OPINESUM: Entailment-based self-training for abstractive opinion summarization

Annie Louis*
Google Research, UK
annielouis@google.com

Joshua Maynez*
Google Research, UK
joshuahm@google.com

Abstract

A typical product or place often has hundreds of reviews, and summarization of these texts is a challenging problem. Recent progress on abstractive summarization in domains such as news has been driven by supervised systems trained on hundreds of thousands of news articles paired with human-written summaries. However for opinion texts, such large scale datasets are rarely available. Unsupervised methods, self-training, and few-shot learning approaches bridge that gap. In this work, we present a novel self-training approach, OPINESUM for *abstractive opinion summarization*. The self-training summaries in this approach are built automatically using a novel application of textual entailment and capture the consensus of opinions across the various reviews for an item. This method can be used to obtain silver-standard summaries on a large scale and train both unsupervised and few-shot abstractive summarization systems. OPINESUM outperforms strong peer systems in both settings.

1 Introduction

Abstractive summarization is particularly promising for fluently comparing opinions from a set of reviews about a place or product. However, while language models trained on huge numbers of source-summary pairs have driven summarization performance in domains such as news, it is harder to find such pairs on the web for opinions, and immensely difficult to present tens or hundreds of reviews to human annotators and train them to write informative summaries. This paper presents a new self-training approach that automatically leverages *common opinions* across reviews, for example as in Table 1, to create powerful abstractive models.

So far, many abstractive summarization methods for opinions are based on auto-encoders (Chu and Liu, 2019; Bražinskas et al., 2020b; Isonuma et al., 2021), and do not use any supervision from gold human summaries. A few recent approaches propose self-training of encoder-decoder models on a task of predicting missing reviews from topically related ones (Amplayo and Lapata, 2020; Bražinskas et al., 2020a; Amplayo et al., 2021; Elsahar et al., 2021; Brazinskas et al., 2022). While this method is greatly useful for pretraining, inherently the objective predicts full review texts (which standardly contain a lot of non-summary worthy content as well), and has only weak signals for how to aggregate content across multiple reviews.

R1	...very large and clean with a nice size kitchen. The hotel is located right across the street from balboa park and within walking distance of a rite aid drugstore..
R2	...reserve a refurbished room and get that promise in writing! The location was great for tourists, right across from balboa park. You could walk to the zoo (about 1/4 mi)...
R3	...I decided to stay at the park manor suites hotel since it seemed to be close to san diego zoo. The hotel is conveniently located in front of balboa park, walking distance to san diego zoo,...
R4	...The staff are both pleasant and professional. Hotel is across from balboa park on sixth ave. This is the park west area, and features a diverse array of restaurants...
R5	...As other reviewers have said, it's easy to be here without a car — balboa park is just across the road and the airport is a short taxi ride away.

Table 1: An example consensus or common opinion between 5 reviews for a hotel on TripAdvisor.com, taken from the SPACE corpus (Angelidis et al., 2021)

We present an improved self-training method based on automatically generated silver-standard summaries. These silver summaries use *textual entailment* to locate the *consensus or most agreed upon user opinions* in the source reviews, and present them as self-training signals. For the example in Table 1, multiple mentions that "the location is right across balboa park", intuitively make this statement a worthy candidate for a summary of the hotel. In an improvement over prior self-training methods, our silver-summaries are *high quality and combine information across multiple reviews*. We show how to generate these silver summaries on a large scale to train encoder-decoder transformer models. We evaluate our models in unsupervised as well as few-shot learning, and show gains on both

* Both authors contributed equally to the work.

Findings of the Association for Computational Linguistics: ACL 2023, pages 10774–10790
July 9-14, 2023

content quality and fluency of output summaries.

2 Related work

Opinion summarization is a widely studied problem, where the role of sentiment, and product aspects (such as 'lens' and 'focus' for a camera) are well documented. We focus on abstractive methods for general-purpose summaries (not aspect-based), and we overview the closest approaches here.

Unsupervised neural networks. As in other areas of text generation, modern opinion summarization methods are also predominantly neural networks. Since large scale training data is largely absent in this domain, many prior methods are unsupervised. Common techniques include auto-encoders (Chu and Liu, 2019) and associated generative approaches such as VAEs (Bražinskas et al., 2020b; Isonuma et al., 2021). The underlying goal in these approaches is also to steer systems towards common information in the source reviews.

Our work also presents an unsupervised method, but based on encoder-decoder models.

Self-training methods. Some very recent solutions take advantage of recent large pretrained encoder-decoder models via self-training (Amplayo and Lapata, 2020; Bražinskas et al., 2020a; Amplayo et al., 2021; Elsahar et al., 2021; Brazinskas et al., 2022). These approaches create large number of pairs of source review sets, paired with a pseudo or silver summary as an approximate target. In all these methods, one of the reviews from the source set is taken as the pseudo summary, and other random reviews or topically related reviews to the target is taken as the set of source reviews. This dataset is then used for further training of encoder-decoder transformer models to suit the review domain. These self-trained models are usually better than unsupervised generative ones.

While allowing a favorable paradigm shift, and better performance, there are a few limitations of this type of self-training. As pointed out by Bražinskas et al. (2020a), reviews are considerably diverse from one another. So an objective that generates a review from other reviews will need to also predict content not present on the source side, a major difference from actual summaries of reviews. Such pseudo-summaries also contain a lot of first person language which is less desirable.

In this work, we present a novel self-training method. We also create silver-summaries on a large scale. However, our summaries actually contain *propositions* from *multiple* input reviews and in particular those which are reflective of the consensus among the review authors. These summaries are more powerful and faithful signals, and move the training task away from review generation.

Few-shot learning. With increased use of encoder-decoder models, methods have also been proposed to efficiently augment the training with a small number of human-generated summaries (50 to 100). Oved and Levy (2021) train transformer models on a small number of examples and use a ranking approach for inference. In Bražinskas et al. (2020a), a plug-in network is added to predict desired summary properties and thereby augment training signals. Brazinskas et al. (2022) introduce a few additional parameters in the form of adaptors and only these are finetuned instead of the full network, leading to efficient training on few examples.

We also demonstrate our self-trained model in few-shot settings.

Consensus as a goal for summarization. When there are multiple input texts, intuitively the common information across them is one important signal for summary-worthy content. Multi-document news summarization has exploited frequency from early times (Nenkova et al., 2006; Radev et al., 2004) to most recent ones (Ernst et al., 2022a). Consensus is also used as a goal for summarizing scientific publications around health topics (Shah et al., 2021), and identifying agreement and discrepancies in Wikipedia document clusters (Schuster et al., 2022). Instructions to annotators in multiple annotation efforts for opinion summarization explicitly ask annotators to capture what is common and popular (Bražinskas et al., 2020a; Angelidis et al., 2021). Most recently, agreement among opinions has been proposed for evaluating opinion summaries (Bhaskar et al., 2022).

Our self-training approach explicitly captures consensus statements. We acknowledge that majority opinions are only one indicator of useful content, and that others (eg. aspects) will complement it.

3 Textual entailment to identify consensus among review users

We propose a novel approach to create silver source-summary pairs for abstractive opinion summarization. A central idea here is the use of textual entailment to find statements reflecting user consen-

sus. We first present our definition of the idea and describe the steps involved in silver data creation.

3.1 Defining review consensus

We define consensus as the *number of reviews* that support a claim. For example, 60 (out of 100) reviews might claim that the seafood dishes are great at a restaurant. Likewise 40 reviews might say that the staff are friendly. We aim to identify such popular claims (high consensus) automatically.

But note that the same claim may be expressed in different ways or granularity, and so its frequency in reviews cannot be easily computed. Eg. *'This hotel is in the heart of Times Square'* and *'Hotel's location is slap bang in the middle of Times Square.'* are the same claim, and *'The fish is tasty'* and *'The salmon is delicious'*, both support the claim that *'The seafood is great.'*. Our idea is to accomodate this variability using natural language entailment.

At a high level, our approach identifies potential claims in the form of *propositions* from a large collection of texts, uses textual entailment to find out how often the collection supports the proposition, and computes a score for the support.

Now we explain how we obtain these statements and their scores automatically.

3.2 Extracting propositions

For texts, even when they are sentence-level units, it is hard to reason about them precisely. Many review sentences in addition tend to be rather long. For example, *"The room was very nice and clean, quiet location, staff were helpful, easy access to the centre of town by metro, bakeries and a supermarket nearby."* contain a bunch of different claims. It is difficult to find support for such complex sentences since the same information is unlikely to be present in other users' reviews.

Instead, we split review sentences into *propositions* and use these as our key units. We define a proposition as a 'single claim or fact' about the item and extract these as snippets from the original review texts. In fact, recent work on supervised news summarization uses the extraction and clustering of proposition units to find frequent subtopics, and then fuses the information in the biggest clusters into a summary (Ernst et al., 2022b).

Social media text is noisy with missing punctuation, and even syntactically, the long sentences are often multiple sentence-type units/claims concatenated into one, in contrast to embedded clauses which are typical of complex sentences in news etc. So existing approaches for proposition detection such as OpenIE (Stanovsky et al., 2018) which are trained on news and center on predicates, often miss detecting parts of long sentences on our domain. So, we use simple rules to split review sentences into propositions. Of course, there is room to improve proposition creation in future.

We split sentences at conjunctions, period, and comma subject to a minimum clause length of four. Our algorithm processes sentences from left to right to find a delimiter. If the proposed span will create a clause less than the minimum length, we do not split and attach the span to the proposition on the left. Note that these propositions are a linear segmentation of the input sentence, and their concatenation yields the original sentence. Intuitively, this process performs syntactic simplification, without changing the total content that is expressed.

The resulting propositions for different sentences in our data is shown in Table 2. Note that there are some propositions which end up ungrammatical, and our length constraints do not always separate out all the aspects (as in the third example). But overall this simple method works well for review sentences where syntactic embedding is less complex than in genres such as news, and we can scale to large collections efficiently.

We extract propositions from all the reviews for an item. Suppose there are N reviews for an item which result in M propositions, then $M \gg N$.

3.3 Scoring consensus

Our aim is to find the number of supporting reviews for each of the M propositions. We compute this number using natural language entailment. Specifically, consider review R_i and proposition m_j *belonging to the same item*. Let us represent a textual entailment relation as $P \rightarrow H$, where P is a premise and H is a hypothesis. In our case, if $R_i \rightarrow m_j$, then we consider that R_i *supports* m_j. The final score for proposition m_j, $S(m_j) = \sum_{1 \leq i \leq N} E(R_i, m_j)$ where $E(R_i, m_j) = 1$ if $R_i \rightarrow m_j$ else 0.

We obtain $E(R_i, m_j)$ using the predictions of an entailment classifier which treats R_i as the premise and m_j as the hypothesis. If the top label from the classifier is 'entailment', then $E(R_i, m_j) = 1$ and 0 if other labels had the highest probability.

In this work, we use a cross attention model, BERT-large (Devlin et al., 2019) to obtain these predictions. The input to the model concatenates

Review sentence	Extracted propositions
There was loads of cupboard space and a fantastic easy to use safe.	There was loads of cupboard space and$_1$ a fantastic easy to use safe.$_2$
Metro station (llcuna, line 4) is 5 minute walk away, beach is a 10 minute walk away.	Metro station (llcuna, line 4) is 5 minute walk away,$_1$ beach is a 10 minute walk away.$_2$
The room was very nice and clean, quiet location, staff were helpful, easy access to the centre of town by metro, bakeries and a supermarket nearby.	The room was very nice and clean, quiet location, staff were helpful,$_1$ easy access to the centre of town by metro, bakeries and a supermarket nearby.$_2$

Table 2: Example propositions split from source sentences. The propositions on the right are numbered according to their position in the sentence.

the premise and hypothesis with a separator symbol, and the CLS token's embedding is sent through a linear layer to predict three classes: entailment, contradiction and neutral. We trained this model on the MNLI corpus (Williams et al., 2018) reaching a development accuracy of 84%. Note that the training data for the entailment model does not contain any examples from the review domain. But we found that predictions are reasonable and even better when a higher threshold is applied on the probability of the entailment label.

Note that this score computation for all propositions requires an entailment prediction between all pairs of (R_i, m_j). Even though the computation is done only within each item, there are still a quadratic number of pairs per item.

So we implement the full computation of silver summaries in a Apache Beam[1] pipeline which allows to create parallel data-processing pipelines. Our typical pipelines do inference billions of times by the entailment models. It is also for this reason that we used BERT large since it was the largest cross-attention model that gave us the best tradeoff between accuracy and inference speed. For our silver data, the pipeline involved 1.3B entailment predictions taking 30hrs.

In Table 3, we show some of the entailment predictions from our models.

3.4 Silver summaries

We order the propositions in decreasing order of their scores $S(m_i)$, and take the top n as the silver summary sentences. We trim the silver summary up to a certain summary length expressed in tokens. Additionally, we employ a MMR (Goldstein and Carbonell, 1998) style redundancy removal technique to keep diverse content in the summary. We implement this control using a simple method of content word overlap.[2] Suppose S is the set of propositions selected in the summary so far. The next proposition chosen is the highest scoring proposition p_k where $overlap(p_k, s_i) < 2$, $\forall i, 1 \leq i \leq |S|$. $overlap$ is computed as the number of content words in common. We used the stopword list within NLTK (Bird et al., 2009).

The top propositions for two hotel items from our dataset is shown in Table 4.

This final set of summary propositions, S, chosen for a given summary length, are then concatenated in the chosen order to create the silver summary. When the propositions are not full sentences, we polish them for capitalization and punctuation to match natural texts. Some other disfluencies remain and several are related to the noisy nature of the text. We note that in most cases, the list of top propositions is a very reasonable summary, and in this first work, we have not carried out further processing for coherence.

3.5 Source texts

The silver summaries from the previous step are composed of extracted propositions from the source reviews. A system trained to produce such sequences from the full set of input reviews will predominantly copy from the input texts. So we make changes to the set of source reviews to make the data suitable for abstractive summarization.

Let N be the total set of input reviews. For each proposition p_i in the summary, we remove the *entire review* R_j, where p_i came from, i.e. p_i is a span in R_j. This deletion discourages the verbatim copying of text spans from the source. The final input reviews on the source side is a reduced set N', $|N'| < |N|$. Note that sentences (propositions) in the silver standard are supported by many other reviews, albeit in different surface forms, so the signals to produce the silver summary are still present

[1] https://beam.apache.org/

[2] We also explored entailment based diversity measures, but we found that simple content word overlap kept the maximum diversity in aspects commented on within the summaries. Entailment is a strict notion focusing on the truth of the hypothesis. The claims: (i) 'The hotel has a warm pool' and (ii) 'The hotel had a big pool' do not entail each other, and that works during consensus detection, but these claims on the same topic add redundancy in a summary. Word overlap was a more flexible way to focus on diversity. In the absence of data/models for redundancy detection, we used the best settings chosen empirically on dev. data.

Proposition: "the property has a lot of character" **Supporting reviews:** R1. ...Though i understand the previous posters point that the park manor has charm, I'd say that the actual "charm" happens in all the wrong places. That there's a nice and funky lobby with some amazing artistic featurettes and a cute patio with a coy boy, or the spacious rooms with a hodgepodge of furniture and beautiful molding on the walls that seems to go nowhere - yes, charming. R2. ...but the views higher would have been spectacular. A quirky place which people will love or hate... R3. ...this hotel is beautiful! It 's so elegantly decorted but in an antique way. The ceiling in the lobby... a huge king bed, sofa, armoire, vanity desk, kitchen - stove, refridgerator and the necessary kitchenware. I loved all the antique furniture, so nice to look at and change from standard hotel decor... R4. ...I would highly recommend this hotel to anyone who is looking for accommodations with more character than you'll find at the big chain hotels. A marriott looks like a marriott whether you're in singapore or st. Louis. Why not try the local flavor?... R5. ...This hotel is old and dated. The furnishings are very old and the whole hotel needs refurbishing . there are gas stoves in the rooms...
Proposition: "obvious neglect to fixtures and fittings." **Supporting reviews:** R1. ...i leant on the bannister at one point and almost fell down three floors...the window would not close... the electricity in our room kept cutting out if we had more than one item on... R2. ...my friends also got two leaks in their room... the carpets were old and they were obviously never hoovered in years...i saying they should knock the building down and do the whole thing up... R3. ...there were loose electric wires hanging from the ceilings-which i tripped over constantly...the locks on the doors were poor... R4. ...there are no elevators and the stairs are falling apart- literally!...broken window which was taped up with parcel tape and cardboard... broken heaters...wardrobe with door falling off... R5. ...could not charge phones because outlets did not work...cable tv was finnecky...internet was one computer on the second floor and did not work most of the time...broken fixtures and missing electrical covers...building seemed to be crumbling and it leaked in the foyer when it rained...

Table 3: Two example propositions (from two hotels in our dataset) with 5 reviews snippets which entail them. The reviews were randomly selected from the full list of reviews which entail each proposition. Our model does not explicitly do any sentiment classification: we have picked a positive and negative proposition for demonstrating how precise and clear our entailment based support prediction tends to be.

Hotel with 106 reviews	Hotel with 61 reviews
1. very comfortable (a big deal for me). (58%) 2. well maintained, clean, comfortable suites, (57%) 3. the rooms were very comfortable, and (55%) 4. they have a place to eat but (52%) 5. the size of the room is nice, (51%) 6. that was a great rate for a suite. (50%) 7. still professional; the room was clean and (50%)	1. well equipped with good privacy setting. (82%) 2. the family-owned vacation spot is very family oriented. (68%) 3. this resort is a comfortable family retreat providing a great getaway. (60%) 4. a very family friendly place to stay. (60%) 5. our unit was very clean, comfortable.. (55%) 6. units have had great proximity to a pool and (54%)

Table 4: The top propositions for two hotels in our dataset. We take the top 10 propositions and show only the ones kept after redundancy filtering. The percentage of total reviews which entail each proposition is within braces.

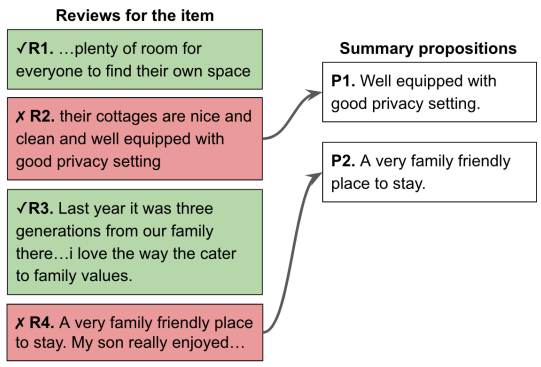

Figure 1: Example which demonstrates how reviews are removed from the summarization input side if they were the original source from which a proposition was extracted. Here, P1 was extracted from R2 and P2 from R4. R2 and R4 will be removed entirely from the summarization input. But note that the summary content is still present in other reviews which entail P1 and P2.

in N'. An illustration of input review selections is shown in Figure 1. This way of creating source-summary pairs resembles one of the powerful pre-training objectives for abstractive summarization known as Gap Sentences Generation, introduced by the Pegasus (Zhang et al., 2020) model.

In practice, the number of source reviews that can be processed as input to most standard sequence to sequence models is fewer than the hundreds present in N'. So we sample a smaller set N'', size k, of reviews to fit the sequence length of the encoder. We could further aid the training by adapting N'' to be most useful for generating the target sentences. We sample l reviews from the entailment set of each summary proposition, where l is proportional to the size of the entailment set. For example, if 'seafood is great' is a summary proposition with 40% entailment support, then 40% of the summarization input are reviews on the topic of great seafood, although the review containing the verbatim proposition is filtered out.[3] In this way, the source *always* contains entailing reviews for every summary proposition. So our silver data explicitly guards against hallucinations.

In the next sections, we describe how we use this data to train abstractive summarization systems.

[3]We tried other sampling methods, but they performed similarly during development.

4 Datasets

We use two types of data in our experiments: *unlabelled* and *eval*. The first is an unlabelled review corpus (no gold or human summaries are available for the items). This dataset is used to create silver-standard summaries for self-training. The second type is an evaluation dataset containing a much smaller set of items (not seen during training) and here for each item, the set of source reviews is paired with one or more human summaries.

We use publicly available anonymized corpora of reviews and summaries as per their terms.

SPACE-unlabelled. The SPACE corpus collected by Angelidis et al. (2021), comprises of reviews for hotels from the TripAdvisor website. There are a total of 1.1 million reviews for 11,000 hotels. These reviews are not paired with human summaries. We use this dataset for silver-summary creation.

We use two sources of gold-standard abstractive summaries for evaluating our systems.

SPACE-eval. The SPACE corpus (Angelidis et al., 2021) also contains gold summaries for a small set of 50 hotels (25 dev and 25 test). For each of these hotels, 100 input reviews are paired with 3 human-written abstractive summaries. The human-summaries were created via a two-step process where annotators first selected key sentences from the input reviews, and then wrote a general summary based on the sentence collection.

This evaluation dataset ideally suits our task since the input contains 100 reviews on which one could ask for common opinions and claims.

AMAZON-eval. This dataset (Bražinskas et al., 2020a) contains summaries for reviews of Amazon products. There are 60 products (13 dev, 20 test and 28 train). The training items are only used for few-shot learning experiments. For each product, there are only 8 (randomly chosen) input reviews on the source side, and with 3 human-written abstractive summaries. These inputs comprising 8 random reviews offer little scope for common themes across them. Previous few-shot learning studies use this corpus, so we include it for comparison while acknowledging that the inputs are less than ideal.

5 Experiments

In this section, we explain how we trained our abstractive summarization models.

5.1 Models

We build our abstractive systems using pretrained encoder-decoder models based on T5's (Raffel et al., 2020) framework. These models encode the input reviews as a sequence and autoregressively generate the output summary words as a sequence.

In multi-document summarization, especially opinions, source reviews are easily in the hundreds. Standard self-attention layers found in current transformer models have a polynomial scale relationship to input length, making it impossible to encode and attend to several reviews at once. Many summarization systems avoid this issue by including a content selection component as a first step of a pipeline. Recent work has shown that sparse transformers can overcome this issue, simplifying models and often outperforming pipeline based alternatives. For this reason, we have built models on top of LongT5 (Guo et al., 2022), which implements sparse attention and allows tokens to attend locally and globally via transient global nodes.

We employ LongT5 models (size: Large with 770M parameters) with a limit of 8,192 sentence pieces. We use the public pretrained checkpoint. [4] Larger sizes XL (3B parameters) did not improve performance. We also compared how many reviews, size k, should be present on the input side. Typically more reviews, 160 (filling the model's sequence length), performed the best.

During development, LongT5 models always outperformed their T5 counterparts.

5.2 Silver Data

We create our silver data using the SPACE-unlabelled corpus (Section 4) and follow the procedure outlined in Section 3.

We retained 4,729 items with a minimum of 50 reviews (since very few reviews may not have a lot in common to extract out). Our beam pipelines computed around 1.3B entailment predictions on the review-proposition pairs from these items. The resulting silver data has the same number of items, but now each item is paired with a silver summary.

5.3 Self-training

We explore the usefulness of our self-training in two setups: unsupervised and few-shot learning. For the unsupervised case, we train our models on the silver-data only. For few-shot learning, we use

[4] https://github.com/google-research/longt5

a small number of annotated input-summary pairs (<100) for finetuning our self-supervised systems.

Unsupervised training. Given the silver-data, we trained LongT5-Large models on the sequence-to-sequence task of generating the highest consensus opinions given a concatenated sequence of the input reviews. These models do not use any gold-annotated examples for training. We select the best checkpoint based the ROUGE scores (mean of R2 and RL) on the validation set. We compare these systems with prior unsupervised work.

Few-shot Learning. was implemented by finetuning our self-trained models (i.e. trained by unsupervised method above) on a few human annotated source-review and summary pairs. To facilitate this setup, we divide the development examples in SPACE-eval (25 total) into a training set with 15 items and a validation set with 10 items. In our preliminary experiments, we did not find much performance difference in how we chose these sets. The test set remains unchanged. The AMAZON-eval data is already divided into 3 sets (Bražinskas et al., 2020a). The best checkpoint was again selected based on ROUGE scores on the validation set. These models trained better with a reduced learning rate, $1/5th$ of the standard $1e-4$.

In addition to prior few-shot systems, we will also compare these models with transformer baselines which do not use self-training with silver summaries. Rather these baselines are warm started from the public pretrained checkpoints and then similarly finetuned on the (few-shot) train splits.

6 Results

We present our findings on the SPACE-eval and AMAZON-eval testsets (Section 4). We compute the automatic ROUGE (Lin, 2004) metric, and also confirm the improvements with human raters. For comparison, we included recent systems where the best system outputs were available publicly.

6.1 Unsupervised models

We compare OPINESUM with previous abstractive systems: **Meansum** (Chu and Liu, 2019), **ACESUM** (Amplayo et al., 2021) (current best for SPACE), and **Copycat** (Bražinskas et al., 2020b) (competitive on AMAZON).

Often extractive systems are strong baselines, so we also include: **Centroid**: a review closest to the mean review vector computed using BERT (Devlin et al., 2019); **Lexrank** (Erkan and Radev, 2004), **Quantized Transformer** (Angelidis et al., 2021), **Random** (review), **Lead:** first sentences of reviews, and **Clustroid:** review with highest ROUGE-L scores with other reviews.

Model	R1	R2	RL
Vanilla LongT5 model			
LongT5L	27.22	4.81	15.33
Previous extractive systems			
Centroid	33.20	7.01	17.97
Lexrank	33.08	6.37	19.46
QT	38.68	11.51	22.45
Previous abstractive systems			
Meansum	36.16	9.16	21.38
Copycat	38.52	11.17	23.41
Acesum	42.57	14.50	25.05
OPINESUM abstractive system			
LongT5L	**45.84**	**16.30**	**29.18**

Table 5: Unsupervised test results on SPACE-eval. Here OPINESUM uses silver-summaries only.

To separate out model contributions from those of silver data, we also include a **vanilla longT5** system (no silver data). LongT5 was pretrained with a summarization objective (Zhang et al., 2020) and hence already suited to the problem.

These results are in Tables 5 and 6.[5] We see that OPINESUM obtains very good performance, achieving the best results on SPACE-eval. Note that these improvements are reached with only a new self-training method and without any additional learning or domain-specific changes which are present in most of peer systems. The improvement is further evidenced from the fact that a vanilla longT5 model by itself is only a weak baseline.

On AMAZON-eval, we are close to copycat but do not outperform it. As we outlined already in Section 4, our self-training aims to capture consensus among input reviews and on this data, the input comprises 8 random reviews with little in common. Hence the self-trained models are less successful at predicting the summary content chosen by those human annotators (especially in an unsupervised manner). Still they are close to one of the best systems and outperform other peers.

[5]We noticed that ROUGE results from previous papers are not always consistently reproducible, since they either choose to compare peer summaries with a union of multiple references or take the maximum ROUGE from individual reference comparisons. So we computed or obtained output summaries from previous authors and calculated ROUGE scores in a consistent manner (max from 3 comparisons) and report these results. Note hence that these results may differ from that recorded in previous papers, and we only report numbers where we could obtain summaries from authors of previous papers. But our numbers can be systematically compared.

Model	R1	R2	RL
Vanilla LongT5 model			
LongT5L	27.06	5.49	17.99
Previous extractive systems			
Random	27.35	5.43	17.93
Lead	28.84	7.32	17.13
Clustroid	29.11	5.80	18.89
Lexrank	30.50	7.32	19.38
Previous abstractive systems			
Meansum	29.36	7.63	19.41
Copycat	**34.22**	8.84	**23.85**
OPINESUM abstractive system			
LongT5L	30.85	**10.70**	20.86

Table 6: Unsupervised test results on AMAZON-eval. Here OPINESUM uses silver-summaries only.

6.2 Few-shot learning models

These results are in Tables 7 and 8. There are no prior few-shot results on SPACE-eval. Nevertheless, T5 models trained without silver-data but with few-shot learning are a strong ablation to compare with OPINESUM. We list these under *vanilla few-shot* models. On AMAZON-eval, we compare with Fewsum (Bražinskas et al., 2020a) and Adasum (Brazinskas et al., 2022) (See Sec. 2 for details).

Here, the baseline T5 models are already rather strong, especially longT5. Our simple self-training still leads to further improvements on SPACE-eval.

On AMAZON-eval, the LongT5 systems are better than Fewsum but not as good as the current best Adasum which is tailored for efficient few-shot training. As already alluded to, it is also likely our methods have less to leverage across only 8 input reviews. Nevertheless, note that our contribution demonstrates a self-training method alone which can complement other training strategies.

We show an example output of our system compared with gold standards and prior system in the Appendix (Table 10). There is a noteworthy difference between our unsupervised and fewshot systems. The unsupervised system produces shorter summaries, is less adapted to the domain, and at times, contains disfluencies due to being trained on smoothed propositions. Fewshot learning improves along these dimensions making the summary much closer to the gold standards.

6.3 Human evaluation

Our ROUGE results on SPACE-eval show that given sufficient number of input reviews, our self-training delivers improved summarization performance. We confirm these improvements via a human evaluation, and also examine other dimensions such as summary coherence.

Model	R1	R2	RL
Vanilla fewshot models			
T5 L	42.67	13.03	28.43
LongT5 L	45.51	13.03	29.28
Opinesum fewshot model			
LongT5 L	**47.19**	**14.60**	**30.13**

Table 7: Results for the few-shot learning setting on the SPACE-eval dataset. All the models were finetuned on a small set of 15 training examples described in Section 5.3. Repeat sentences were removed from prediction via MMR. 'Vanilla' systems are warm started from public checkpoints and do not see self-training data.

Model	R1	R2	RL
Previous fewshot systems			
Fewsum	37.55	10.51	25.21
Adasum	**44.14**	**15.61**	**29.13**
Vanilla fewshot models			
T5 L	29.74	10.28	21.96
LongT5 L	38.65	12.31	27.28
OpineSum fewshot model			
LongT5 L	39.56	11.47	25.77

Table 8: Results for the few-shot learning setting on the AMAZON-eval dataset. All the models were finetuned on a small set of 28 training examples described in Section 4. Repeat sentences were removed from prediction via MMR. 'Vanilla' systems are warm started from public checkpoints and do not see self-training data.

We conduct this evaluation on the 25 test examples in SPACE-eval using crowd annotators. They were presented with the three gold summaries for each example and four system summaries. They were asked to pick the best summary (no ties) for two criteria: (i) *content*: where the content of the system summary matches best with the gold summaries and (ii) *coherence*: the most well-written system summary. Each example was rated by 3 annotators. More details are in the Appendix.

We also conducted two rounds of annotations. In the first *Eval A*, annotators compared summaries from four unsupervised abstractive systems. None of these systems were trained with gold summaries. In the second *Eval B*, we seek to understand the improvements from few-shot learning where even with few examples, the system is given the opportunity to learn the style of a domain. Table 9 indicates how often each system was picked as best by the raters (out of 75 ratings per question).

The OPINESUM outputs are definitely noticed as higher content quality compared to baselines and on par with the best systems. There a greater edge to these systems on the coherence criteria.

Eval A: Unsupervised systems

	Copycat	Meansum	Acesum	OPINESUM
Content	0.17	0.21	**0.31**	**0.31**
Coherence	0.20	0.08	0.24	**0.48**

Eval B: Impact of few-shot learning

	Unsupervised		Fewshot	
	Acesum	OPINESUM	LongT5 L	OPINESUM
Content	0.08	0.12	**0.40**	**0.40**
Coherence	0	0.28	0.35	**0.37**

Table 9: Human evaluation results on SPACE-eval showing the proportion of times an annotator picked a summary as the best. In *Eval A*, annotators compared unsupervised systems. In *Eval B* they were shown a mix of few-shot and unsupervised systems.

6.4 Faithfulness of summaries

For abstractive systems, faithfulness is also an important dimension of quality. Following standard practice (Maynez et al., 2020; Dušek and Kasner, 2020; Honovich et al., 2022), we use an entailment classifier to ascertain whether the content of OPINESUM's summary sentences originate from the source reviews.

For a score, we compute the percentage of faithful summary sentences from OPINESUM, across all inputs (i.e. macro-average). A summary sentence S_j is faithful if at least one input review R_i entails S_j. We use the entailment classifier from Honovich et al. (2022)[6] which comprises a T5-11B model (Raffel et al., 2020) trained on the ANLI dataset (Nie et al., 2020). We set the threshold for entailment label at 0.7 for greater precision on the out-of-domain review examples.

We do this evaluation on the SPACE-eval dataset where we found the entailment predictions to be more reliable. Here we find that 98.6% of all sentences from an *unsupervised* OPINESUM model are faithful to the source. As described in Section 3.5, our models are explicitly trained to produce sentences which are entailed by the source reviews. This aspect has likely led to the high performance.

We do not report these scores for other systems as they produce longer sentences, and hence have an outright disadvantage when entailment is computed for their sentence-level units. For OPINESUM models, the sentences are shorter and correspond to propositions and it is easier to check accurately if they are entailed by the source.

While not comparing systems, we find that OPINESUM summary faithfulness is desirably high.

[6]https://github.com/google-research/true

6.5 Errors analysis

To further understand these performance numbers, we present sample outputs in Tables 10 and 11.

Table 11 presents *unsupervised* summaries for two inputs from the SPACE-eval test set. For both these inputs, all the three human annotators in our evaluation (see Section 6.3) rated OPINESUM summaries as the best, both in terms of content quality and coherence. The text of these summaries provides some insights into properties which might correlate with the quality judgements.

A distinguishing property of *unsupervised* OPINESUM outputs is that the text is composed of small sentences corresponding to propositions. Each of these relates to a single aspect. On the other hand, other unsupervised systems produce longer sentences. We observe that in many such long sentences there is considerable incoherence among the parts e.g. from ACESUM: *there is a lot to do if you are looking for a good place to stay, but the location is very close to the beach and the beach is just a block away from the sand beach*. This difference could lead to the perceived higher content quality and coherence of OPINESUM texts. During few-shot learning, OPINESUM summaries adjust to longer length sentences as in Table 10.

We also note that redundancy is a quality issue across the summaries including OPINESUM, and this is also an avenue to improve for future work.

7 Conclusion

We have presented a simple self-training approach which leads to gains on both unsupervised and few-shot abstractive opinion summarization.

Our work is one of the first to demonstrate how an intuitive idea of review consensus can be incorporated during self-training. It opens up a number of challenges and new problems for future work. In particular, while our silver data contains the *provenance* for each top proposition—meaning the set of reviews which support each the proposition—this information is only minimally used at the moment. Future work could explore how the entailment weights (scores) of each proposition and their links to entailing reviews could be used for further improvements and faithful generation.

We also hope that such self-training models could serve as good checkpoints for other tasks in the opinion domain such as review helpfulness or product popularity prediction.

Limitations

In this first effort, OPINESUM was demonstrated for the English language. Since the wealth of review information on many websites span several languages, extending our work to other languages is a key area for future work. There are two language specific components—the proposition identification rules, and the textual entailment model. For the latter, there are multilingual resources such as mT5 models (Xue et al., 2021) and multilingual entailment datasets (Conneau et al., 2018) which are good starting points. The proposition rules are much more language specific. Very recent work has introduced a corpus and learned model for proposition identification (Chen et al., 2022), and future research in languages other than English could strength this component.

A second noteworthy point is the scalability of the silver data creation. As described in Section 3.3, we perform a quadratic number of entailment queries per item. In this work, this was of the order of a few billion. We used an Apache beam pipeline to scale our computation using a lot of parallel computation on CPUs. Readers must be aware of this computation when applying such an approach for their work. However, note that the processing only needs to be performed once for training data creation. Future work on more efficient transformer models such as Khattab and Zaharia (2020), will help vastly improve these types of computations.

Ethics Statement

This paper focuses on abstractive summarization where text is generated using an encoder-decoder model. Hence typical issues associated with text generation must be kept in mind while assessing the outputs from such models. As language models improve in faithfulness and accuracy of generated texts, we expect our systems to benefit similarly.

Acknowledgements

We would like to thank Ryan McDonald, Filip Radlinski, and Andrey Petrov for their suggestions and feedback on our work.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive Opinion Summarization in Quantized Transformer Spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Adithya Bhaskar, Alexander R Fabbri, and Greg Durrett. 2022. Zero-shot opinion summarization with gpt-3. *arXiv preprint arXiv:2211.15914*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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*, pages 4119–4135.

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.

Arthur Brazinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. Efficient few-shot fine-tuning for opinion summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1509–1523, Seattle, United States.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2022. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. *arXiv preprint arXiv:2212.10750*.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1223–1232. PMLR.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

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.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137.

Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. 2021. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1646–1662.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022a. Proposition-level clustering for multi-document summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1765–1779.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022b. Proposition-level clustering for multi-document summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jade Goldstein and Jaime Carbonell. 1998. Summarization: (1) using MMR for diversity- based reranking and (2) evaluating summaries. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 181–195.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161—-175.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901.

Nadav Oved and Ran Levy. 2021. PASS: Perturb-and-select summarizer for product reviews. 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 351–365.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. Stretching sentence-pair nli models to reason over long documents and clusters. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Findings*.

Darsh Shah, Lili Yu, Tao Lei, and Regina Barzilay. 2021. Nutri-bullets hybrid: Consensual multi-document summarization. In *Proceedings of the*

2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5213–5222.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

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.

A Example system output

Table 10 shows the outputs from our best unsupervised and few-shot learning systems compared with gold standards and best prior approach.

For error analysis, in Table 11, we present various unsupervised summaries for two inputs from the SPACE-eval test set. For both these inputs, all the three human annotators in our evaluation (see Section 6.3) rated OPINESUM summaries as the best, both for content quality and coherence.

B Human evaluation

We employed three crowd annotators for our manual evaluation. The instructions provided to them are in Figure 2.

Task: Mark the quality of a summary of user reviews.

Below you will find summaries produced for the reviews of a product or place.

At the top, we show three expert summaries. These 3 experts were asked to read all the reviews for a place or product, and they each wrote a summary of the information. Consider these as good quality summaries.

After that we show 4 machine-generated summaries called A, B, C, and D. We would like you to tell us which machine-generated summary matches the information mentioned by experts, and which of these summaries are well-written.

Name of the product or place: Hotel 1

Expert Summaries

<gold summary 1>	<gold summary 2>	<gold summary 3>

Summary A

Summary B

Summary C

Summary D

Q1: Which of A, B, C and D matches the content of the expert summaries the most? Be sure to penalize or rank lower when a summary contradicts what experts say. If there is a tie for the top place, please select a random one.

- ◯ Summary A
- ◯ Summary B
- ◯ Summary C
- ◯ Summary D

Q2: For this question, you need not compare to the expert summaries. Which of the summaries A, B, C and D is more coherent and well-written? Please select a random one if there is a tie for top place.

- ◯ Summary A
- ◯ Summary B
- ◯ Summary C
- ◯ Summary D

Figure 2: Template used for human evaluation with the questions rated by annotators.

Gold standard summaries G1. This hotel was very nice and within walking distance of the Vatican, Colosseum, Forum, ST Peters, etc. Staff were helpful in every way, and the attention to each request and question was efficient and treated with courtesy. The air-conditioned rooms were very nice, clean, and comfortable, with immaculate bathrooms to boot. Breakfast, which is included, was pretty good for a continental buffet. G2. Staff received mixed reviews, but were overall considered friendly, attentive, and helpful. The hotel, rooms, and bathrooms were very clean, with daily maid service and linen change. The room was beautiful and airy. The Breakfast was great and varied. The location is excellent, away from the hordes of tourists. It's just a short walk over Ponte Umberto to Piazza Navona, or across Ponte Cavour to reach the popular shopping areas. The building is nice. The restaurant was first rate. However, some thought that the hotel is pricey for the quality of the room. G3. The staff was extremely courteous and helpful. The wooden floors were all cleaned and maintained; as well as everything else in the hotel. The rooms were beautiful and large, and the bathroom was immaculate. There was a good, buffet style breakfast with particularly enjoyable cold meats, and with anything else desired. The hotel is located close enough to the Vatican, Colosseum, the Forum, and St. Peters- overall a great location.
ACESUM (Amplayo et al., 2021) The staff were very friendly and helpful. the room was clean and clean. it was a great place to stay. if you want to stay in cicerone, it is a good place to get to the shopping area. there are many restaurants, restaurants and restaurants. but the staff are very friendly, friendly and friendly. they are a very nice hotel, a nice place to eat, and a lot of good food, as well as a small restaurant, the breakfast was very good, but a bit of.
OPINESUM**-unsupervised** The hotel is located within walking distance of the Vatican. The rooms were clean and comfortable. The Cicerone is a nice hotel. As far as the hotel goes. The reception area is nice but the rooms. The breakfast buffet was fine. The room was a good size.
OPINESUM**-fewshot** The staff was friendly and helpful. The rooms and hotel itself is modern, extremely clean! The rooms are a good size, with comfy beds, a breadth of amenities such as a great shower and a comfortable bed. The breakfast buffet is average, but very good, with lots of variety. The location is very central. The hotel is within walking distance of the Vatican and Piazza Navona. The Cicerone is a beautiful hotel, but the hallways need refurbishing.

Table 10: Example summaries for one item in our SPACE-eval dataset. We show the 3 gold standard summaries available for the example, along with previous best system (ACESUM), and our unsupervised and few-shot self-trained systems.

Unsupervised system summaries	
Input A	**Input B**
AceSum the staff were very helpful and helpful. the room was clean, and the location was excellent. a few minutes walk from the beach. it was a great place to walk to the street/food truck fair, venice beach and a short walk from abbot kinney. there is a lot to do if you are looking for a good place to stay, but the location is very close to the beach and the beach is just a block away from the sand beach, the hotel was very nice. and the staff was very friendly.	**AceSum** the staff were very friendly and helpful. the room was clean and clean, the breakfast was good, and the location was excellent. it was very close to st. mark's, a few minutes walk from san marco. if you are a big walker, it is a great place to eat the food, but it's not a bad thing. there was a lot to do with a good breakfast, as well as a nice breakfast's and the staff was very friendly.
MeanSum It was a great stay! The food at the hotel is great for the price. I can't believe the noise from the street is very loud and the traffic is not so great, but that is not a problem. The restaurant was great and the food is excellent.	**MeanSum** This was a great find in the central location. One of the best hotels I've ever stayed at, but that was my second stay. We have been to Venice many times and this was by far the best hotel we have stayed at. The location is great, close to the train station and the ferry to the Vatican. Lovely weather, and a good night sleep. We were there for a week and had a blast. Would highly recommend this hotel.
Copycat This hotel is in a great location, just off the beach. The staff was very friendly and helpful. We had a room with a view of the beach and ocean. The only problem was that our room was on the 4th floor with a view of the ocean. If you are looking for a nice place to sleep then this is the place for you. If you are looking for a good place to stay in Venice, this is the place to stay.	**Copycat** This hotel is in a great location, just off the Grand canal and within easy walking distance of all the main attractions. The staff were very friendly and helpful. Our room was small but very clean and the bed was very comfortable. I would recommend this hotel to anyone who is looking for a good place to stay in Venice.
OpineSum The view from the rooftop was awesome. The staff was very nice and helpful. The location of the hotel is perfect. The rooms were clean and comfortable. High, the hotel's rooftop bar. We had a view of the beach. The room was clean and well-appointed. The location is great- right by the beach.	**OpineSum** The staff were very helpful and polite. The breakfast was good and the staff friendly. The rooms were clean and well maintained. The hotel is in a great location close to everything. Giorgione is a lovely hotel in a quiet area of Venice. The room was a good size. We had a standard room that was clean. The rooms are not very large.

Table 11: Two examples from SPACE-eval test set where *all 3 human annotators* chose OPINESUM outputs as the best with regard to content as well as coherence.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☑ A2. Did you discuss any potential risks of your work?
9

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

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

B ☑ Did you use or create scientific artifacts?

We use publicly available datasets for evaluation. Section 4

☑ B1. Did you cite the creators of artifacts you used?
4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
4

☑ 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)?
4

☑ 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?
4

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4

☑ 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.
4

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5 and 3.3

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?
5

☑ 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?
5.3

☑ 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.)?
Yes, Section 5 and 6, especially see footnote in Section 6.1

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
6.3

☑ 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.?
Section 6.3 and Appendix B

☑ 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)?
We recruited 3 crowd annotators on a proprietary platform discussed in Section 6.3.
However, due to privacy concerns, we did not include the estimated hourly wage paid to participants or the total amount spent on participant compensation. We feel that individuals' hourly wage or compensation is personal information and we cannot disclose this under privacy law. However, this work was carried out by paid contractors, and we can confirm that they received their standard contracted wage, which is above the living wage in their country of employment.

☒ 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 discussed in the paper. But annotators were told the task was evaluation of system summaries and instructions were provided to them before data collection.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We use standard evaluation questions for summary outputs on public data, so we did not need an approval process.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We used 3 English-speaking annotators but do not collect any demographic data.