On the Intractability to Synthesize Factual Inconsistencies in Summarization

Ge Luo[1], **Weisi Fan**[1], **Miaoran Li**[1], **Youbiao He**[1], **Yinfei Yang**[2]*, **Forrest Sheng Bao**[1]
[1]Iowa State University , Ames, IA, USA
[2]Sunnyvale, CA, USA
{gluo,weisifan,limr,yh54}@iastate.edu
{yangyin7,forrest.bao}@gmail.com

Abstract

Factual consistency detection has gotten significant attention for the task of abstractive summarization. Many existing works rely on synthetic training data, which may not accurately reflect or match the inconsistencies produced by summarization models. In this paper, we first systematically analyze the shortcomings of the current methods in synthesizing inconsistent summaries. Current synthesis methods may fail to produce inconsistencies of coreference errors and discourse errors, per our quantitative and qualitative study. Then, employing the parameter-efficient finetuning (PEFT) technique, we discover that a competitive factual consistency detector can be achieved using thousands of real model-generated summaries with human annotations. Our study demonstrates the importance of real machine-generated texts with human annotation in Natural Language Generatioon (NLG) evaluation as our model outperforms the SOTA on the CoGenSumm, FactCC, Frank, and SummEval datasets.

1 Introduction

With the advancements in neural conditioned generation, abstractive summarization systems, which are dominantly based on neural networks, have achieved phenomenal performances. However, summaries generated so often contain content that is factually inconsistent with the source documents (Kryscinski et al., 2020; Maynez et al., 2020) and thus undermines the reliability and usability of the summaries. Thus detecting factual inconsistencies is an important task associated with summarization.

However, detecting inconsistencies in machine-generated summaries is not trivial. Due to the high labor cost of examining model-generated summaries, no existing datasets contain enough samples with human-annotated consistency labels for supervised learning in the conventional sense. As a workaround, data synthesis has been employed to increase the amount of training data in FactCC (Kryscinski et al., 2020), DocNLI (Yin et al., 2021), and MFMA (Lee et al., 2022b). They generate inconsistent summaries by negative sampling with pre-defined rules. Apart from training with synthetic inconsistent summaries, some other approaches (Kryscinski et al., 2020; Laban et al., 2022) leverage human-crafted claims in the Natural Language Inference (NLI) (Bowman et al., 2015) datasets. They measure factual consistency using the entailment relation between the source document and the summary. A recent work, SummaC (Laban et al., 2022), proposes to aggregate sentence-level pairwise entailment scores into a final consistency score.

We believe that the clue to improve inconsistency detection lies in the inconsistent samples that the state of the art (SOTA) fails to detect. By analyzing such samples in the famous SummaC benchmark, we find that certain types of factual inconsistencies are hard to be synthesized and thus are uncovered in the training of SOTA. Specifically, they are the coreference errors and discourse link errors defined by the Frank dataset (Pagnoni et al., 2021). A coreference error happens when a pronoun in the summary has a wrong referent than that in the document. A discourse error happens when the summary mistakenly mixes multiple statements in the document. These errors can occur in the summary when the source information is either in a single sentence or across multiple sentences.

The intractability to synthesize the said inconsistent training samples motivates us to take a different route to build an inconsistency detector via efficient use of limited human annotations on machine-generated summaries. Thanks to the Parameter-Efficient Fine-Tuning (PEFT) methods, we manage to finetune only 0.14% of the 0.9B parameters of

[*] No affiliation, currently working at Apple Inc.

Findings of the Association for Computational Linguistics: EACL 2024, pages 1026–1037
March 17-22, 2024 ©2024 Association for Computational Linguistics

the `DeBERTa-v2-xlarge-mnli` model using thousands of samples in the validation set of SummaC. Our model outperforms the SOTA on the CoGenSumm, FactCC, Frank, and SummEval datasets. Error rates in nearly all types of inconsistencies are improved by our approach.

Our code is available at `https://github.com/NKWBTB/FactFT`. We organize the paper as follows:

- First, we review the current synthetic methods on how they generate inconsistent summaries and their potential limitations.
- Then, we present a comprehensive case study on the inconsistent summaries missed by SOTA, revealing the gap between the summarizer-generated inconsistencies and synthesized inconsistencies.
- Finally, we present a document-level factuality classifier through parameter-efficiently finetuning a 0.9B model using only a few thousand human-annotated samples that outperforms all baselines, including ChatGPT, on four datasets.

2 How Good Are We at Synthesizing Inconsistencies?

The SOTA inconsistency detectors trained with synthetic inconsistent summaries still have a huge room for improvement. For example, the balanced accuracy of MFMA (Lee et al., 2022a) tops at 84.5% on six major inconsistency datasets. To propose an improvement, we argue that it is important to analyze the nature of factually inconsistent samples undetected by the SOTA detectors.

In this section, we first theoretically analyze the gap between the inconsistencies synthesized by SOTA for training and the real inconsistencies in summaries generated by neural generative models. Then we empirically study the gap using a case study on the SummaC benchmark with two SOTA approaches.

2.1 Existing Approaches to Synthesizing Inconsistent Summaries

We begin our study by reviewing how inconsistencies are introduced into synthetic data before such data is used to train SOTA inconsistency detectors and their potential limitations.

In summarization, the input and output texts are called the *document* and the *summary*, respectively. A *reference summary*, usually written by a human, is the expected, gold output or target in the ML sense. Many of the SOTA synthesize inconsistent summaries by manipulating the documents and/or the reference summaries.

FactCC (Kryscinski et al., 2020) synthesizes inconsistent summaries by sampling sentences from the document and applying the following transformations onto them: entity and number swapping, pronoun swapping, sentence negation, back translation, and token duplication and deletion. Potential limitations: Such token-level transformations may be too limited to cover the great variety of inconsistencies. In addition, such transforms operate on individual sentences, while an inconsistency often involves multiple sentences.

MFMA (Lee et al., 2022b) operates by masking tokens on both the document and the reference summary. First, a BART (Lewis et al., 2020) model is trained to reconstruct a masked reference summary from the corresponding document with noun phrases and entities randomly masked. Then, using this BART model, negative summaries are generated from an unseen, masked reference summary, with or without the corresponding document masked. The idea is that with the salient information masked, the trained model can only guess, if not make up, to fill masks in the masked summary and thus result in a strongly inconsistent summary. Potential limitations: Only noun phrases and entities are masked out whereas inconsistencies may also occur in other parts of a text, e.g. a whole clause.

SummaC (Laban et al., 2022) does not synthesize data itself but employs models trained on NLI (Natural Language Inference) datasets, which contain human-written hypotheses that are entailing, neutral, or contradictory to individual claims. NLI is similar to inconsistency detection in the sense that an inconsistent summary is not entailed by the document. Potential limitations: Human-crafted hypotheses for training NLI models may exhibit different characteristics than those of the machine-generated summaries. In addition, SummaC works at the granularity of individual sentences whereas inconsistencies are often cross-sentence.

2.2 The Inconsistencies Undetected by the SOTA: A case study

The analysis above indicates a potential gap between inconsistencies synthesized using SOTA and the actual inconsistencies exhibited by neural network-based summarizers. Here we quantitatively and qualitatively verify the gap on real data. Using the test sets of the SummaC benchmark, a widely used benchmark bearing the same name of an aforementioned method, we examine the false positive (inconsistent by predicted otherwise) samples predicted by two best-performing approaches on the SummaC benchmark: MFMA (Lee et al., 2022b) and SummaC-Conv (Laban et al., 2022), the latter of which is superior than SummaC-ZS, the other version of SummaC. FactCC (Kryscinski et al., 2020) is not covered here because it is outperformed by MFMA and SummaC-Conv on the SummaC benchmark.

The SummaC benchmark comprises six summary factual consistency datasets: CoGenSumm (Falke et al., 2019), FactCC (Kryscinski et al., 2020), Frank (Pagnoni et al., 2021), Polytope (Huang et al., 2020), SummEval (Fabbri et al., 2021) and XSumFaith (Maynez et al., 2020). These six datasets contain a) summaries generated using various summarizers and b) human annotation to whether each summary is consistent to its corresponding document. Documents in CoGenSumm, FactCC, SummEval, and Polytope come from the famous CNN/Dailymail dataset whereas documents in XSumFaith come from the XSum dataset. Frank has documents from both CNN/Dailymail and XSum, denoted as Frank-CNNDM and Frank-XSum respectively thereafter.

Taxonomy of Factual Inconsistencies. We are very interested in the performance of SOTA approaches on different types of factual inconsistencies. Among of the six datasets of the SummaC benchmark, three of them provide subcategories for factual inconsistencies:

- **XSumFaith** has 2 subcategories: Extrinsic and Intrinsic.
- **Polytope** has 5 subcategories: Addition, Omission, Inaccuracy Intrinsic, Inaccuracy Extrinsic and Positive-Negative Aspect.
- **Frank** has 8 subcategories: Predicate Error (RelE), Entity Error (EntE), Circumstance Error (CircE), Coreference Error (CorefE), Discourse Link Error (LinkE), Out of Article Error (OutE), Grammatical Error (GramE) and Other Error (OtherE).

The divided taxonomies used by different datasets make a unified analysis difficult. Here, we borrow the taxonomy from Frank's eight subcategories because Frank has the finest granularity. This also limits the discussion in this section to Frank, excluding the rest five datasets. We will use data from all six datasets later in the experiments (Section 4).

Quantitative Study. We first examine the error rate of MFMA and SummaC-Conv on Frank's test set for each subcategory of inconsistencies. The error rate is calculated as:

$$Error\ Rate = \frac{FP}{N}$$

where FP and N are the number of false positive samples and the number of total samples, respectively, in the subcategory.

The error rates of MFMA and SummaC-Conv are given in Table 4 along with other experimental results to be discussed later. Coreference errors (CorefE) and discourse link errors (LinkE) are the two most difficult subcategories of inconsistencies for SOTA approaches where they perform even worse than random guess which has a 50% accuracy. MFMA has error rates of 67.9% and 66.7% on CorefE and LinkE, respectively. SummaC-Conv has error rates of 67.9% and 57.1% on CorefE and LinkE, respectively. Both approaches have <32% error rates on other factual inconsistency subcategories excluding the Other Error subcategory.

Qualitative Study. Next, we qualitatively examine four samples (Table 1) falsely detected as positive (consistent) by both MFMA and SummaC-Conv to show that existing synthesizing methods are really difficult in mimicking inconsistencies produced by modern summarizers. We focus on the two most difficult subcategories, coreference errors and discourse link errors.

A coreference error occurs when a pronoun refers to the wrong object. The first two examples in Table 1 presents coreference errors. It would be difficult for simple heuristics like pronoun swapping in FactCC or pronoun masking in MFMA to mimic such a kind of inconsistency errors. In either of the two examples, the same pronoun ("he" in Example 1 or "him" in Example 2 in Table 1) will be interpreted differently in the document and in the summary due to the information of the true referent is missing in the summary.

A discourse error occurs when two statements are mixed. It can happen when summarizing either a single sentence (Example 3, Table 1) or a plurality of sentences (Example 4, Table 1). In Example 3, the inconsistent summary fuses "goldfish" with information about "koi carp" which is men-

ID	Document sentence(s)	Inconsistent summary	Explanation
1	*Mr Katter said the Government believes Mr Gordon would quit after he was recently accused of domestic violence.*	*Mr Katter said he would quit after he was accused of domestic violence.*	Coreference error: "he" in the summary will be misinterpreted as "Mr Katter" while it actually should refer to "Mr. Gordon".
2	*Barcelona club president Josep Maria Bartomeu has insisted that the La Liga leaders have no plans to replace Luis Enrique and they're 'very happy' with him.*	*Barcelona club president Josep Maria Bartomeu says the La Liga leaders are very happy with him.*	Coreference error: "him" in the summary will be misinterpeated as "Josep Maria Bartomeu" while it actually should refer to "Luis Enrique".
3	*Goldfish are being caught weighing up to 2kg and koi carp up to 8kg and one metre in length.*	*Goldfish are being caught weighing up to 8kg and one metre in length.*	Discourse error: the summary attaches the statement for "koi carp" mistakenly to "Goldfish".
4	*Paul Merson had another dig at Andros Townsend after his appearance for Tottenham against Burnley ...Townsend hit back at Merson on Twitter after scoring for England against Italy.*	*Paul Merson had another dig at andros townsend after scoring for England against Italy.*	Discourse error: the summary concatenates an event later in the document to a previous statement.

Table 1: Examples failed to be detected by SOTA factuality classifiers. Related contents are in the same color.

tioned in the second half of the source sentence. In Example 4, the summary mistakenly mixes two statements about two persons from two sentences of the document. However, introducing discourse errors by fusing statements has not been touched by current synthesis methods, and we speculate that it would be difficult to do in current methods which manipulate individual tokens. In addition, existing NLI datasets usually contain only single-sentence statements and thus are incapable of mimicking multi-sentence discourse errors.

It's also worthy noting that for all the examples in Table 1, the summary is or almost is the concatenation of sub-strings from the document. This is probably because, according to the training data, certain summarization models have learned to copy phrases from the document and stitch them into a summary. Because it is difficult to predict the behavior of neural network-based summarizers, it is difficult to come up with heuristics to mimic factual inconsistencies they may exhibit.

The intractability of synthesizing inconsistency summaries. According to the discussion above, there is a gap between the inconsistencies created by current data synthesis methods and the actual inconsistencies exhibited by neural network-based summarizers. We could iteratively add data synthesis heuristics, including those using generative LLMs, after examining falsely classified samples. However, due to the potential diversity of factual inconsistency, this "accident-and-patch" strategy requiring recurring manual effort may not be scalable. On top of that, some types of errors, such as discourse errors, are hard to be defined. Therefore, in this paper, we take another avenue by directly finetuning on existing but limited human annotations.

3 FactFT: Inconsistency Detection Using Machine-Generated Summaries with Human Annotations

Given a source document $D = [d_0, d_1, \ldots]$ and a machine-generated summary $S = [s_0, s_1, \ldots]$, where d_i or s_i is a sentence, a factual consistency detector is a binary classifier predicting whether the summary is factually consistent with the document, i.e., $f(D, S) \in \{0, 1\}$ where 0 and 1 represent inconsistent (negative) and consistent (positive). Realizing the difficulty to cover the diverse errors synthetically (Section 2), we directly train a factual consistency classifier using an NLI model as the foundation and the currently available but limited machine-generated summaries with human annotations as the training data. The recent advances in parameter-efficient finetuning (PEFT) has made this approach feasible.

3.1 Preprocessing

Instead of feeding the whole document D into the classifier f, we select the document sentences that are most relevant to the summary and feed such sentences to the classifier, i.e., our model predicts $f(D', S)$ where $D' \subseteq D$ instead of $f(D, s)$. Adapting from an approach used by Balachandran et al., 2022, for each summary sentence s_i, only the document sentence d_j that is most relevant to it and its two preceding and two succeeding sentences in the document, namely $d_{j-2}, d_{j-1}, d_{j+1}$ and d_{j+2}

Dataset	Validation Split			Test Split	
	# of samples		% Positive	# of Samples	% Positive
	Before filtering	After filtering			
CoGenSumm	1281	1281	49.7	400	78.0
FactCC	931	886	86.6	503	87.7
Frank	671	444	45.0	1575	33.6
-*CNNDM*	375	360	54.2	875	56.3
-*XSum*	296	84	6.0	700	5.1
SummEval	850	0	N/A	850	90.6
Polytope	634	201	5.9	634	6.5
XSumFaith	1250	45	6.7	1250	10.4

Table 2: Statistics of the training and test data. Validation split is used for training.

which provide the context, are included into D'. By filtering out less irrelevant information from the document, the NLI model can benefit from a relatively similar input length of the text pair. In addition, this saves the limited input length set by the Transformer models.

3.2 Parameter Efficient Fine-Tuning

The major concern when fine-tuning with a limited amount of data is that the model can be prone to overfitting. One reason is that the number of trainable parameters is relatively large compared with the number of samples. This is a major reason that previous SOTA uses synthetic data for training. Emerged recently, parameter Efficient Fine-Tuning (PEFT) methods address this issue by freezing most parameters of a large language model and only fine-tuning a small number of additional parameters. Such an approach has been shown to perform better (Pu et al., 2023) than full finetuning in low-data and out-of-domain scenarios. We employ one of the most famous PEFT methods, LoRA (Hu et al., 2021), in this paper. LoRA appends two smaller matrices to the original model through low-rank decomposition, while the original weight matrix is frozen for further adjustment. With LoRA, our inconsistent classifier finetuned on only 0.14% parameters of an NLI model can achieve SOTA performance using only a few thousand samples.

4 Experiments

4.1 Training and Testing Data

We use the validation sets of the SummaC benchmark (Laban et al., 2022) as the training data. Among the six datasets in SummaC benchmark, CoGenSumm, FactCC, and Frank come with original validation splits. For the rest three datasets, SummaC splits the validation set by the parity of sample index.

Because the six datasets are all sampled from the CNN/DailyMail (See et al., 2017) or XSum (Narayan et al., 2018) dataset, to ensure no data leakage, we filter out the samples in any validation set that share a document with any test set. The statistics of the validation and test sets are shown in Table 2. Note that the Polytope and XSumFaith datasets are extremely negatively skewed.

We perform a stratified k-fold validation with non-overlapping groups where samples from the same document always belong to one group to prevent data leakage. The best model for each fold is found using the test split in the cross validation. Finally, we report the average performance from the k folds on each of the six test sets of SummaC.

4.2 Settings

Given the SOTA results achieved by SummaC, we select a similar NLI model for finetuning. The `DeBERTa-v2-xlarge-mnli` (He et al., 2021) model hosted on HuggingFace is used as the base model. We use HuggingFace's `peft` (Mangrulkar et al., 2022) library to apply LoRA. For LoRA settings, following the experience of Hu et al., 2021, we add the low rank update matrices only to the query and value module in every self-attention layer with rank $r_q = r_v = 8$, and LoRA scaling factor $\alpha = 8$. The dropout probability of the LoRA layers is 0.1. Under these settings, 1.3M parameters which are 0.14% of the total 0.9B parameters of `DeBERTa-v2-xlarge-mnli` are trainable. The training process has a learning rate of 5e-5, using the paged 8-bit AdamW optimizer with a linear scheduler. Fold number $k = 5$, the number of training epochs is set to 10, and the model is validated for every 400 steps for identifying the best performing model. The training process can be done on a single consumer-level NVIDIA RTX 3090 GPU with tf32 precision and a batch size of 5.

Model Type	Methods	Test Sets in SummaC Benchmark						
		CoGenSum	FactCC	Frank	SummEval	Polytope	XSumFaith	Overall
Other	NER Overlap	53.0	55.0	60.9	56.8	52.0	63.3	56.8
Parsing	DAE	63.4	75.9	61.7	70.3	62.8	50.8	64.2
QAG	FEQA	61.0	53.6	69.9	53.8	57.8	56.0	58.7
	QuestEval	62.6	66.6	82.1	72.5	**70.3**	62.1	69.4
LLM	ChatGPT-ZS	63.3	74.7	80.9	76.5	56.9	64.7	69.5
	ChatGPT-ZS-COT	74.3	79.5	82.6	83.3	61.4	63.1	74.0
NLI	MNLI-doc	57.6	61.3	63.6	66.6	61.0	57.5	61.3
	SummaC-ZS	70.4	83.8	79.0	78.7	62.0	58.4	72.1
	SummaC-Conv	64.7	89.5	81.6	81.7	62.7	**66.4**	74.4
	SENTLI	79.3	89.5	82.1	77.2	52.4	59.3	73.3
	-RerankSoft	79.6	86.1	80.4	78.5	52.8	62.7	73.4
	-RerankHard	80.5	83.3	78.4	79.9	55.1	64.2	73.6
Classifier	FactCC-CLS	63.1	75.9	59.4	60.1	61.0	57.6	62.9
	MFMA	64.6	84.5	81.3	75.5	58.0	53.6	69.6
	FactFT	**82.3±1.5****	**91.0±1.5****	**87.1±1.8****	**85.7±0.5****	51.0±1.8	57.7±2.1	**75.8****

Table 3: Balanced Accuracy (%) on the SummaC benchmark. Best on each dataset in bold. The notation ** indicates 99% confidence in our approach FactFT over SummaC and MFMA, the two strongest baselines. Significance tests for SENTNLI & ChatGPT are excluded due to code/data/model reproducibility. Our FactFT results present as the k-fold mean ± the standard deviation.

4.3 Baselines

We post the baseline metrics evaluated by SummaC in the Table 3: NER Overlap (Laban et al., 2021), MNLI-doc (Zhuang et al., 2021), FactCC (Kryscinski et al., 2020), DAE (Goyal and Durrett, 2020), FEQA (Wang et al., 2020), QuestEval (Scialom et al., 2021) and SummaC (Laban et al., 2022). In addition, SEN**TLI**(Schuster et al., 2022) is included as another strong NLI baseline. We also rerun MFMA (Lee et al., 2022b) on the SummaC benchmark because it is currently the best performing metric using rule-generated negative samples known to us. ChatGPT (Luo et al., 2023) (gpt-3.5-turbo-0301) as a fact inconsistency evaluator is also treated as a baseline and its performances are included in Table 3.

4.4 Results and Discussion

4.4.1 Balanced Accuracy

Balanced Accuracy is used to measure the performance on the benchmark due to the varying class imbalance of the 6 test sets. It is calculated as follows:

$$BAcc = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$$

where TP, FP, TN, and FN are the numbers of samples that are true positive, false positive, true negative, and the false negative respectively.

The full Balanced Accuracy results can be seen in Table 3. The overall performance is calculated as the macro average of all test sets. Our approach has the best overall performance and is best-performing on four out of the six datasets. In particular, it outperforms ChatGPT with chain of thought (COT) prompts by 8.00, 4.50, 2.36 percentage points on the CoGenSumm, Frank, and SummEval datasets, correspondingly. Our model exhibits a relatively low performance on the extremely negatively skewed XSumFaith and Polytope datasets. We attribute this to the extreme imbalance in the two datasets.

4.4.2 FPR and FNR

Figure 1 shows a more detail analysis on the False Positive Rates (FPRs) and False Negative Rates (FNRs) of our approach and MFMA and SummaC-Conv, two best-performing baselines on the SummaC benchmark. Measuring the ratio of inconsistent summaries missed, the FPR is calculated as:

$$FPR = \frac{FP}{FP+TN}.$$

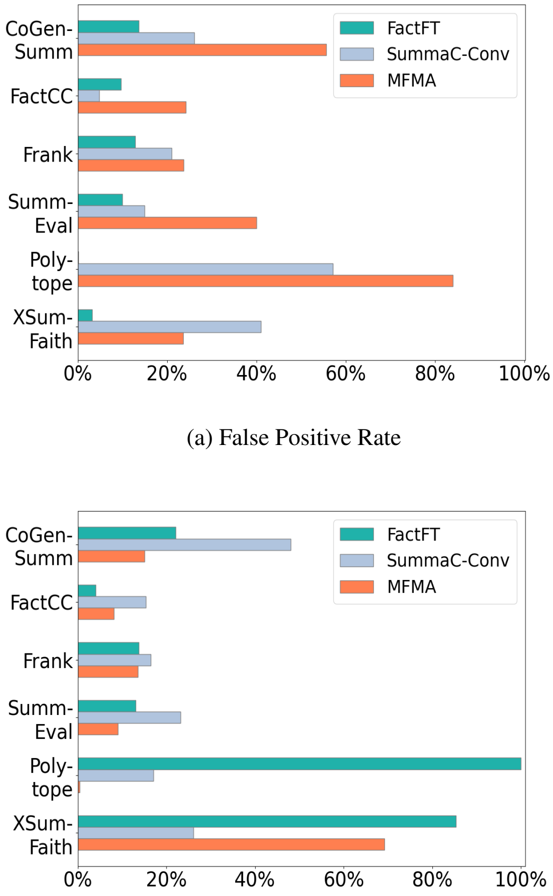
Measuring the ratio of false alarms, the FNR is calculated as:

$$FNR = \frac{FN}{FN+TP}.$$

Our approach FactFT has the lowest FPR on all datasets except for FactCC (where it is the second best), indicating that finetuning on human-annotated data indeed expands the model's ability

	CorefE	LinkE	GramE	EntE	CircE	RelE	OutE	OtherE
SummaC-Conv	67.9	57.1	31.6	23.4	15.5	18.1	**2.4**	75.0
MFMA	67.9	66.7	30.6	20.6	20.0	21.9	9.6	87.5
FactFT	**51.9**	**47.6**	**23.5**	**7.8**	**10.9**	**6.7**	2.7	**62.5**

Table 4: Per-category error rate (%) of three approaches on Frank's test set.



(a) False Positive Rate

(b) False Negative Rate

Figure 1: False Positive Rates and False Negative Rates on six datasets. The lower the better.

to detect more inconsistency errors. In the meantime, our approach has the second lowest FNR on four out of the six datasets, behind MFMA.

The relatively high FNR of our approach on the XSumFaith dataset is potentially due to a substantially lower proportion of training data from XSum than CNN/DailyMail. The low positive rate in the XSum data makes the classifier further leaning towards negative prediction. The high FNR on the Polytope dataset may be due to the annotation protocol used by Polytope that are quite different from protocols used in other CNN/DailyMail based datasets. As a result, our model fails to recognize the few consistent samples in Polytope.

4.4.3 Categorical Error Rate

In Table 4, we further examine the error rate of our approach on each inconsistency subcategory labeled in the Frank test set. Compared to MFMA and SummaC-Conv, FactFT has achieved lower error rate on almost every factual error type except out-of-article errors (OutE). This supports the importance of machine-generated summaries with human annotations that they contain more inconsistency patterns than data synthesized by SOTA on nearly any category of inconsistencies. On the two major inconsistency types that are difficult to detect, CorefE and LinkE, FactFT lowers the error rate by 16.0 and 9.5 percentage points respectively with respect to the best of MFMA and SummaC-Conv.

4.4.4 Ablation Study: Cross-Dataset

In the previous experiments, the validation sets of all datasets in the SummaC benchmark are used as the training data. Here we study the cross-dataset robustness of our approach in a leave-one-group-out cross validation: in each fold, training a model using validation sets of five datasets in the SummaC benchmark and testing the model on the test set of the remaining dataset. We denote results obtained so as FactFT-Cross.

In Table 5 (the row **w/ cross dataset training**), we compared the balanced accuracy between the original FactFT and FactFT under the cross-dataset setting (referred to as FactFT-Cross). FactFT-Cross has a minor performance drop on CoGenSumm, but it still outperforms all baselines. The performance drop on FactCC, Frank, and SummEval is very marginal. Interestingly, FactFT-Cross gains performance on Polytope and XSumFaith, probably because of in-domain validation. For XSumFaith, k-fold cross validation can dilute the samples from BBC/XSum due to CNN/DM is the major source for most of the datasets, while leave-one-group-out retains all samples for validation. For Polytope, the in-domain validation is beneficial because of its unique annotation protocol mentioned earlier. The

	CoGenSum	FactCC	Frank	SummEval	Polytope	XSumFaith	Overall
FactFT	**82.3**	**91.0**	**87.1**	**85.7**	51.0	57.7	75.8
Ablation Settings							
w/ cross dataset training	77.4	89.1	86.8	85.6	57.9	63.1	**76.7**
w/o irrelevance filtering	81.4	86.7	85.1	84.6	53.6	59.6	75.2
using FactCC synthetic data only	78.0	89.3	78.2	74.2	**60.9**	**66.0**	74.4

Table 5: Balanced accuracy(%) for three ablation settings.

performance improvement on Polytope and XSumFaith also results in a slight overall performance improvement.

4.4.5 Ablation Study: Irrelevance Filtering

In the preprocessing stage, we first retrieve the document sentences highly similar to the summary and then only feed those sentences with some context sentences to the NLI model. To understand the effect of the preprocessing step, we re-evaluated FactFT without filtering out irrelevant sentences. According to Table 5 (the row **w/o irrelevance filtering**), skipping irrelevance filtering will cause a slight performance drop on 4 out of the 6 test sets. We believe that irrelevance filtering helps the model avoid exceeding token limits when evaluating with a longer context.

4.4.6 Ablation Study: Real vs. Synthetic Data

Due to the various foundation models used in baselines in Table 3, it is difficult to perform a fair comparison between different metrics. Thus, in this ablation setting, using the same foundation model, we explore the effect of training with real machine-generated summaries versus synthetic data. In Table 5 (the row **using FactCC synthetic data only**), we show the performance of `DeBERTa-v2-xlarge-mnli` finetuned with LoRA using FactCC's synthetic data. Despite trained with much more data than FactFT (millions vs. thousands), it was outperformed by FactFT, whose training data is real machine-generated summaries, on 4 out of 6 data sets. This shows the important of real data and echos the intractability of synthesizing factual inconsistencies.

5 Related Work

Categories of Factual Inconsistencies. According to Maynez et al. (Maynez et al., 2020), factual inconsistencies made by summarization systems can be categorized into two types: *intrinsic errors* and *extrinsic errors*. Intrinsic errors refer to content that is hallucinated using the material from the source document, while extrinsic errors occur when the summarizer model generates content that is irrelevant to the source material. It has also been discovered (Maynez et al., 2020; Kryscinski et al., 2020) that abstractive summarizers often use forged entities.

Relevant Evidence Discovery. The widely used summarization metric ROUGE (Lin, 2004) has been reported (Fabbri et al., 2021) to have low correlation with consistency annotations but high correlation in terms of relevance. As a result, some post-editing methods (Lee et al., 2022a; Balachandran et al., 2022) have adopted ROUGE to extract the most relevant sentences in the document related to a summary, aiming to correct inconsistent summaries. In our work, we adopt this idea of relevance checking to bridge the gap between the unmatched input granularity (sentence-level to document-level) of the NLI model and save input length.

Measuring the Factuality. Significant efforts have been made recently to automatically evaluate the factual consistency of abstractive summarization. Based on the category proposed in (Koh et al., 2022), current methods can be divided into two groups: QA-based and entailment classification methods. QA-based methods evaluate factual consistency using QA frameworks. These approaches (Wang et al., 2020; Scialom et al., 2021; Durmus et al., 2020) first generate questions based on given summaries and answer questions conditioning on source documents and summaries. A summary is considered consistent if the answers based on source text and summaries match. These methods are reference-free and more correlated to human judgments, but they suffer from complex computations and error propagation. Entailment classification approaches (Kryscinski et al., 2020; Yin et al., 2021; Lee et al., 2022b; Utama et al., 2022; Soleimani et al., 2023) mainly construct synthetic datasets by corrupting sentences from the source document or reference summary to create negative samples and then train classifiers by contrastive learning. Among them, Falsesum (Utama et al., 2022) and NonFactS (Soleimani et al., 2023) are similar methods to MFMA (Lee et al., 2022b), as they all use masked language model to generate inconsistencies intentionally. SummaC (Laban et al., 2022) breaks the summary into small pieces and perform the evaluation on sentence or phrase level using NLI models. Other than classifying based on plain text, FactGraph (Ribeiro et al., 2022) builds a consistency classifier upon the semantic graph structural representation of the texts, and FineGrainFact (Chan et al., 2023) enhances text input with semantic role labeling. In this work, we focus on the drawbacks of the entailment based methods with plain text as input and propose to improve such methods.

6 Conclusion

To identify directions to improve the detection accuracy of summary factual consistency, we begin this study by examining the inconsistency synthesis methods used in SOTA summarization consistency detectors, both theoretically and empirically. We find that coreference errors and discourse errors are the two most difficult types of factual errors missed by SOTA consistency detectors trained with synthetic data because existing methods to synthesize inconsistencies may fail to produce them.

Realizing the diversity of inconsistencies and the challenges to mimic them by manually designed synthesis heuristics, we propose to use limited but actual machine-generated summaries with human annotation to parameter-efficiently finetune an NLI model of 0.9B parameters. The finetuned classifier outperforms SOTA on four datasets. This finding highlights the importance of using real machine-generated texts for building metrics for NLG. We hope our effort can encourage the community to build more and better summarization consistency datasets with unified taxonomy.

Acknowledgment

This work is partially supported by NSF grants CNS-1817089 and CNS-2141153.

Limitations

In Section 3.1, our model uses ROUGE to discover the most relevant sentences in the document with a given summary. When the abstraction level becomes very high, or the summary is very short, the ROUGE metric may fail to retrieve the related evidences. One can use the whole document as input, but the long document may hit the token length limit set by the transformer model. Instead, we can use a sentence similarity model with a relatively slower processing speed.

With limited human annotations, we have successfully mitigated the false positive rate of the classifier. However, there are still some hard examples. Our model can direct benefit from more human annotations. Meanwhile, inconsistency annotation is laborious and skill-demanding. We hope to explore more on improving the annotation protocol and reducing the cost for such NLG evaluation tasks.

Another limitation worth mentioning is the domain transferability. Our model performs better on CNN/DailyMail-based datasets than on XSum-based datasets. The large proportion of the CNN/DailyMail samples in the training data made the classifier weak on classifying XSum test sets. We seek better parameter efficient methods to enable better cross domain testing performance.

References

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9818–9830, Abu Dhabi, United Arab Emirates. 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.

Hou Pong Chan, Qi Zeng, and Heng Ji. 2023. Interpretable automatic fine-grained inconsistency detection in text summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6433–6444, Toronto, Canada. Association for Computational Linguistics.

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.

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

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

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Comput. Surv.*, 55(8).

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.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. 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 6365–6378, Online. Association for Computational Linguistics.

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

Hwanhee Lee, Cheoneum Park, Seunghyun Yoon, Trung Bui, Franck Dernoncourt, Juae Kim, and Kyomin Jung. 2022a. Factual error correction for abstractive summaries using entity retrieval. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 439–444, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2022b. Masked summarization to generate factually inconsistent summaries for improved factual consistency checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1019–1030, Seattle, United States. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Sourab Mangrulkar, S Gugger, L Debut, Y Belkada, and S Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

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.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

George Pu, Anirudh Jain, Jihan Yin, and Russell Kaplan. 2023. Empirical analysis of the strengths and weaknesses of peft techniques for llms.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

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 *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 394–412, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

Amir Soleimani, Christof Monz, and Marcel Worring. 2023. NonFactS: NonFactual summary generation for factuality evaluation in document summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6405–6419, Toronto, Canada. Association for Computational Linguistics.

Prasetya Utama, Joshua Bambrick, Nafise Moosavi, and Iryna Gurevych. 2022. Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. 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.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A ROC-AUC Results

In addition to the Balanced Accuracy, we also include the ROC-AUC results in Table 6. SENTNLI and ChatGPT are excluded due to code/data/model reproducibility.

Model Type	**Methods**	**Test Sets in SummaC Benchmark**						
		CoGenSum	FactCC	Frank	SummEval	Polytope	XSumFaith	Overall
Others	NER Overlap	53.0	53.1	60.9	56.8	51.6	61.7	56.2
Parsing	DAE	67.8	82.7	64.3	77.4	64.1	41.3	65.2
QAG	FEQA	60.8	50.7	74.8	52.2	54.6	53.4	57.8
	QuestEval	64.4	71.5	87.9	79.0	**72.2**	66.4	73.6
NLI	MNLI-doc	59.4	62.1	67.2	70.0	62.6	59.4	63.5
	SummaC-ZS	73.1	83.7	85.3	85.5	60.3	58.0	74.3
	SummaC-Conv	67.6	92.2	88.4	86.0	62.4	**70.2**	77.8
Classifier	FactCC-CLS	65.0	79.6	62.7	61.4	63.5	59.2	65.2
	MFMA	74.9	88.3	86.0	84.0	59.9	55.4	74.8
	FactFT	**88.9****	**96.5****	**92.3****	**91.8****	66.8	64.7	**83.5****

Table 6: ROC-AUC (%) on the SummaC benchmark. The notation ** is for 99% confidence in our approach FactFT over SummaC and MFMA.