AgreeSum: Agreement-Oriented Multi-Document Summarization

Richard Yuanzhe Pang[1*§] **Adam D. Lelkes**[2*] **Vinh Q. Tran**[2*] **Cong Yu**[2]
[1]New York University, New York, NY 10011, USA
[2]Google Research, New York, NY 10011, USA
yzpang@nyu.edu, {lelkes,vqtran,congyu}@google.com

Abstract

We aim to renew interest in a particular multi-document summarization (MDS) task which we call AgreeSum: agreement-oriented multi-document summarization. Given a cluster of articles, the goal is to provide abstractive summaries that represent information common and faithful to all input articles. Given the lack of existing datasets, we create a dataset for AgreeSum, and provide annotations on article-summary entailment relations for a subset of the clusters in the dataset. We aim to create strong baselines for the task by applying the top-performing pretrained single-document summarization model PEGASUS onto AgreeSum, leveraging both annotated clusters by supervised losses, and unannotated clusters by T5-based entailment-related and language-related losses. Compared to other baselines, both automatic evaluation and human evaluation show better article-summary and cluster-summary entailment in generated summaries. On a separate note, we hope that our article-summary entailment annotations contribute to the community's effort in improving abstractive summarization faithfulness.

1 Introduction

Recent works have made great progress in single-document summarization (SDS) thanks to the encoder-decoder framework and pretraining procedures (Cho et al., 2014; Rush et al., 2015; Narayan et al., 2018; Zhang et al., 2020a). There is a growing interest in multi-document summarization (MDS; Zopf, 2018; Fabbri et al., 2019; Liu and Lapata, 2019; Chu and Liu, 2019; Wang et al., 2020b, et seq.), with applications in search engines, news clustering, timeline generation, and other areas. Past MDS research has primarily focused on summarizing articles such that the summary covers an event "comprehensively" while "avoiding redundancy" (Fabbri et al., 2019). We can say that most existing MDS tasks summarize the "union" of the articles.

In this paper, we discuss *agreement-oriented multi-document summarization* (AgreeSum), in which we aim to *abstractively* summarize the "intersection" of the articles. More specifically, the input to the task is a cluster of articles, and the expected output is a summary that represents information common and faithful to all input articles in the cluster (Section 3). A few works (discussed in Section 2) have investigated the problem, without using modern neural-network-based methods. The motivation for reviving interest in AgreeSum is twofold. First, given that certain microscopic details are not likely present in all articles in a given cluster, they would be filtered out through AgreeSum. If the source articles reflect different points of view, AgreeSum provides a way of capturing the common ground.

The second motivation for AgreeSum lies in the pursuit of summarization faithfulness. AgreeSum is timely, given that recent works have shown difficulty of producing faithful abstractive summaries (Falke et al., 2019; Maynez et al., 2020; Kryscinski et al., 2020; Durmus et al., 2020; Zhou et al., 2021), though in the SDS setting. AgreeSum could allow practitioners of abstractive summarization systems to carefully study topics related to faithfulness and hallucination.

Given the scarcity of readily available data, we create a dataset[1] based on English Wikipedia current events portal (WCEP)[2]. WCEP contains neutral human-written news summaries with links

[§] Work completed during internship at Google Research.
[*] Equal contribution.

[1] https://github.com/google-research-datasets/AgreeSum
[2] https://en.wikipedia.org/wiki/Portal:Current_events

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3377–3391
August 1–6, 2021.

to news articles (with usually one link per summary). We first extract human-written summaries on WCEP, the linked articles, and semantically similar articles to the linked articles, in order to obtain article clusters associated with summaries. We then annotate the entailment relationship for a subset of the cluster-summary pairs (i.e., whether or not a given article from the cluster semantically entails the summary). From there, we build cluster-summary pairs for AgreeSum (Section 3.2).

We build upon previous SDS research, using the top-performing pretrained PEGASUS model (Zhang et al., 2020a) as the starting point for our models. Using our dataset, we first examine a few baseline models and show that several pretraining-based summarization models fail at generating summaries that satisfy AgreeSum's requirements. We also propose an approach that integrates both supervised (using the annotated portion of the dataset) and unsupervised losses (namely, an entailment loss and a language loss to be discussed later, using both the annotated portion and the unannotated portion) while leveraging PEGASUS. We show the effectiveness of a simple policy gradient-based algorithm (Sutton et al., 2000) in which the rewards are based on a T5-based article-summary entailment model (Raffel et al., 2020). To summarize the contributions:

- We introduce the AgreeSum task and the WCEP-based dataset (§3.2).
- A subset of *article*-summary pairs are annotated with entailment information (§4). On a separate note, the article-summary-level recognizing textual entailment (RTE)[3] task could stand as a challenging task on its own. The annotations could be of interest to the research in improving abstract summarization faithfulness, in the context of not only AgreeSum but also general single-document summarization tasks.
- We develop simple ways of applying PEGASUS to AgreeSum. We provide a few baselines as well as a model that uses unsupervised entailment-related and language-related losses to complement the supervised finetuning of PEGASUS (§5). Both automatic and human evaluations are performed (§6). In particular, we show that the T5-based entailment model can be integrated into summarization models to produce summaries that are entailed in source articles.

[3]The RTE datasets (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007) correspond to the following two-class classification task: given a premise and a hypothesis, a model would decide if the premise entails the hypothesis.

2 Related Work

Traditional MDS. DUC (Paul and James, 2004; Dang, 2005) and TAC (Owczarzak and Dang, 2011) MDS data are among the first high-quality relevant datasets. These datasets are human-curated, but tiny in terms of the number of examples. Recent works have explored creative methods for obtaining low-cost MDS datasets. Liu et al. (2018) use Wikipedia articles as summaries and the cited articles as inputs. Antognini and Faltings (2020) use Wikipedia in a similar way, in the video games domain. Fabbri et al. (2019) rely on the website Newser with lengthy human-aggregated extractive summaries. Gholipour Ghalandari et al. (2020), also based on WCEP, is especially relevant.

However, our dataset is different in the following ways. First, all of our articles in the same cluster are about the same event. Next, a large part of our dataset is annotated with article-summary entailment information (i.e., in each of the clusters, for each article in the cluster, whether the article entails the summary; see Section 3). Further, among the annotated article-summary pairs, about half of the articles entail the summary, and half of the articles do not entail the summary. This property makes a realistic and difficult setting for AgreeSum tasks.

In terms of recent MDS neural methods, Chu and Liu (2019) summarize opinions using an auto-encoder, in which case the input is much shorter than a typical article. Liu et al. (2019a) improve the model by encoding articles and summaries in the same space. Other novel approaches include using sentence compression in the seq2seq framework (Baziotis et al., 2019), jointly learning sentence fusion and paraphrasing (Nayeem et al., 2018), using graph neural networks to help extraction (Wang et al., 2020b), using spectral methods (Wang et al., 2020c), using transfer learning based on a novel pretraining method called gap-sentence prediction (Zhang et al., 2020a) on a news-specific corpus, among a few others (Li et al., 2020; Gu et al., 2020; Mao et al., 2020).

For MDS tasks that summarize the intersection of articles, a few past works have discussed the helpfulness of models that identify common infor-

mation "centroids" among multiple related documents, so as to allow internet users to more efficiently understand events (Radev et al., 2000; Barzilay and McKeown, 2005). The attempted non-neural models rely heavily on topic/theme detection and tracking, and are more extractive than abstractive (Radev et al., 2004). The AgreeSum idea has not been fully explored by researchers since much stronger text generation technologies became available. It is timely to revisit the problem also because recent stronger neural abstractive summarization models are prone to hallucination, as are neural text generation in general (Wiseman et al., 2017; Tian et al., 2019; Wang and Sennrich, 2020; Pang and He, 2020).

Summarization hallucination and evaluation. Non-hallucination is a necessary but not sufficient condition for performing well in AgreeSum: the summary not only needs to be entailed in the union of the articles, but also must be entailed in each of the articles. Unfortunately, recent works have shown the difficulty of identifying and mitigating hallucination (Maynez et al., 2020).

Evaluation-wise, researchers have found that metrics like ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020b) are only weakly correlated with factuality. Durmus et al. (2020) and Wang et al. (2020a) have therefore proposed using question-answering systems for evaluating summarizers. Recently, Zhou et al. (2021) have made progress in creating token-level hallucination detectors which rely on negative (i.e., hallucination) data augmentation to train.

In terms of improving faithfulness and factuality, researchers did not find natural language inference (NLI)[4] models trained on standard NLI datasets to be robust enough for summarization-related downstream tasks (Falke et al., 2019). Contemporaneously, entity chains are used to explicitly ground the generations so that they become more faithful (Narayan et al., 2021). More broadly, there have been other recent works striving to develop techniques for high-precision generation (Malmi et al., 2019; Tian et al., 2019; Pang and He, 2020; Parikh et al., 2020; Dušek and Kasner, 2020).

[4]The NLI datasets, beginning with SNLI (Bowman et al., 2015), correspond to a three-class (entailment, neutral, contradiction) entailment classification task.

3 AgreeSum Task and Datasets

3.1 Task

Short description of AgreeSum. The input is a cluster of around four articles (refer to Section 3.2 for more details) that describe the same event. However, the articles may have different levels of details and/or different levels of neutrality; e.g., one article in a cluster may be an opinion, while other articles may be neutral news. The expected output is an abstractive summary that represents information common and faithful to all input articles in the cluster. Moreover, the summary needs to be informative.

3.2 English AgreeSum Dataset Based on WCEP

Step 1. Recall that WCEP contains neutral human-written news summaries, each of which is linked to a news article. The first step is to obtain the summaries and 8 on-topic articles[5] for each summary based on WCEP. Specifically, we collect 5564 human-written summaries $\{\boldsymbol{y}_i\}_{i=1}^{5564}$ from WCEP. For each summary, we have one linked article to each summary on WCEP; we call the set of such articles $\{\boldsymbol{x}_i^{(0)}\}_{i=1}^{5564}$. Given that we want to generate abstractive summaries, and to make the dataset challenging, the set of articles $\{\boldsymbol{x}_i^{(0)}\}_{i=1}^{5564}$ will *not* be used in the final dataset, to prevent excessive textual overlaps between the input articles and the target summaries. For each i, we obtain 8 other news articles $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(8)}$ that are semantically similar to $\boldsymbol{x}_i^{(0)}$, based on a proprietary BERT-based clustering algorithm.

Step 2. The second step is to annotate entailment relations. Annotators are asked to judge if each of the articles in a cluster entails the summary (i.e., "does the article contain all the information presented in the summary?"). 1025 *cluster*-summary pairs are annotated. We designate ∼10% of the annotated clusters to be in the dev set. This step is discussed further in Section 4.

Step 3. To make AgreeSum moderately but not overly difficult, we set the maximum number of articles per cluster to be four, and the next step is to transform the dataset to have four articles per cluster. To take advantage of entailment annotations, we duplicate each annotated cluster in the training

[5]A small amount of clusters (233, or ∼4%) have fewer than 8 articles to ensure relevance of all articles in a cluster.

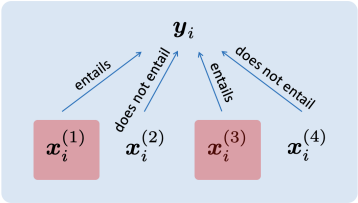

Figure 1: An annotated example (i.e., input-output pair where the input is a cluster of articles and the output is a summary) in the training set. In the figure, two articles in the input cluster entail the summary while the other two do not. An example is either annotated (meaning all article-summary pairs in the cluster-summary pair are annotated) or unannotated (meaning none of the article-summary pairs are annotated). Note that in the dev set, all articles entail the summary given that we would like to compare between generated summaries and the gold summaries.

set $\boldsymbol{c}_i = \{\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(8)}\}$ ten times so that there is roughly an equal number of annotated and unannotated clusters.[6] Next, for each cluster, we randomly choose four articles to keep in the cluster, and discard the rest.

Dev and test sets. For the development set, we aim to designate the WCEP summaries as gold summaries. Therefore, for each cluster, we only keep the articles that entail the corresponding summary based on human annotation. In the case that a cluster has > 4 articles that entail the summary, we split the cluster into two, such that each newly formed cluster has ≥ 2 articles that entail the summary.

We use WCEP entries corresponding to dates before August 2019 for the training and development sets; for the test set, we sample 150 clusters of articles containing WCEP linked articles published between August 2019 to August 2020 to ensure that the test set does not have overlapping articles or overlapping publication times with the training and development sets.

For each sampled test cluster, we sample four articles from the cluster. Unlike the development set, these four articles are not annotated and are not guaranteed to entail any particular common summary. As a result, the provided WCEP summaries in the test set are **not** gold-standard summaries.

The test set is expected to be challenging due to construction methods, as well as the shift in topics of recent events (e.g., the COVID-19 pandemic). The test set WCEP summaries are not gold-standard summaries, but are nevertheless provided given the potential use to approximate generation informedness (Section 6).

We thus obtain the dataset split shown in Table 1.

	# of cluster-summary pairs			# of article-summary pairs		
	train	dev	test	train	dev	test
all	18208	132	150	70137	423	600
annotated	9130	132	0	33841	423	0
(at least 1) entailed	7610	132	0	17951	423	0
unannotated	9078	0	150	36296	0	600

Table 1: Dataset information. A cluster contains $\leq$ 4 articles. The "(at least 1) entailed" row refers to the number of *annotated* clusters containing at least one article that entails the summary (in the cluster-summary case), or the number of articles that entail the summary *among the annotated pairs* (in the article-summary case).

4 Article-Summary Entailment

4.1 The Entailment Dataset

Workers have annotated the entailment relation of 1025 cluster-summary pairs (a subset of the cluster-summary pairs obtained in Step 1 in Section 3.2), which correspond to 7750 article-summary pairs. For each article-summary pair, we ask *five* professionally trained annotators, randomly chosen from a pool of ∼800 raters, whether the summary is semantically entailed in the article; we then take the majority answer. See the appendix for more details.

Our AgreeSum models leverage this entailment dataset, as discussed in Section 5. In addition, this dataset could be seen as a challenging RTE-style (or two-class NLI-style) task.

4.2 Model Performance

To explore how models perform on the article-summary entailment dataset, and to see whether we can have a good entailment-classification model to guide the learning of our summarizer that encourages faithfulness to all input articles in the same cluster, we treat the dataset as a two-class NLI task where the label is either "entailed" or "not entailed," and examine the classification accuracy. Specifically, we evaluated the following three models. RoBERTa-large (Liu et al., 2019b) fine-tuned on MNLI fails on our entailment data[7], likely in

[6] A side effect is that such choices could allow for more supervised cross-entropy updates.

[7] In this case, the majority of the predictions are "not entailed."

part because the premises in MNLI are sentences but are articles in our article-summary entailment task. It is also worth noting that unlike T5 (Raffel et al., 2020), RoBERTa-large is not pretrained on a multi-task mixture of many tasks, with each task converted into the text-to-text format.

We also attempt a model that integrates the PEGASUS-encoder which is pretrained in the news domain using the HugeNews corpus which contains around 1.5 billion articles. Given that pretrained PEGASUS does not include `[CLS]` tokens and it is very expensive to re-pretrain, we use a CNN-based classifier (Kim, 2014) whose input is the PEGASUS-encoder-outputs; more specifically, the convolutional layers pool over the sequence of encoder outputs. Based on the architecture by Kim (2014), our CNNs use filter n-gram sizes of 2, 3, 4, and 5, with 256 filters each. The resulting classifier achieves ∼68% accuracy on our entailment dataset.[8]

In comparison, T5 (Raffel et al., 2020) shows encouraging results. The multi-task-trained T5-large fine-tuned on our training set achieves 81.3% accuracy (79.1% for vanilla T5-large). T5-small fine-tuned on our training set achieves 79.5% accuracy (76.4% for vanilla T5-small).

We aim to see if using an entailment signal from a moderately good article-summary entailment classification model would help produce summaries that satisfy the AgreeSum criteria.

5 AgreeSum Baselines and Approaches

5.1 Notations and Baselines

We first provide some notations so as to allow easier discussion. Suppose the clusters of articles in the training set are denoted by $\{\boldsymbol{c}_i\}_{i \in N}$. Each cluster contains at most four articles: $\boldsymbol{c}_i = \{\boldsymbol{x}_i^{(j)}\}_{j \leq 4}$, with the summary $\boldsymbol{y}_i$.

For cluster i, let $\boldsymbol{e}_i$ be the set of indices that correspond to articles that entail the summary. For example, in Figure 1 which corresponds to the ith cluster, we have $\boldsymbol{e}_i = \{1, 3\}$. Let E and D denote the encoder and the decoder of PEGASUS, respectively.

The following baselines are based on PEGASUS pretrained using gap-sentence prediction on HugeNews with 1.5 billion articles (3.8 terabytes). Note that despite being an SDS model, PEGASUS also achieves near-SoTA results on the Multi-News MDS dataset (Fabbri et al., 2019), so it is a competitive baseline for MDS as well.

B1: finetuning on $(\boldsymbol{x}_i^{(0)}, \boldsymbol{y}_i)$ pairs. Recall that for each summary $\boldsymbol{y}_i$, we drop the WCEP-linked article $\boldsymbol{x}_i^{(0)}$ to prevent excessive textual overlaps between the input cluster and the output summary. In this baseline, however, we use $\boldsymbol{y}_i$'s in the training set and its corresponding $\boldsymbol{x}_i^{(0)}$'s (5452 pairs) for supervised finetuning.

Why can we use $\boldsymbol{x}_i^{(0)}$ as gold targets? On WCEP, it is reasonable to assume that the summaries are directly connected to the linked articles, and therefore, $\boldsymbol{x}_i^{(0)}$ entails $\boldsymbol{y}_i$. Moreover, these summaries also inform the model of the style of the WCEP summaries. However, the downside is that the model could potentially prioritize extractions from the articles over entailment (i.e., the property that the generated summary is entailed in each article), as we would see in Section 6.

B2: concatenating truncated inputs. We finetune PEGASUS on the following: for each cluster $\boldsymbol{c}_i$, we truncate $\{\boldsymbol{x}_i^{(j)}\}_j$ for each $j \in \boldsymbol{e}_i$ and concatenate them such that the concatenated sequence has length ≤ 1024, given hardware constraints. We use a special symbol to delineate article boundaries.

B3: B1+B2. We first train using B1 and then finetune using B2, which may improve over B1 or B2 alone.

B4: merging encodings and decode. Inspired by Chu and Liu (2019), we first encode $\boldsymbol{x}_i^{(j)}$ separately for each $j \in \boldsymbol{e}_i$, and pass the average of the encodings to the decoder (i.e., $\frac{1}{|\boldsymbol{e}_i|} \sum_{j \in \boldsymbol{e}_i} E(\boldsymbol{x}_i^{(j)})$). Next, we do supervised learning based on the WCEP summaries.

B5: best lead-1 sentence by entailment score. We first extract the first sentence of each article in the cluster. Next, we rerank the sentences using an entailment score, which is the mean of the binary entailment labels, predicted by fine-tuned T5-large (Section 4), between each article in the cluster and the given sentence. We choose the summary corresponding to the highest score, breaking ties randomly.

[8] Doing intermediate training on MNLI (Pruksachatkun et al., 2020) and then training on our entailment dataset, unfortunately, does not strengthen performance.

5.2 AgreeSum Model (ASM): Leveraging Unannotated Data

We introduce another baseline model called ASM. The distinguishing feature is that to target cluster-summary entailment (meaning the summary is entailed in each article in the cluster), we attempt to use the T5-entailment-classification results (discussed in Section 4) as a training signal.

Supervised pretraining and training. Given pretrained PEGASUS, we first fine-tune according to B1. Next, for each cluster, we concatenate all elements of $\boldsymbol{c}_i$ similar to B2, and mask out the articles that do not entail $\boldsymbol{y}_i$ by the padding symbol; effectively, we use $\{\boldsymbol{x}_i^{(j)}\}_{j \in \boldsymbol{e}_i}$ as input to the transformer. Then, we use the standard cross-entropy loss L_{ce} to train the summarizer. However, only a small number of clusters are annotated, providing a very limited supervised signal.

Unsupervised entailment loss. We complement L_{ce} with the entailment loss L_e to learn entailment behavior. We fine-tune T5-small on our training dataset to predict entailment, and use this as our entailment classifier F_e. In practice, we obtain T5-outputs using remote procedure calls (RPCs).

F_e takes in an article $\boldsymbol{x}$ and a summary $\boldsymbol{y}$ as inputs, and outputs 1 if $\boldsymbol{x}$ entails $\boldsymbol{y}$ and -1 otherwise. The loss L_e is based on policy gradients (Williams, 1992; Sutton et al., 2000); we aim to maximize a sequence-level metric of a summary decoded from the model during training: $\mathbb{E}_{\tilde{\boldsymbol{y}} \sim p_\phi} \sum_{k=1}^{4} F_e(\boldsymbol{x}_i^{(k)}, \tilde{\boldsymbol{y}})$ where ϕ stands for the parameters of the encoder E and the decoder D. We thus have the following gradient:

$$\nabla_\theta L_e(\phi) = -\mathbb{E}_{\tilde{\boldsymbol{y}} \sim p_\phi} \nabla_\phi \log p_\phi(\tilde{\boldsymbol{y}}) \hat{Q}(\tilde{\boldsymbol{y}}),$$

where

$$\hat{Q}(\tilde{\boldsymbol{y}}) = \sum_{k=1}^{4} F_e(\boldsymbol{x}_i^{(k)}, \tilde{\boldsymbol{y}})$$

is the sequence-level return. Intuitively, during training, we sample a summary $\tilde{\boldsymbol{y}}$ from our model, and run it through the T5 entailment classifier (which is separate from our summarizer) to obtain $\hat{Q}(\tilde{\boldsymbol{y}})$. We then weight the MLE gradient (taking $\tilde{\boldsymbol{y}}$ as the target) by $\hat{Q}(\tilde{\boldsymbol{y}})$. L_e thus aims to guide our summarizer to generate summaries that entail in all or most of the articles.

Unsupervised language loss. Only using the entailment loss L_e may result in degenerations. One way to encourage the model to generate fluent summaries and summaries that look like WCEP summaries is by using GAN-style (Goodfellow et al., 2014) objectives, which have achieved good performance in some conditional generation tasks like textual style transfer and machine translation (Shen et al., 2017; Wu et al., 2018; Pang and Gimpel, 2019). We thus use F_l, a "language classifier" (i.e., discriminator) that distinguishes model generations from real dataset summaries (no matter annotated or not), to force our summarizer to generate sentences that look like the human-written summaries. F_l is based on the CNN settings by Kim (2014) identical to the setting introduced in Section 4, while the inputs to CNNs are sentence representations obtained using PEGASUS.

Specifically, suppose there are k examples in a minibatch, then

$$L_l = -\frac{1}{k} \sum_{i=1}^{k} \left[\log F_l(\boldsymbol{h}_i) + \log(1 - F_l(\tilde{\boldsymbol{h}}_i)) \right],$$

where $\boldsymbol{h}_i$ is the decoder hidden states of WCEP summaries, and $\tilde{\boldsymbol{h}}_i$ is the decoder hidden states of model generations, by professor forcing (Lamb et al., 2016). In the professor forcing algorithm, the input to F_l is the hidden states instead of hard tokens so as to address the mismatch between training-time sequence prefix and test-time sequence prefix.

Summary. We alternate updates among the following:

(1) supervised training: updating E, D to minimize the loss L_{ce};

(2) unsupervised training: updating E, D to minimize the loss $L_e - \lambda L_l$;

(3) language classifier training: updating F_l to minimize the loss L_l.

5.3 Implementation

We implement our model as a fork of the open-source 568M-parameter PEGASUS model (Zhang et al., 2020a).[9] We initialize PEGASUS from the "mixed & stochastic" checkpoint, which was pretrained for 1.5M steps.

All baseline models are fine-tuned with a learning rate of 1e-4. All of the ASM models use 5e-5

[9] https://github.com/google-research/pegasus

model	ROUGE-dev 1	2	L	ROUGE-test‡ 1	2	L	article-summary entail % (↑) ¶	cluster-summary entail % (↑) ¶	hallucination % (↓) ¶	language % (↑)	n-gram overlap % (↓) $n=3$	$n=4$	$n=5$	$n=6$
B1	45.7	25.0	38.3	30.7	12.3	24.2	56.8 / 54.0	22.7 / 20.1	6.0 / 6.0	90.0	60.4	49.8	42.6	38.1
B2	46.1	23.5	37.9	28.5	10.4	22.0	50.8 / 49.0	23.3 / 22.2	18.7 / 19.5	92.0	27.8	13.5	7.3	4.1
B3	47.1	23.7	38.1	29.1	10.8	22.4	49.7 / 50.7	22.7 / 23.3	16.0 / 20.0	96.0	26.1	12.8	6.9	4.0
B5	33.9	13.6	26.1	7.70	2.59	5.39	58.8 / 65.1	22.0 / 27.3	0.0 / 0.0	93.3	100.0	100.0	100.0	100.0
ASM	44.4	22.8	37.0	27.5	11.0	22.4	62.8 / 66.0	30.7 / 39.1	10.0 / 8.8	96.0	47.3	36.4	30.2	26.1

Table 2: Results (on test set if not specified). ¶: the first number corresponds to T5-large-evaluated results, and the second number corresponds to human-evaluated results. For each row, human raters annotated the entire test set; each article-generation pair is annotated by three raters, and we take the majority answer. The best result in each column is in blue; the worst in red. n-gram overlap: the proportion of generation n-grams that are also in the source. ‡: the test set WCEP summaries should not be treated as references, given that there is no guarantee that the WCEP summary is entailed in each of the articles; the test set WCEP summaries are provided as an approximate measure of informedness. Note: B4 results in extensive hallucinations and very low ROUGE (∼10), and the ablation study of ASM without L_l produces heavy degenerations, so they are omitted.

(tuned in {1e-5, 5e-5, 1e-4, 2e-4}). All models use beam search for decoding (beam size 8, beam alpha 0.8). Given hardware constraints, all models use a max input length of 1024. The max output decoding length is set to be 128. In addition, we tune $\lambda \in \{0.05, 0.1, 0.3, 0.5, 1, 2\}$ and choose $\lambda = 0.1$ for all reported experiments that include the L_l. We do not change any other default hyperparameter settings adopted from PEGASUS. Please refer to the appendix for more details.

6 Results

The following two models are not included in the table given their poor performance. (1) The ablation study of ASM without language loss does not produce meaningful outputs. In this case, given that the T5 entailment classifier does not encourage language quality, the summaries in fact degenerate heavily. (2) Model B4 (merging encodings and decode) is omitted from the table given very poor ROUGE performance (∼10) and extensive hallucination. Our conjecture is that the mean of the encodings of articles does not correspond to a meaningful encoding in the case of PEGASUS.

6.1 Agreement and Hallucination

First, we claim that ASM achieves better agreement. There are two types of agreement: *article-summary* agreement which is the proportion of summaries entailed in the articles, and *cluster-summary* agreement which is the proportion of clusters in which *all* articles entail the summary. Table 2 reports both the T5-automatic evaluation and the human evaluation results on agreement.

(Preliminary) automatic metrics. Given an article-summary pair, we use T5-large fine-tuned on our entailment dataset (Section 4) to predict whether the summary is entailed in the article. For article-summary agreement, we compute the percentage of "entailed" classifications. For cluster-summary agreement, we compute the percentage of clusters where all article-summary pairs lead to "entailed" classifications. We see that the T5-evaluation results for ASM models perform better than the respective results for baseline models.

Human evaluation. For each row of Table 2, human raters annotated the entire test set (150 clusters, which corresponds to 600 article-generation pairs), on whether the generated summaries are entailed in the article. Workers were asked:

> *Does the article contain all the information presented in the summary?*

The full prompt is available in the appendix. One design choice is that we merge all article-summary pairs for each cluster together into one task/HIT. Therefore, each task/HIT corresponds to four article-summary annotations. To reduce the inherent variance in human evaluation, each article-generation pair is annotated by three different raters, and we take the majority answer. Please see the appendix for more details.

For article-summary agreement, we see that the human evaluation for the ASM models performs a little better than the extractive results (B5). ASM models perform more than 10 points better than the best abstractive baseline results (B1), which is in turn ∼5 points better than the other abstractive baseline results (B2, B3). For cluster-summary

agreement, this improvement is even clearer, with ASM models performing more than 10 points better than any baseline.

In addition, we see that ASM reduces hallucination compared to B2 and B3. One way to approximate hallucination is by the number of clusters in which none of the articles entail the generated summary. Using both automatic and human evaluation results, we see that our model does better than B2 and B3, but a little worse than B1 which copies extensively from the source articles.

6.2 Discussion on ROUGE

In SDS tasks, Durmus et al. (2020) and Wang et al. (2020a) observe that ROUGE and BERTScore have a small correlation with summary factuality.

A hallucination or non-entailment[10] can have major text-span overlaps with the reference, thereby having a large ROUGE score. Given the nature of AgreeSum, and by Table 2, we confirm that high ROUGE does not imply entailment and should not be considered heavily in evaluation.

On the other hand, intuitively, we do recognize that an overly small ROUGE may indicate bad generations (e.g., extremely short generations, off-topic generations, and other degenerations like repetitions), which is the case for B4 generations as well as ASM-minus-L_l generations.

Thus, practitioners need to rely on and determine the desired tradeoff between the following two automatic metrics: (1) ROUGE as a coarse proxy for summary quality and informedness, and (2) entailment-related and hallucination-related metrics (Section 6.1).

As a reminder, the development set summaries can be treated as gold-standard summaries. However, the test set summaries are not gold-standard summaries; they are only provided so as to allow one way to measure generation quality and informedness. Unlike the development set, none of the test-set input articles are filtered out through the summary-article entailment annotation procedure, given that we do not want to introduce potential bias through too much manipulation and filtering on raw test clusters.

6.3 More Observations

Language. We also asked workers to judge the language of the generations:

> *Is this summary coherent and well-written with no self-contradictions or capitalization, spelling, punctuation, or grammar errors?*

The full prompt is in the appendix. We see that the ASM-generated summaries are marginally better at the above. However, ASM without language loss results in heavy degenerations. On a separate note, we see that ASM generations tend to copy more than B2 and B3, but less than B1.

Examples. The appendix contains a few examples that compare generations from different models. For example, in Table 5, given four articles, we see the different generations that the systems produce. Article 4 is an opinion piece. ASM model correctly abstractively summarizes Article 2 and 4 in a way that agrees with 1 and 3. B1 copies from Article 1, while B2 and B3 have hallucinations. Given space constraints, please refer to the appendix.

6.4 Extension: Post-hoc Entailment Reranking after Decoding a Beam

To generate summaries that achieve better agreement, we also attempted a decoding trick which we name as entailment-oriented decoding, denoted by entdec in Table 3.

We first define an entailment score used in this case. Given a cluster of articles and a generated summary, the entailment score is the mean of the T5-large-predicted binary labels (1 corresponds to "entailed" and 0 corresponds to "not entailed").

A model is suffixed as X-entdec$k$ if we decode from X using beam search with beam size k; and after obtaining the size-k beam, we select the generation that corresponds to the largest T5 entailment score. We pick the beam with the largest score, using the original beam probabilities to break ties. Intuitively, this trick picks the best-entailment summary locally given that the generations in the same beam are usually similar.

Table 3 shows that the entdec-trick generations indeed achieve higher article-summary agreement and human-summary agreement. We see that ASM still maintains the advantage in agreement, even if compared to entdec-decoded generations from other baselines.

6.5 Discussion: Improving Entailment Using T5-Based NLI-Style Models

Falke et al. (2019) find that NLI models trained on

[10] Non-entailed summaries are not necessarily hallucinations, given that the non-entailed summaries could correspond to some articles in the cluster but not the rest of the articles.

model	ROUGE-dev 1	2	L	ROUGE-test‡ 1	2	L	article-summary entail % (↑) ¶	cluster-summary entail % (↑) ¶	hallucination % (↓) ¶	language % (↑)	n-gram overlap % (↓) 3	4	5	6
B1-entdec8	45.7	24.8	38.3	29.5	12.0	23.4	60.8 / 56.8	26.0 / 26.2	4.67 / 6.0	93.3	61.1	50.7	43.5	38.8
B2-entdec8	46.1	23.3	38.0	28.5	10.4	22.0	53.2 / 58.3	23.3 / 34.2	18.0 / 18.1	95.3	28.1	13.8	7.4	4.1
B3-entdec8	47.2	23.8	38.1	28.9	10.6	22.1	53.7 / 59.3	24.7 / 33.6	15.3 / 13.4	94.0	26.7	13.0	6.9	4.0
ASM	44.4	22.8	37.0	27.5	11.0	22.4	62.8 / 66.0	30.7 / 39.1	10.0 / 8.8	96.0	47.3	36.4	30.2	26.1
ASM-entdec8	44.8	23.4	37.7	26.9	11.0	22.1	68.2 / 63.3	40.0 / 37.0	8.0 / 10.9	91.3	46.9	35.4	29.1	24.6
ASM-entdec16	44.7	23.1	37.5	26.3	10.8	21.5	70.5 / 63.8	42.7 / 39.9	8.0 / 11.9	90.0	48.7	36.9	30.3	25.6

Table 3: Results using the entdec decoding strategy. The results correspond to the test set performance, if not specified. ¶: the first number in each cell corresponds to T5-evaluated results, and the second corresponds to human-evaluated results. Given that B5 relies on pure extraction, the entdec methods are not applicable. ‡: the test set WCEP summaries should not be treated as references, given that there is no guarantee that the WCEP summary is entailed in each of the articles; the test set WCEP summaries are provided as an approximate measure of informedness.

standard NLI datasets do not offer robust benefits to improving summarization factuality. Maynez et al. (2020), on the other hand, rerank four summaries generated by four different models using BERT-based MNLI models, and find small improvements in faithfulness and factuality. However, these works rank different summaries after decoding assuming an existing summarizer (similar to our entdec trick), instead of updating the model parameters directly. Our contribution lies in the fact that we successfully use entailment models to improve the model during training time.

The major feature of our T5-based NLI model is that (1) our NLI model is based on multi-task-pretrained T5, implying that pretrained T5 can already handle article-length inputs well in certain tasks, and (2) our model is obtained after finetuning on our article-summary entailment dataset. Therefore, our T5-based NLI model is much better adjusted to the length of the premises (given that traditional NLI tasks correspond to sentence-level entailment, but our case corresponds to article-summary entailment). We thus see that using simple T5-based binary signals can successfully improve entailment. However, more complicated modeling may be necessary if the AgreeSum cluster size becomes much larger.

7 Conclusion

We discuss the AgreeSum task with its dataset, and a range of baseline models. AgreeSum is timely given the recent focus on summarization faithfulness. In fact, we show that the summaries produced by several powerful pretraining-based baseline models are not able to follow AgreeSum's requirements satisfactorily. We welcome the community to contribute more advanced methods that work well on AgreeSum, especially when only a small subset of the dataset is labeled with article-summary entailment information.

Within the AgreeSum dataset, we also provide article-summary entailment annotations on a subset of clusters, which we hope can contribute to the recent effort in improving abstractive summarization faithfulness.

Moreover, while there is contemporaneous development of complex approaches to encourage generated abstractive summaries to be entailed in the source articles, we show that it is feasible to improve the entailment behavior of generated summaries based on a binary article-summary entailment classifier.

Acknowledgement

The authors would like to thank Pepa Atanasova, He He, Abe Ittycheriah, Jialu Liu, Tianqi Liu, Ji Ma, Shashi Narayan, Alicia Parrish, Jiaming Shen, Gonçalo Simões, and You (Will) Wu (alphabetical order) for the valuable discussions, and the anonymous reviewers for the detailed reviews.

References

Diego Antognini and Boi Faltings. 2020. GameWikiSum: a novel large multi-document summarization dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6645–6650, Marseille, France. European Language Resources Association.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3:

Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. 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 673–681, Minneapolis, Minnesota. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232, Long Beach, California, USA. PMLR.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

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

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

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating representative headlines for news stories. In *Proceedings of The Web Conference 2020*, pages 1773–1784.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. 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.

Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, pages 4601–4609.

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

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

Peter J. Liu, Yu-An Chung, and Jie Ren. 2019a. Summae: Zero-shot abstractive text summarization using length-agnostic auto-encoders. *arXiv preprint arXiv:1910.00998*.

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

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

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

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.

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

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*, pages 1906–1919, Online. Association for Computational Linguistics.

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

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simoes, and Ryan McDonald. 2021. Planning with entity chains for abstractive summarization. *arXiv preprint arXiv:2104.07606*.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1191–1204, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Bruce Jay Nelson. 1981. Remote procedure call.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 138–147, Hong Kong. Association for Computational Linguistics.

Richard Yuanzhe Pang and He He. 2020. Text generation by learning from off-policy demonstrations. *arXiv preprint arXiv:2009.07839*.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Over Paul and Yen James. 2004. An introduction to duc-2004. In *Proceedings of the 4th Document Understanding Conference (DUC 2004)*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

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.

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

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020a. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020b. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Kexiang Wang, Baobao Chang, and Zhifang Sui. 2020c. A spectral method for unsupervised multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 435–445, Online. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. volume 95 of *Proceedings of Machine Learning Research*, pages 534–549. PMLR.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chunting Zhou, Jiatao Gu, Mona T. Diab, Paco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation.

Markus Zopf. 2018. Auto-hMDS: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

A Appendix

A.1 Human Evaluation

Recall that human evaluation results are reported in Table 2 and Table 3. Specifically, we asked human raters to annotate the entailment relationship of each article-generation pair. Each row in the table corresponds to a model, and for each model, human raters annotated the entire test set that contains 150 clusters (which corresponds to 600 article-generation pairs). Moreover, to reduce the variance of human evaluation and to improve the confidence to our claims, each article-generation pair is annotated by three different raters, and we take the majority answer.

Our prompts to human raters are described as follows. We merge all article-summary pairs for each cluster together into one task/HIT. Therefore, each task/HIT corresponds to four article-summary annotations.

Instruction at the top of the annotation page: "In this task you will be given one summary of a news story and the text of four news articles. You will be asked to evaluate whether the summary is coherent and well written with no self-contradictions or capitalization, spelling, punctuation, or grammatical errors. Furthermore, for each news article, you will be asked to evaluate: whether the article contains ALL the information presented in the summary. This is equivalent to asking, 'Using only the text of this news article, would it be possible for someone to write this summary?"'

Language: "Is this summary coherent and well-written with no self-contradictions or capitalization, spelling, punctuation or grammar errors? Select "Yes" if this is a coherent summary written in fluent English with perfect grammar and style. Select "No" if the summary contains one or more capitalization, spelling, punctuation, or grammatical errors, or is incoherent, self-contradictory, or otherwise badly written."

Entailment: "Does the article contain all the information presented in the summary?"

A.2 Entailment Annotation in Dataset Creation

We used the same entailment prompt described in Section A.1 to obtain entailment information for a subset of the clusters. The difference is that we used five annotators per article-summary pair instead of three, to ensure the quality of the supervised split of the training set as well as the entire development set.

A.3 More on Reproducibility

We implement our model as a fork of the open-source 568M parameter PEGASUS model.[11]

Baseline models were trained for 20k steps, with the best checkpoint selected based on dev-R2. Final selections are B1 at 3k steps, B2 at 6k steps, and B3 at 4k steps. Proposed model was trained until convergence based on T5-small entailment scores reported during the sampling steps in the policy gradients REINFORCE algorithm. Final selection of this model was at 43k steps.

Runtime details. Given the chosen PEGASUS implementation, our models are trained on TPUs. Given hardware constraints, F_e, implemented via T5-small fine-tuned on our annotated entailment dataset, is served on CPU on a separate machine. In fact, we run 80 replicas of this machine to improve throughput. We make RPC calls (Nelson, 1981) to this cluster of T5-serving machines during policy gradients training (using the f.contrib.rpc module). Baseline models took approximately 24 hours each to train, with the proposed model taking approximately 4 days. Our conjecture is that for the latter, most of the time is spent on the communication between PEGASUS and T5, as well as the expensive computation of T5 on CPU.

A.4 Examples

We now provide some example generations of the AgreeSum task. Given that the input articles are long, we comment out parts of the articles in the tables. Please refer to Table 4, Table 6, and their captions.

For example, in Table 5, Article 4 is an opinion piece. ASM model correctly abstractively summarizes Article 2 and 4 in a way that agrees with 1 and 3. B1 copies from Article 1, while B2 and B3 have hallucinations. In Table 4, all models except for B1 perform well here. B1 entails Article 1 and 2, but not 3 and 4. Similar comparisons can be made in Table 6.

[11] https://github.com/google-research/pegasus

ASM: Sérgio Moro resigns as Brazil's Justice Minister after accusing President Jair Bolsonaro of interfering in the country's federal police.
B1: Brazil's Supreme Court authorises a police investigation into President Jair Bolsonaro.
B1+entdec8: Brazil's Supreme Court authorises a police investigation into President Jair Bolsonaro.
B2: The Minister of Justice of Brazil, Sérgio Moro, resigns after accusing President Jair Bolsonaro of interfering in the operations of the federal police.
B2+entdec8: The Minister of Justice of Brazil, Sérgio Moro, resigns after accusing President Jair Bolsonaro of interfering in the operations of the federal police.
B3: The Minister of Justice of Brazil, Sérgio Moro, resigns after accusing President Jair Bolsonaro of interference in the federal police.
B3+entdec8: The Minister of Justice of Brazil, Sérgio Moro, resigns after accusing President Jair Bolsonaro of interference in the federal police.
B5: Brazil's government has been plunged into turmoil after the resignation of one of Jair Bolsonaro's most powerful ministers sparked protests, calls for the president's impeachment and an investigation into claims he had improperly interfered in the country's federal police.
Article 1: [Link: https://www.theguardian.com/world/2020/apr/24/justice-ministers-sacking-plunges-brazil-into-turmoil; the article is not duplicated due to length and copyright]
Article 2: [Link: https://www.ft.com/content/62d04bb5-6825-41ec-b263-4ceeaec58049; the article is not duplicated due to length and copyright]
Article 3: [Link: https://www.theguardian.com/world/2020/apr/26/bolsonaro-in-fresh-crisis-over-sons-alleged-links-to-fake-news-racket; the article is not duplicated due to length and copyright]
Article 4: [Link: https://www.thedailybeast.com/brazils-justice-minister-sergio-moro-quits-accuses-president-jair-bolsonaro-of-misconduct-resigns; the article is not duplicated due to length and copyright]

Table 4: Example generations from test set. All models except for B1 perform well here. B1 entails Article 1 and 2, but not 3 and 4.

ASM: The NFL announces that it will play during the coronavirus pandemic.
B1: NFL Commissioner Roger Goodell sends a letter to fans outlining the league's plans to play during the coronavirus pandemic.
B2: The NFL cancels the remainder of the 2020 season due to the coronavirus outbreak.
B3: The NFL cancels the remainder of the 2019 preseason due to the ongoing coronavirus outbreak.
B5: NEW YORK (AP) — NFL Commissioner Roger Goodell has sent a letter to fans outlining the league's plans to play during the coronavirus pandemic.
Article 1: [Link: https://apnews.com/article/nfl-sports-virus-outbreak-health-football-c10944a1b88bd593198b660d207c7b56; the article is not duplicated due to length and copyright]
Article 2: [Link: https://www.washingtonpost.com/sports/2020/07/27/nfl-cautiously-optimistic-despite-mlb-coronavirus-outbreak/; the article is not duplicated due to length and copyright]
Article 3: [Link: https://www.wxyz.com/sports/roger-goodell-sends-letter-to-nfl-fans-explaining-plans-for-season; the article is not duplicated due to length and copyright]
Article 4: [Link: https://www.wsj.com/articles/nfl-playersand-a-lot-of-new-england-patriotsare-opting-out-of-the-2020-season-11596030916; the article is not duplicated due to length and copyright]

Table 5: Example generations from test set. Note in particular that Article 4 is an opinion piece. The ASM model correctly abstractively summarizes Article 2 and 4 in a way that agrees with 1 and 3. B1 and B5 copy from Article 1, while B2 and B3 have hallucinations. More examples can be found in the appendix.

ASM: Late-night talk shows in New York City begin taping without audience members.
B1: The Late Show with Stephen Colbert, The Tonight Show with Jimmy Fallon, and The Full Frontal with Samantha Bee begin taping without audience members due to the spread of the coronavirus in New York City.
B1+entdec8: The Late Show with Stephen Colbert, The Tonight Show with Jimmy Fallon, and The Full Frontal with Samantha Bee begin taping without audience members due to the spread of the coronavirus in New York City.
B2: In response to the outbreak of coronavirus in the United States, all New York-based late-night talk shows will cease taping without an audience.
B2+entdec8: In response to the outbreak of coronavirus in the United States, all New York-based late-night talk shows will cease taping without an audience.
B3: Following the lead of the Centers for Disease Control and Prevention, all late-night talk shows in New York City will stop taping and start without an audience due to the outbreak of coronavirus. all late-night shows in New York City will stop taping and start without an audience Due to the outbreak of coronavirus, all late-night shows in New York City will stop taping and start without an audience.
B3+entdec8: Following the lead of the Centers for Disease Control and Prevention, all late-night talk shows in New York City will stop taping and start without an audience due to the outbreak of coronavirus. all late-night shows in New York City will stop taping and start without an audience Due to the outbreak of coronavirus, all late-night talk shows in New York City will stop taping and start without an audience Due to the outbreak of coronavirus, all late-night talk shows in New York City will stop filming and start without an audience due to the outbreak of coronavirus.
B5: The New York late-night circuit was the antithesis of "Live in Front of a Studio Audience" this week, prerecording shows without a crowd due to coronavirus fears.
Article 1: [Link: https://www.latimes.com/entertainment-arts/tv/story/2020-03-13/coronavirus-jimmy-fallon-stephen-colbert-no-audience; the article is not duplicated due to length and copyright]
Article 2: [Link: https://variety.com/2020/tv/news/late-night-shows-new-york-coronavirus-1203530972/; the article is not duplicated due to length and copyright]
Article 3: [Link:]https://abcnews.go.com/Entertainment/wireStory/late-night-comics-adjust-shows-audience-69581648; the article is not duplicated due to length and copyright]
Article 4: [Link: https://apnews.com/article/38f1afde2a5676fd3c2377f3719f5c86; the article is not duplicated due to length and copyright]

Table 6: Example generations from test set. ASM is very abstractive and entails all articles. B1 only entails Article 1 and 2. B2 incorrectly says that shows will "cease taping." B3 has repetition issues.