCG	TCG - Output of first GPT-3-Chunking Layer
QG	QFSumm-long + GPT-3
TQG	Topicwise-Clustering + QFSumm-long + GPT-3
RG	Review-Stratification + Recursive GPT-3-Chunking
First-RG	RG - Output of first GPT-3-Chunking Layer
FewSum	
Q	QFSumm
FS	FewSum
G	GPT-3
QG	QFSumm-long + GPT-3
CG	GPT-3-Chunking + Recursive GPT-3-Chunking
First-CG	CG - Output of first GPT-3-Chunking Layer

Table 1: The pipelines compared for SPACE and FewSum, and their constituents.

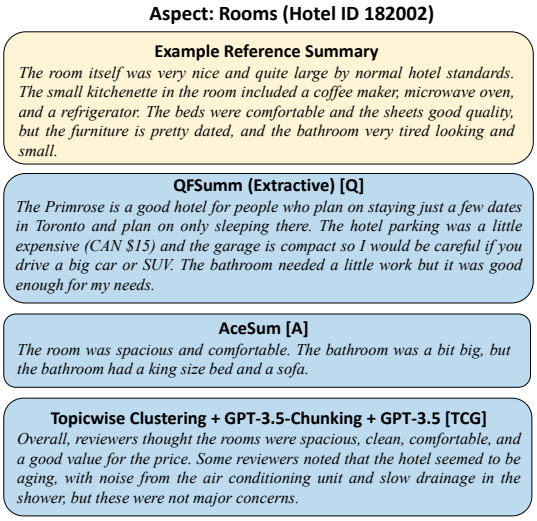

Figure 2: Example summaries from TCG, Q, and A, and a reference summary from the SPACE dataset.

ally do appear in the set of reviews, without containing extrinsic hallucinations. For instance, if five reviews found hotel rooms to be small, but three found them large, the statement *The rooms were large* is considered factual despite the viewpoint being in the minority. By contrast, *A pipe burst and flooded my room* is unfactual if this is never actually reported in the reviews.

Finally, the summaries should be **relevant**: the points raised in them should only discuss topics relevant to the specified aspect. For example, in a summary about the cleanliness of a hotel room, bad food should be omitted even if it was frequently brought up in the reviews.

2.2 Framework

Based on the desiderata, we need to ensure that the summaries represent all of the reviews; however they are too many in number and too long in combined length. We, therefore, define a *summarization pipeline* to be a series of summarization systems $\mathcal{S}_1, \cdots, \mathcal{S}_m$ where each system takes as input the condensed results of the previous system. Specifically,

$$S_0 = \mathcal{R}, \;\; C_i = \texttt{combine}(S_{i-1}), \;\; S_i = \mathcal{S}_i(C_i)$$

We showcase an example pipeline in Figure 1, with one stage extracting the relevant sentences from the reviews and the next summarizing the extracted sentences.

3 GPT-3.5 Summarization Pipelines

The components of our summarization pipelines may be broadly categorized into *extractors* and *summarizers*, which we describe next. More details can be found in Appendix A. First, *extractors* select relevant parts of a set of reviews, optionally conditioned on an aspect. Our extractors include:

GPT-3.5 Topic Clustering (T) We prompt GPT-3.5 to produce a single word topic for each sentence, which we map to the closest aspect with GloVe (Pennington et al., 2014) similarity. This defines a set of sentences to be used for aspect-based summarization. This step is only used for pipelines on SPACE, as FewSum is aspect-agnostic.

QFSumm-long (Q) We use the aspect-specific *extractive* summarization model introduced in (Ahuja et al., 2022) to extract up to 35 most relevant sentences from the input text. QFSumm was designed to allow extremely long inputs, and thus no truncation is required at this stage.

Review Stratification (R) This involves clustering reviews by reviewer scores (given in the dataset) and summarizing each cluster with GPT-3.5.

In addition to extractors, we also utilize **GPT-3.5-chunking (C)** in some of our pipelines. We segment the sentences from the prior step into non-overlapping chunks, then summarize each individually with GPT-3.5. The results are then concatenated for the next step.

Our *summarizers* summarize the text one final time to produce the output summary. All of our pipelines use GPT-3.5 as the summarizer. However, we also compare to QFSumm (Ahuja et al., 2022), AceSum (Amplayo et al., 2021a) and the model

Pipeline	ROUGE-1	ROUGE-L	BERTScore
SPACE			
Q	19.2	16.7	85.4
A	**32.4**	**30.2**	**89.8**
TCG	23.5	20.6	88.7
QG	25.1	22.1	89.1
TQG	25.2	22.3	89.0
RG	23.0	20.5	88.5
FewSum - Amazon			
Q	27.0	24.3	86.2
FS	**32.5**	**29.6**	**88.8**
G	27.0	23.9	88.7
QG	26.2	23.7	88.4
CG	25.7	22.9	88.6
FewSum - Yelp			
Q	23.8	20.6	84.3
FS	**34.1**	**31.4**	**89.0**
G	26.1	21.4	88.4
QG	27.1	22.1	88.5
CG	26.5	21.5	88.2

Table 2: ROUGE-1, ROUGE-L, and BERTScore (F1) for the compared models.

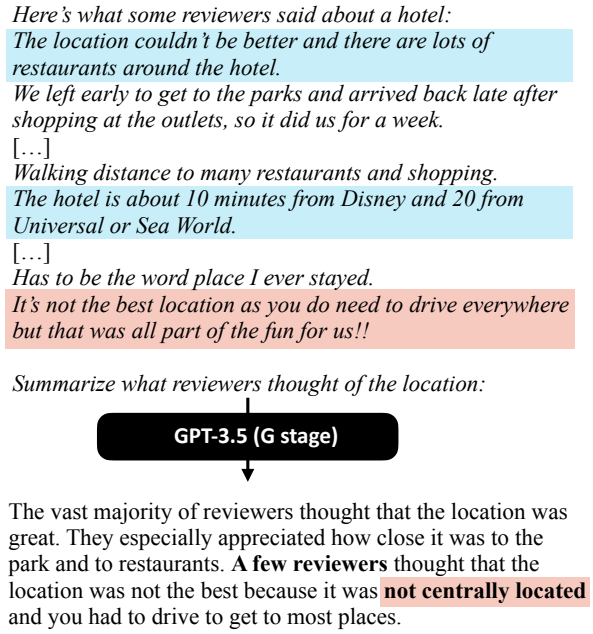

Figure 3: Example of errors made by GPT-3.5. The viewpoint of a single reviewer is wrongly expressed as that of a "few reviewers" and generalized to the hotel not being centrally located, contradicting other reviews (blue).

released with FewSum (Bražinskas et al., 2020a), also referred to as FewSum.

These building blocks are composed to build various summarization pipelines, which we list in Table 1. An illustration of one pipeline (TCG) is shown in Figure 1. Since topic-wise clustering is unnecessary for FewSum (due to lack of aspects), we only compare G (vanilla GPT-3.5 used to summarize the set of product reviews, truncated to fit if necessary), CG (Chunking + GPT-3.5), QG (QFSumm-long + GPT-3.5), Q (QFSumm), and FS (FewSum) for this dataset. The table also lists some approaches that are the first stages of pipelines that begin with GPT-3.5-chunking, which we also compare against in Section 5.

4 Evaluation

	SPACE	FewSum
Average #reviews per entity	100.00	22.41
Average #sentences per review	9.16	3.37
Average #words per sentence	17.56	12.12

Table 3: SPACE and FewSum dataset statistics.

4.1 Datasets

SPACE (Amplayo et al., 2021a) involves the summarization of reviews of hotels along the aspects {*general, rooms, building, cleanliness, location, service, food*} and provides three human-written summaries for each *(hotel, aspect)* pair. Table 3 shows that the reviews of SPACE are too long to summarize with a non-pipelined system given `text-davinci-002`'s context window size. We exclude the *general* aspect from our experiments.

FewSum (Bražinskas et al., 2020a) contains product reviews from Amazon and Yelp. As opposed to SPACE, FewSum is not aspect-oriented, and the reviews are typically much shorter. For many of the products, the combined length of the reviews falls below 900 words, enabling direct summarization with GPT-3.5. FewSum provides three gold summaries for only a small portion of the products. Across these two splits, FewSum provides golden summaries for 32 and 70 products in the Amazon and Yelp categories respectively.

We list SPACE and FewSum statistics in Table 3.

4.2 Automatic Eval: ROUGE and BERTScore

We compute ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020b) and show results in Table 2.

The BERTScores for AceSum, as well as all GPT-3-related models, are in the range of $88 - 90$, and differences in performance are unclear. AceSum achieves the highest ROUGE-1 as well as ROUGE-L scores by far, and is followed by TQG and QG. QFSumm does particularly poorly on the

Pipeline	Factuality	Represent-ativeness	Faithful-ness	Relev-ance
TCG	2.85	2.99	4.86	**4.60**
TQG	2.86	2.95	4.83	4.32
QG	2.88	2.97	4.79	3.93
A	**3.00**	2.96	**4.91**	3.62
Q	**3.00**	**3.00**	4.88	2.30
Maximum	3	3	5	5
Fleiss-Kappa	0.64	0.49	0.49	0.64

Table 4: Results of Human Evaluation on the SPACE dataset. Colors indicate moderate (light green) and substantial (darker green) agreement, respectively.

Pipeline	Factuality	Represent-ativeness	Faithful-ness	Relev-ance
G	2.63	2.89	4.68	**4.98**
CG	2.72	2.95	**4.73**	**4.98**
QG	2.68	2.90	4.63	**4.98**
Q	**2.96**	**2.98**	4.52	4.92
FS	2.74	2.32	4.30	4.90
Maximum	3	3	5	5
Fleiss-Kappa	0.26	0.53	0.19	0.15

Table 5: Results of Human Evaluation on the FewSum dataset. Colors indicate moderate (light green), fair (yellow) and slight (red) agreement respectively.

ROUGE scores. The scores are all in the same ballpark on FewSum apart from FS, with it being difficult to draw any conclusions. The latter achieves the highest ROUGE-L as well as BERTScore. The GPT-3.5 systems perform slightly better than QFSumm on the Yelp split which we attribute to the smaller combined review lengths of Yelp.

We argue that these scores are not informative and that they are at times unreliable when comparing the quality of two summaries. ROUGE and BERTScore have been critiqued in prior work as inaccurate indicators of summary quality (Fabbri et al., 2021; Liu and Liu, 2008; Cohan and Goharian, 2016), particularly as the fluency and coherence of the outputs increase to near-human levels (Goyal et al., 2022). Figure 2 demonstrates this by with an example. n-gram methods penalize GPT-3.5 for generating summaries in a slightly different style: "*The reviewers found the rooms to be clean*" instead of "*The rooms were clean*." Similarly, the extractive nature of QFSumm drives it to produce sentences like "*We were served warm cookies on arrival*." While its selections are factual, they are not completely representative of the review opinions themselves. The actual mistakes in our systems include over-generalization and misrepresentation of viewpoints of popularities thereof, which are not well-represented by matching n-grams. Figure 3 shows an example of such errors. We conclude that metrics benchmarking the summaries on different dimensions are necessary.

4.3 Human Evaluation

For a more reliable view of performance, we manually evaluated the summaries of the pipelines TCG, TQG, AceSum (A) and QFSumm (Q) for 50 randomly chosen *(hotel, aspect)* pairs from the SPACE dataset, and G, CG, QG, Q and FS for 50 randomly chosen products (25 each from the *Amazon* and *Yelp* splits) from the FewSum dataset. The axes of evaluation were the attributes established in Subsection 2.1, namely *Factuality*, *Faithfulness* and *Relevance*. In addition, as we often observed our systems produce summaries of the form "*While most reviewers thought ..., some said ...*" to highlight contrasting opinions, we also evaluate on *Representativeness*. Representativeness is a more restricted form of Faithfulness that measures if the more popular opinion was exhibited between two opposing ones. For instance, if four people found the rooms of a hotel clean but two did not, the summary is expected to convey that the former was the more popular opinion.

The three authors of this paper independently rated the summaries along the above axes on Likert scales of 1-3 for both variations of factuality, and 1-5 for faithfulness and relevance. The average scores, along with the Krippendorff's Alpha and Fleiss Kappa scores (measuring consensus among the raters) are presented in Table 4. Among the compared pipelines, TCG improves upon TQG and QG substantially in terms of relevance. All three have a very high score under Factuality, showing that GPT-3.5 models seldom make blatantly wrong statements. Viewpoints selected by QFSumm are generally faithful, and factual due to their extractive nature, but may include irrelevant statements.

We list the corresponding metrics for FewSum in Table 5. CG tends to perform well, but the consensus is low for Faithfulness and Relevance. FS performs poorly across the board due to hallucinated statements harming its Factuality and bad viewpoint selection resulting in low Faithfulness. The lack of aspects may contribute to the low agreement on FewSum; dimensions such as Relevance may be considered underconstrained, and thus more difficult to agree upon in this setting (Kryscinski et al., 2019).

We remark that all of our systems are achieving close to the maximum scores; the small differences belie that the pipelines all demonstrate very strong performance across the board.

5 New Tools for Evaluation and Analysis

Enabling fast automatic evaluation of systems will be crucial for the development of future opinion summarizers. Furthermore, when a large number of reviews are presented to a system, it may be nearly impossible even for a dedicated evaluator to sift through all of them to evaluate a summary. We investigate the question of how we can automate this evaluation using existing tools.

One of the areas where automatic evaluation may help is **faithfulness**. Since faithfulness represents the degree to which a system is accurate in representing general consensus, it requires measuring the proportion of reviews supporting each claim of a summary. A viewpoint with larger support is more popular and, consequently, more faithful. Our key idea is to use entailment as a proxy for support. Past work (Goyal and Durrett, 2021; Laban et al., 2022) has used Natural Language Inference (NLI) models to assess summary factuality by computing entailment scores between pairs of sentences.

However, the summaries produced by GPT-3.5 and related pipelines often consist of compound sentences that contrast two viewpoints. In addition, GPT-3.5 prefers to say "*The reviewers said...*" instead of directly stating a particular viewpoint. We found these artifacts to impact the entailment model. We use a split-and-rephrase step to split these sentences into atomic value judgments by prompting GPT-3.5 as shown in Figure 4. We then use the zero-shot entailment model from SummaC (Laban et al., 2022) to compute the entailment scores for these atomic value judgments. Similar to the approach in the SummaC paper, we observe that a summary statement is factual when strongly entailed by at least one sentence and thus select the top entailment score of each summary sentence as its **factuality score**, and aggregate this score to produce per-system numbers. The choice of the model as well as that of using GPT-3.5 for the split-and-rephrase step are explained further in Appendix B, and the relevant metric of abstractiveness is discussed in Appendix D.

A system could potentially game this metric by producing relatively "safe" statements (like *most reviewers found the rooms clean*). We therefore also want to evaluate **genericity**.

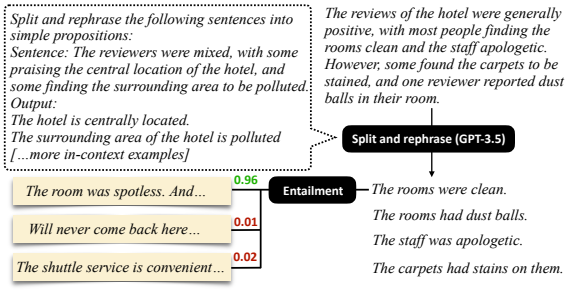

Figure 4: Per-sentence entailment scores are calculated by taking the maximum among the various candidates.

5.1 Terminology

The set of sentences in the summary of the reviews of a hotel $h \in \mathcal{H}$ w.r.t aspect $a \in \mathcal{A}$ is called $S_{h,a}$. Passing these to the split-and-rephrase step gives us a set of split sentences $Z_{h,a}$. For any two sentences s_1, s_2 we denote the entailment score of s_2 with respect to s_1 according to the SummaC-ZS (Laban et al., 2022) model by $e(s_1, s_2) \in [-1.0, 1.0]$. A score of 1.0 indicates perfect entailment while that of -1.0 denotes complete contradiction. Finally, we denote by $N_n(s)$ the (multi-)set of n-grams (with multiplicity) of the sentence s. In particular, $N_1(s)$ is the set of words in the sentence s.

5.2 Evaluation of Entailment

We first evaluate whether entailment is effective at identifying the support of the mentioned viewpoints by human evaluation. The three authors of this paper marked 100 random pairs (50 each from SPACE and FewSum) of sentences and assertions entailed with a score above 0.5 on the scale of $0-2$. Here, 2 indicates that the assertion is completely supported, and 1 that the assertion's general hypothesis is supported, but some specifics are left out. The average score of the selection across the raters was **1.88** with a Fleiss Kappa consensus score of 0.56 (moderate agreement). Many of the lower-rated entailed sentences also had lower entailment scores (closer to 0.5). The score illustrates that the precision of the entailment approach is high.

5.3 Faithfulness: Support Set Sizes

We propose an entailment metric for determining how the viewpoints in the summary reflect the consensus of the input. We first compute per-sentence entailment scores as shown in Figure 4. For each sentence of the split-and-rephrased summary, we

Pipeline	Percentage of split-and-rephrased sentences with n supports			
SPACE				
	$n=0$	$n=1$	$n=2-4$	$n=5+$
Q	8.1	29.0	21.2	41.8
A	7.7	8.6	12.7	71.0
First-TCG	18.7	16.8	18.1	46.2
TCG	22.8	16.9	19.4	41.0
QG	14.9	16.6	16.3	52.2
TQG	18.6	19.2	17.8	44.4
First-RG	23.7	22.0	19.9	34.4
RG	27.4	22.1	20.8	29.6
FewSum				
(Amazon)	$n=0$	$n=1$	$n=2-4$	$n=5+$
Q	9.5	51.6	26.1	12.7
FS	76.9	11.2	8.39	3.45
G	28.0	32.4	27.7	12.0
QG	27.6	34.7	23.6	14.2
First-CG	27.8	26.6	25.0	20.5
CG	31.9	32.2	22.6	13.3
(Yelp)	$n=0$	$n=1$	$n=2-4$	$n=5+$
Q	8.2	46.2	31.3	14.2
FS	52.3	17.1	20.0	10.6
G	27.2	24.3	29.3	19.3
QG	30.6	30.3	27.4	11.6
First-CG	24.4	25.6	26.8	23.3
CG	26.3	28.3	26.2	19.2

Table 6: Percentages of split-and-rephrased sentences binned according to support sizes, for all compared pipelines. The threshold used is $\tau = 0.75$.

measure the number of review sentences that entail it with a score greater than a threshold $\tau = 0.75$ (the "support" of the sentence). This threshold was determined based on manual inspection. We bin these counts into $0, 1, 2-4$ and $5+$. The frequencies of the bins are converted to percentages and listed in Table 6. FS performs poorly due to presenting hallucinated viewpoints, and repeated summarization slightly hurts CG on the Amazon split. G and CG outperform other methods on the Yelp split, likely because it has fewer reviews per product than Amazon, making it much likelier for the combined reviews of a product to fit in a manageable number of words. The "pure" GPT-3.5 systems generally perform well on the short review sets of FewSum. As we move to the long combined lengths of the reviews on SPACE, however, the pure GPT-3.5 pipelines fall behind in terms of faithfulness. Repeated summarization causes a major dip from First-TCG to TCG, indicating that this is not effective for long-form inputs. QG outperforms other GPT-3-related pipelines by a large margin. As we saw in human evaluation, however,

Pipeline	Average Top Score	Pipeline	Average Top Score	
SPACE		FewSum		
Q	91.59		**(Amazon)**	**(Yelp)**
A	**92.49**	Q	**85.29**	**86.62**
First-TCG	84.96	FS	24.36	47.23
TCG	82.06	G	65.81	68.59
QG	87.50	QG	67.63	65.04
TQG	84.68	First-CG	68.34	69.86
First-RG	81.54	CG	66.43	68.58
RG	79.85			

Table 7: The average Top Score for each pipeline on the SPACE and FewSum datasets.

QG may include some irrelevant viewpoints in this process. Abating this behavior by performing a topic-clustering step first brings its numbers down to a level comparable with First-TCG, which is still more faithful than the TCG pipeline. AceSum has the largest number of statements with $5+$ supports on the SPACE; however, as we will see later, many of its summaries are very generic, and support for them can be easily found among the large number of reviews. Q has the smallest percentage of statements with no support because it is extractive.

5.4 Factuality: Top Score

As depicted in Figure 4, averaging the per-sentence entailment scores (first per-summary, then per-system) gives us the *Top Score* metric. The average top score is a proxy for factuality since true statements will typically be strongly entailed by at least one sentence of the reviews. We list the computed average top scores in Table 7. FS performs poorly on FewSum in terms of Factuality. The numbers for other systems are similar, with QG and CG performing best on the Amazon and Yelp splits. However, on the longer inputs of SPACE, the differences in factuality become more apparent. In particular, to reconcile similar but distinct viewpoints, repeated summarization leads to a type of generalizing that hurts the factuality of TCG and TG. Among the GPT-3.5 pipelines, QG performs the best, followed by TQG. TQG yet again delivers performance comparable to First-TCG and therefore presents a reasonable trade-off with some gains on factuality and increased relevance.

5.5 Genericity

As mentioned before, we want to measure whether reviews contain largely generic statements like *the service was helpful*, which are likely to be faithful

Pipeline	Genericity		Percentage of scores greater than τ	
SPACE				
Q	**0.640**		**64.6**	
A	0.828		82.8	
TCG	0.781		80.1	
QG	0.759		76.5	
TQG	0.738		73.7	
RG	0.788		80.0	
FewSum				
	(Amazon)	**(Yelp)**	**(Amazon)**	**(Yelp)**
Q	**0.339**	**0.406**	**32.6**	**37.8**
FS	0.529	0.636	54.2	62.6
G	0.582	0.654	56.9	65.2
QG	0.565	0.653	53.9	64.7
First-CG	0.604	0.732	63.4	69.1
CG	0.554	0.682	56.7	68.1

Table 8: Semantic genericity based on entailment, along with the raw percentage of scores above the threshold. The threshold used is $\tau = 0.5$.

Pipeline	Average IDF	Pipeline	Average IDF	
SPACE		FewSum		
Q	**12.00**		**(Amazon)**	**(Yelp)**
A	5.77	Q	**4.38**	**4.33**
TCG	8.40	FS	3.16	3.26
QG	6.93	G	3.02	2.93
TQG	7.82	QG	3.10	2.93
RG	8.87	CG	3.00	2.86

Table 9: Measurement of lexical genericity. Average IDF (larger is better) for the compared pipelines. The FewSum pipelines report lower ranges for average IDF due to fewer total number of documents.

and factual but not very useful to a user of a system.

We first focus on *semantic* genericity, i.e. the use of statements generally applicable to other products/services in the same class. On the other hand, *lexical* genericity involves the overuse of generic words and is tackled next. Our approach to measuring semantic genericity employs the observation that generic sentences from a summary are often widely applicable and thus likely to be strongly entailed by statements from other summaries. We calculate the similarity $\texttt{sim}(S, S')$ of two sets of sentences using the averaged top score, as Figure 4 shows. Similarly, we also measure the fraction $\texttt{frac}(S, S', \tau)$ of sentences whose top score exceeds a threshold τ. Equation 1 computes the average similarity score between sentences that belong to two reviews by the same system but different

Evaluation Axis	Entailment-Based Metric	ROUGE
Factuality	0.36	0.05
Faithfulness	0.29	-0.03

Table 10: Spearman Correlation Coefficients of our metrics and ROUGE with human judgments.

(*hotel, aspect*) pairs (normalizing by the number of pairs N). Equation 2 computes the corresponding metric based on `frac`.

$$G = \frac{1}{N} \sum_{(h,a) \neq (h',a')} \texttt{sim}(Z_{h,a}, Z_{h',a'}) \quad (1)$$

$$F_\tau = \frac{1}{N} \sum_{(h,a) \neq (h',a')} \texttt{frac}(Z_{h,a}, Z_{h',a'}, \tau) \quad (2)$$

We report these two metrics in Table 8. On the short inputs of FewSum, all GPT-3.5 pipelines give similar results, with FewSum being slightly less generic. Moving to SPACE, however, the range of scores becomes much wider. Forced to reconcile disparate opinions during repeated summarization, TCG and RG produce generic summaries, although AceSum is the most generic. We note that pre-extraction with QFSumm and Topic-wise clustering help QG and TQG remain less generic.

To measure *lexical genericity*, we use the sentences from *all* summaries on the corresponding dataset as the set of documents to calculate an averaged Inverse Document Frequency (IDF) of the summaries, with stopwords removed and stemming applied. Since generic words are likely to occur more frequently and therefore have a low IDF, a smaller score indicates higher genericity. The scores calculated this way are listed in Table 9. As expected, QFSumm is highly specific due to being extractive. We observe that AceSum generates summaries that over-use generic words, in line with our prior observations. We also note that pre-extraction with QFSumm helps with lexical genericity as it did with semantic genericity. Finally, on FewSum, we observe that FS does better than every other pipeline apart from Q. This bolsters our previous claim that its low Factuality and Faithfulness scores were due to hallucinated, but specific, viewpoints.

5.6 Correlation with Human Judgments

Our entailment-based approaches set out to measure Factuality and Faithfulness; how well do these

correlate with our human evaluation? We compute Spearman's rank correlation coefficient on the human-annotated SPACE examples with the averaged annotator scores, as the consensus among rater scores was high on that dataset. In particular, we use the average of the Factuality scores among the raters as the net human score on Factuality on an example and the mean score on Faithfulness as that for Faithfulness. Correspondingly, we consider the Top Score metric as the automatic measurement of Factuality and the percentage of statements with 3 or more supports as Faithfulness. We list the obtained Spearman correlation coefficients in Table 10. While there is room for stronger metrics, the fact that the introduced metrics correlate with human judgments better than ROUGE provides an encouraging signal that these target the factors of interest.

6 Related work

Text Summarization Historically, most work tackling text summarization has been *extractive* in nature (Ku et al., 2006; Paul et al., 2010; Carenini et al., 2006; Angelidis and Lapata, 2018), with more recent work applying pre-trained extractive systems to this task (Zhong et al., 2020; Jia et al., 2020; Kwon et al., 2021; Gu et al., 2022; Ahuja et al., 2022). *Abstractive* approaches (Carenini et al., 2006; Ganesan et al., 2010; Di Fabbrizio et al., 2014) to summarizing reviews have become more successful in recent years (Liu and Lapata, 2019a; Bražinskas et al., 2020b; Amplayo et al., 2021b; Isonuma et al., 2021). We follow in this vein, capitalizing on the strength of GPT-3.5.

Multi-Stage Summarization Most systems of both types are now end-to-end (Liu and Lapata, 2019b; Du et al., 2022; Ahuja et al., 2022). However, multi-stage approaches (Chen and Bansal, 2018; Li et al., 2021; Zhang et al., 2022) like ours have recently shown great promise. For instance, Li et al. (2021) extracts relevant evidence spans and then summarizes them to tackle long documents. Recursive summarization has been explored in (Wu et al., 2021) for book summarization, but involved fine-tuning GPT-3.5 to the task. Other approaches such as the mixture-of-experts re-ranking model Ravaut et al. (2022) can be considered as a two-step approach where the `combine` function ranks and filters the outputs of the first stage.

Evaluation Metrics The domain of news summarization has recently seen interest in using factuality/faithfulness for evaluation (Scialom et al., 2021; Kryscinski et al., 2020; Tang et al., 2023). In news, faithfulness and factuality are quite similar, as news articles usually do not present incorrect information or conflicting opinions. Opinion summarization is therefore quite distinct in this regard, and a separate treatment of factuality and faithfulness is sensible. For the same reason, although unified approaches to evaluating text generation (Deng et al., 2021; Zhong et al., 2022) are useful, more targeted metrics are likely to be more informative for opinion summarization specifically.

Aspect-Oriented Summarization In addition to opinion summarization (Amplayo et al., 2021a), aspect-oriented summarization has also been explored in other domains of NLP (Bahrainian et al., 2022; Yang et al., 2022). However, as highlighted above, opinion summarization differs from news summarization with respect to desired characteristics, and this work focuses specifically on those issues.

7 Conclusion

In this work, we show that GPT-3.5-based opinion summarization produces highly fluent and coherent reviews, but is not perfectly faithful to input reviews and over-generalizes certain viewpoints. ROUGE is unable to capture these factors accurately. We propose using entailment as a proxy for support and develop metrics that measure the faithfulness, factuality, and genericity of the produced summaries. Using these metrics, we explore the impact of two approaches on controlling the size of the input via pre-summarization on two opinion summarization datasets. With the reasonably sized inputs of FewSum, GPT-3.5 and CG produce faithful and non-generic outputs. However, as we move to long-form review summarization, the factuality and faithfulness of these approaches drop. A pre-extraction step using QFSumm helps in this setting but leads to generally shorter and more generic summaries; a topic clustering step can then make summaries less generic and more relevant at a small cost to faithfulness and factuality. We hope that our efforts inspire future improvements to systems and metrics for opinion summary evaluation.

Limitations

Our study here focused on the most capable GPT-3.5 model, `text-davinci-002`, at the time the experiments were conducted. We believe that models like ChatGPT and GPT-4, as well as those in the future, are likely to perform at least as well as these, and if they improve further, the metrics we have developed here will be useful in benchmarking that progress. However, significant further paradigm shifts could change the distribution of errors in such a way that certain of our factors (e.g., genericity) become less critical. In addition, the latest iterations of GPT have a much greater input window size, which help them digest much larger swaths of text in one go and potentially make our pipelined approaches less needed in certain settings.

Furthermore, the `text-davinci-002` model is fine-tuned with data produced by human demonstrations. The precise data used is not publicly available, so it is difficult to use our results to make claims about what data or fine-tuning regimen leads to what failure modes in these models.

Recent work has noted that language models may be susceptible to learning biases from training data (Sheng et al., 2019; Wallace et al., 2019; Shwartz et al., 2020), and this phenomenon has also been observed for GPT-3.5 (Lucy and Bamman, 2021). We did not stress test the models studied for biases and furthermore only experimented on English-language data.

When properly used, the summarization models described in this paper can be time-saving. However, as noted above, summary outputs may be factually inconsistent with the input documents or not fully representative of the input, and in such a case could contribute to misinformation. This issue is present among all current abstractive models and is an area of active research.

Acknowledgments

This work was partially supported by NSF CAREER Award IIS-2145280, a grant from Open Philanthropy, a gift from Salesforce, Inc., and a gift from Adobe. Thanks as well to the anonymous reviewers for their helpful comments.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021a. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021b. Unsupervised opinion summarization with content planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12489–12497.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Seyed Ali Bahrainian, Sheridan Feucht, and Carsten Eickhoff. 2022. NEWTS: A corpus for news topic-focused summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 493–503, Dublin, Ireland. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020a. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020b. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Giuseppe Carenini, Raymond Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative

text. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 305–312, Trento, Italy. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 806–813, Portorož, Slovenia. European Language Resources Association (ELRA).

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

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.

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.

Yanjun Gao, Ting-Hao Huang, and Rebecca J. Passonneau. 2021. ABCD: A graph framework to convert complex sentences to a covering set of simple sentences. 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 3919–3931, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *arXiv preprint arXiv:2202.06935*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News Summarization and Evaluation in the Era of GPT-3. *arXiv*.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. CTRLsum: Towards Generic Controllable Text Summarization. *arXiv*.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2021. Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *Transactions of the Association for Computational Linguistics*, 9:945–961.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3622–3631, Online. Association for Computational Linguistics.

Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. 2021. BiSECT: Learning to split and rephrase sentences with bitexts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6193–6209, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

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

Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, and Manabu Okumura. 2021. Considering nested tree structure in sentence extractive summarization with pre-trained transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4039–4044, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421, Dublin, Ireland. 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.

Haoran Li, Arash Einolghozati, Srinivasan Iyer, Bhargavi Paranjape, Yashar Mehdad, Sonal Gupta, and Marjan Ghazvininejad. 2021. EASE: Extractive-abstractive summarization end-to-end using the information bottleneck principle. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 85–95, Online and in Dominican Republic. 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.

Feifan Liu and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019a. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit.

Li Lucy and David Bamman. 2021. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, Virtual. Association for Computational Linguistics.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Michael Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 66–76, Cambridge, MA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022. SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524, Dublin, Ireland. Association for Computational Linguistics.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Vered Shwartz, Rachel Rudinger, and Oyvind Tafjord. 2020. "you are grounded!": Latent name artifacts in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6850–6861, Online. Association for Computational Linguistics.

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1717–1729, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yahvuz, Wojciech Kryściński, Justin F. Rousseau, and Greg Durrett. 2023. Understanding Factual Errors in Summarization: Errors, Summarizers, Datasets, Error Detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively Summarizing Books with Human Feedback. *arXiv*.

Xianjun Yang, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Xiaoman Pan, Linda Petzold, and Dong Yu. 2022. Oasum: Large-scale open domain aspect-based summarization.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. 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. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Pipeline Details

A.1 Details of the Infrastructure, Models, and Datasets Used

Computational Resources All experiments were run on a machine equipped with an Intel Xeon W-2123, and utilized a TITAN RTX GPU with a 24 GB memory. We estimate the total computational GPU budget to be roughly 100 GPU-hours.

Model Sizes QFSumm (Ahuja et al., 2022) is a fine-tuned version of BERT and therefore has 110M parameters. The FewSum model from (Bražinskas et al., 2020a) has 25.1M parameters including the plug-in network. AceSum (Amplayo et al., 2021a) has a combined total of 142M parameters between the Controller Induction Model and Opinion Summarization Model. We use the VitC variant of the entailment model SummaC-ZS (Laban et al., 2022), which relies on the ALBERT-xlarge architecture with 60M parameters. For all models, we used the default parameters as reported in Ahuja et al. (2022), Bražinskas et al. (2020a), Amplayo et al. (2021a), and Laban et al. (2022). Consequently, no hyperparameter search was necessary. All models have been publicly released under the MIT License on GitHub by the respective authors.

Datasets and Evaluation Both the SPACE and FewSum datasets consist of reviews in English. The former consists of reviews of hotels, and the latter product reviews from Amazon and service reviews from Yelp. We are using pre-existing datasets that are standard in opinion summarization. Through our human evaluation, we did not see any personal identifying information or offensive content in the reviews we assessed. All of our human evaluation experiments were performed once by the authors, and we report the Krippendorff's Alpha and Fleiss Kappa scores as measurements of consensus. We used ROUGE with the default settings.[2] We used NLTK's (Loper and Bird, 2002) WordNet (Miller, 1994) lemmatizer as the lemmatizer where needed. Sentence splitting was done using the `sent_tokenize()` function of NLTK.

A.2 Details of the Configurations and Prompts

Here we provide more details of the configuration and/or prompts used for various models. Below, GPT-3.5 refers to the `text-davinci-002` model.

QFSumm and QFSumm-long (Q) QFSumm allows one to specify the number n of sentences to extract from the reference text to shape into a summary. We use $n = 3$ (the default setting) for QFSumm (summarizer) and $n = 35$ for QFSumm-long (extractor). On the SPACE dataset, we use the aspect-specific keywords from Ahuja et al. (2022) to pass to the model. On the FewSum dataset, however, the set of relevant keywords may be drastically different across examples. Therefore, for each product, we pass 5 randomly chosen reviews to GPT-3.5 with the prompt consisting of the reviews and the directive "*Output up to eight comma-separated keywords that capture these reviews most saliently:*". The produced keywords are then used with QFSumm to summarize the reviews.

GPT-3.5 Topic Clustering (T) The prompt we use is "*Describe the topic of each sentence in one word*", followed by three examples and then the sentence whose topic is to be determined. We then map the produced words to their corresponding normalized GloVe (Pennington et al., 2014) vectors, which are then mapped to the closest aspects in terms of L2 distance. This is functionally equivalent to using cosine similarity as the vectors are normalized.

GPT-3.5 Chunking (C) We strive for the length of the chunks (in sentences) to be both as close to each other and to 30 as possible; thus, when there are l sentences total to be chunked, we take $c = \lceil \frac{l}{30} \rceil$ to be the number of chunks, and allocate $\lfloor \frac{l}{c} \rfloor$ sentences to each chunk (except the last one, which may have fewer).

Review Stratification (R) If a cluster's length exceeds GPT-3.5's upper limit at this stage, it is truncated to the maximum number of sentences that fit.

GPT-3.5 (G) When used as a summarizer, we feed the penultimate set of sentences to GPT-3.5 with the prompt *"Summarize what the X said of the Y:,"* where X is either "*reviewers*" or "*accounts*" based on whether GPT-3.5-chunking was used so far. Y is the aspect being summarized (SPACE) or just "*Product*" (FewSum). The preamble is either *"Here's what some reviewers said about a hotel:"* or *"Here are some accounts of what some reviewers said about the hotel"* in the case of SPACE. The word "*hotel*" is replaced by "*product*" for FewSum.

[2] The `rouge.properties` file at https://github.com/kavgan/ROUGE-2.0

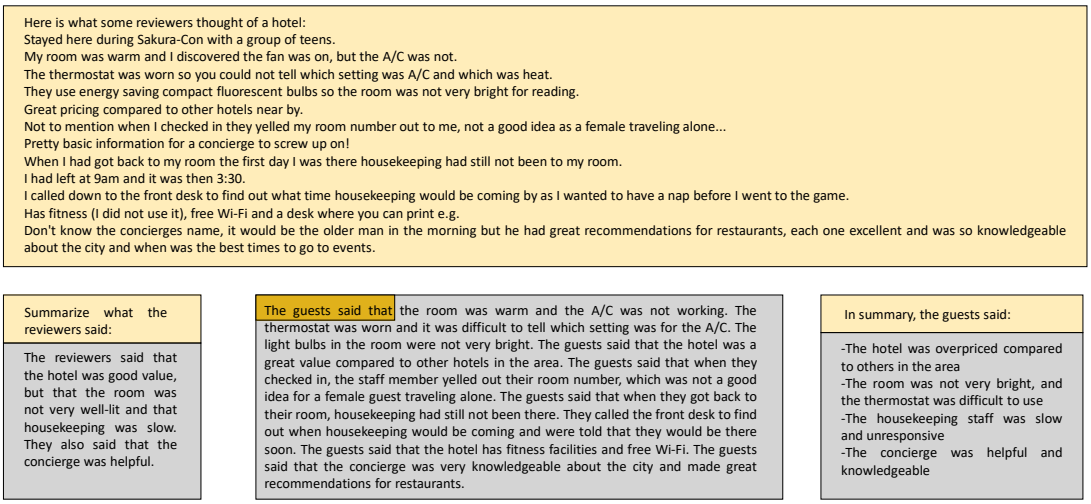

Figure 5: Aspects of summarization such as verbosity or the format of output are affected by the specific wording of the prompt. We use the leftmost prompt, "*Summarize what the reviewers said.*"

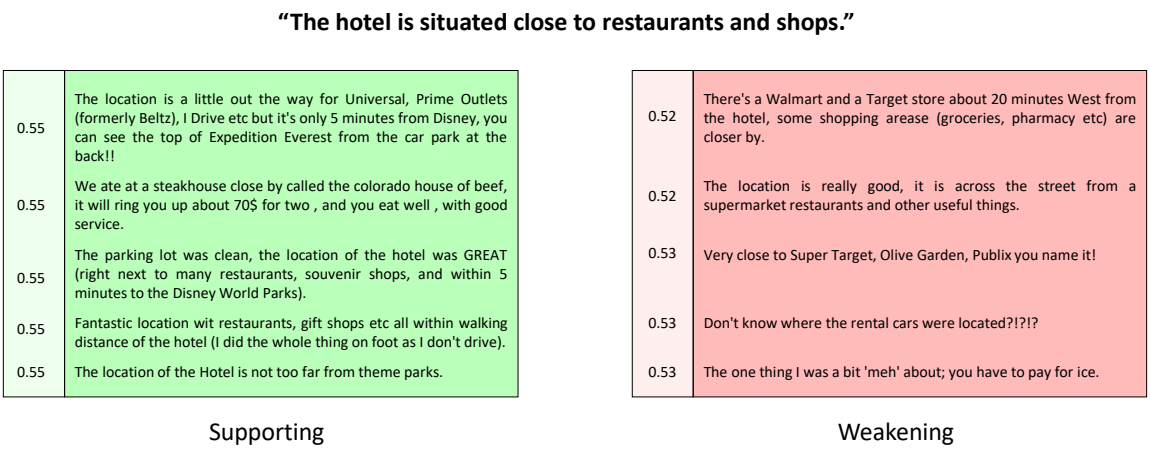

Figure 6: The top 5 supporting and weakening sentences from the reviews for the statement "*The hotel is situated close to restaurants and shops*" as found by the **Conv** SummaC model. The corresponding entailment scores are included in parentheses. We see that the scores are very close to each other and that the "weakening" statements do not weaken the statement at all. These issues led us to use the zero-shot model instead.

B Entailment and Decomposition

In line with our motivation, we would like to be able to use an NLI (Natural Language Inference) model to retrieve entailment scores of the produced summaries with respect to the input reviews. We tested several approaches including BERTScore, due to it being trained on entailment/contradiction pairs, but finally settled on using the zero-shot model from SummaC (Laban et al., 2022) to produce the entailment scores. SummaC is already becoming a standard evaluation tool for summarization factuality. We chose to forego the trained "Conv" SummaC model as we found that it did not generalize well to the kind of data we were working with. Specifically, two common issues were that (1) the range of scores assigned to the sentences from the reviews was very small, and (2) sometimes (especially for the most weakening statements) the scores assigned to the sentences seemed arbitrary and did not make a lot of sense. In comparison, the zero-shot model had neither of these issues. This issue is highlighted in Figure 6.

Further, a proposition X is typically not judged by models to entail statements of the form "*The reviewers said X*", or "*X and Y*", where *Y* is another proposition. Accordingly, the entailment scores are not very high for these two cases. We highlight this in Figure 7. Thus, we decide to split and rephrase all sentences of the produced summary to simple value propositions for all entailment-related metrics. Note that here rephrasing also includes removing any attribution such as "*The guests said...*".

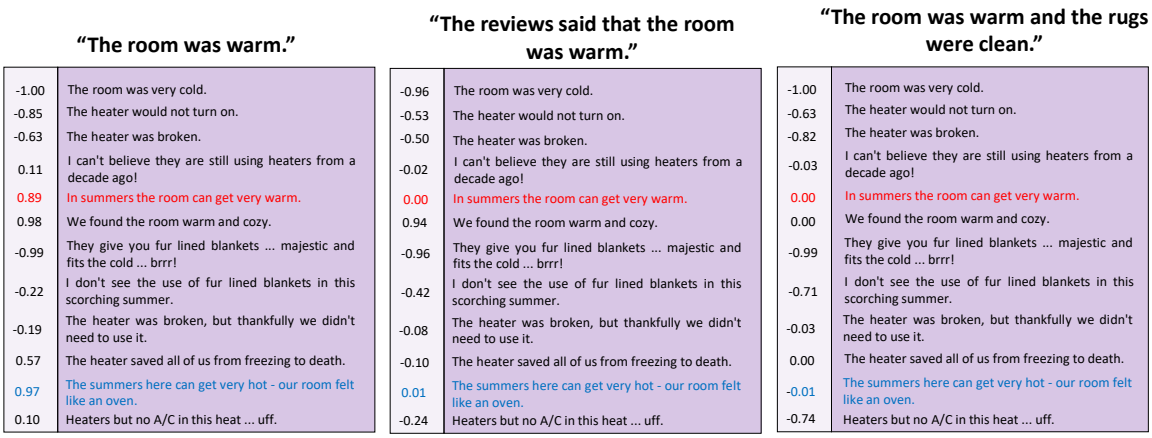

Figure 7: The scores of three statements with respect to a set of sentences, highlighting the issues with directly using the model output to compute entailment scores. Scores rounded to three decimal places are included before the corresponding sentences, with important lines highlighted in color. We note that quoting a proposition as said by someone else or having multiple propositions in the same sentence serve to cloud entailment scores.

We considered several models to this end, including BiSECT (Kim et al., 2021) and ABCD (Gao et al., 2021), but found two common issues with all of them:

- The split sentences maintained the words from the original sentences, so a sentence such as "*The food was received well but it was served late*" would have one output part as "*It was served late*", which requires a round of entity disambiguation to follow the split-and-rephrase step.
- These models do not remove attribution of viewpoints as we would like.
- A statement such as "*I liked the setting of the movie but not its cast*" produces one of the outputs as "*Not its cast*", which does not make any sense by itself.

Thus, we utilize GPT-3.5 to perform the split-and-rephrase task, with few shot prompting used to illustrate the removal of attribution and other desired characteristics. We also experimented with having separate steps for split-and-rephrase and found no significant difference in the outputs or quality thereof. We utilize the split-and-rephrased sentences for all of the automatic metrics that involve entailment of any sort.

C Measuring Complexity

One of the challenges of opinion summarization is that sentences may contrast opinions: "*Most reviewers liked the service, but there were a few complaints about sluggish response times.*" We quantify the percentage of simple and contrasting statements in the model outputs since it is subtly related to the extent of expression of opposing viewpoints. We use the original (non-split) sentences for this purpose and classify a sentence as contrasting if it contains one or more words from the set $\mathcal{K} =$ {'while', 'but', 'though', 'although', 'other', 'others', 'however'}, as Equation 3 depicts. We present these percentages in Table 11.

Pipeline	Complexity (%)	Pipeline	Complexity (%)	
SPACE		FewSum		
			(Amazon)	**(Yelp)**
Q	16.8			
A	5.1	Q	14.7	7.8
First-TCG	28.6	FS	16.8	12.3
TCG	30.7	G	36.1	31.9
QG	27.0	QG	34.6	32.8
TQG	27.3	First-CG	28.8	22.0
First-RG	24.0	CG	27.5	19.6
RG	30.7			

Table 11: Complexity as measured by the percentage of contrasting sentences.

$$C = \frac{\sum_{h \in \mathcal{H}, a \in \mathcal{A}} \sum_{s \in S_{h,a}} \mathbb{1}(N_1(s) \cap \mathcal{K} \neq \emptyset)}{\sum_{h \in \mathcal{H}, a \in \mathcal{A}} |S_{h,a}|} \quad (3)$$

We note that AceSum produces the smallest percentage of contrasting statements. We see that topic-wise clustering pushes up the number of contrasting statements for QG. We hypothesize that this is because when bringing together statements with the same topics in a cluster two opposing statements are likelier to fall into the same chunk. In

Pipeline	Percentage of novel n-grams		
	$n=3$	$n=4$	$n=5$
Q	4.3	5.3	6.3
A	30.1	61.7	79.1
First-TCG	71.9	87.4	92.8
TCG	78.3	93.1	97.5
QG	62.1	81.0	88.2
TQG	70.4	86.4	92.6
First-RG	71.6	87.2	92.6
RG	79.1	93.0	97.1

SPACE

Pipeline	Percentage of novel n-grams					
	Amazon			Yelp		
	$n=3$	$n=4$	$n=5$	$n=3$	$n=4$	$n=5$
Q	4.5	5.7	7.0	4.2	5.4	6.6
FS	89.2	96.4	99.0	90.7	97.5	99.3
G	93.1	97.5	98.8	94.4	97.9	99.4
QG	91.0	95.5	97.7	94.2	97.9	99.0
First-CG	91.8	96.3	98.1	92.9	96.6	97.9
CG	91.8	96.2	97.9	93.3	97.0	98.0

FewSum

Figure 8: Abstractiveness as measured by the percentage of novel n-grams when compared with the source reviews

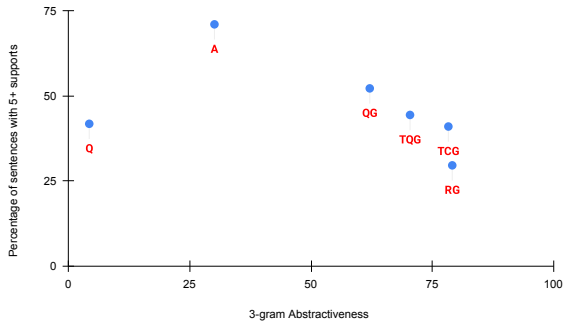

Figure 9: Average Top Score v/s Abstractiveness on the SPACE dataset.

cases where two opposing statements fall into different chunks, say X and Y, the chunks are likely to each contain statements similar to others in the same chunk. Thus, the summaries of those chunks are likely to be highly contrasting and thus increase the above measure even more for the final stage, as is observed above for TCG.

D Abstractiveness

We further investigate how the choice of the pipeline affects abstractiveness. To measure this, we calculate the percentage of n-grams in the summaries that do not appear in the input reviews for $n \in \{3, 4, 5\}$. For this, we use the original (non-split) sentences from the output summaries. The results are tabulated in Table 8.

Since QFSumm is a purely extractive model, it is no surprise that Q has low abstractiveness. The numbers are non-zero due to some quirks of QFSumm about splitting into sentences - this leads to some partial sentences ending up next to each other. The next stand-out is that A has very low abstractiveness. This is in line with our observation that even though AceSum is abstractive, it tends to highly generic observations such as "*The rooms were clean*", which very likely appear almost verbatim in some user reviews. We also observe that QG has a relatively low abstractiveness and that topic clustering drives up abstractiveness. We suspect that the above is a result of GPT-3.5 simply mashing together some sentences when presented with chunks containing highly disparate sentences (since it is hard to find a common thread among them), which promotes extraction over abstraction. Another observation is that multi-GPT-3.5 pipelines (TCG and RG) are more abstractive than single-GPT-3.5 ones since there are two rounds of abstraction as opposed to one. All the GPT-3.5-derived pipelines are highly abstractive in the case of FewSum, and slightly more so than FS. This is unsurprising since the combined length of the reviews in the case of FewSum is much smaller when compared to SPACE, and therefore there are relatively fewer propositions to compress into general statements. Motivated by Ladhak et al. (2022), we display the line graph of the average Top Score vs. 3-gram Abstractiveness for the SPACE dataset in Figure 9. The trio of QG, TQG, and TCG define the best frontier on the Factuality-Abstractiveness tradeoff, followed by RG, then A and Q.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
After section 6

☑ A2. Did you discuss any potential risks of your work?
In the Limitations section (after section 6)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section A.1

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section A.1

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section A.1

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section A.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☑ Did you run computational experiments?

Section 3 introduces the models being run, and Section 5 details the computed metrics.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section A.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section A.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sections 4.3, 5.2, and A.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section A.1

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Sections 4.3 and 5.2

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. The human evaluators were the authors themselves. The ratings were on Likert scales - the explanation of the scales has been included in section 4.3

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. The human evaluators were the authors themselves.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. The human evaluators were the authors themselves.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. The human evaluators were the authors themselves.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. The human evaluators were the authors themselves.