Annotating and Modeling Fine-grained Factuality in Summarization

Tanya Goyal and **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
tanyagoyal@utexas.edu, gdurrett@cs.utexas.edu

Abstract

Recent pre-trained abstractive summarization systems have started to achieve credible performance, but a major barrier to their use in practice is their propensity to output summaries that are not faithful to the input and that contain factual errors. While a number of annotated datasets and statistical models for assessing factuality have been explored, there is no clear picture of what errors are most important to target or where current techniques are succeeding and failing. We explore both synthetic and human-labeled data sources for training models to identify factual errors in summarization, and study factuality at the word-, dependency-, and sentence-level. Our observations are threefold. First, exhibited factual errors differ significantly across datasets, and commonly-used training sets of simple synthetic errors do not reflect errors made on abstractive datasets like XSUM. Second, human-labeled data with fine-grained annotations provides a more effective training signal than sentence-level annotations or synthetic data. Finally, we show that our best factuality detection model enables training of more factual XSUM summarization models by allowing us to identify non-factual tokens in the training data.[1]

1 Introduction

Hallucination of unsupported or incorrect facts is a known shortcoming of current text generation and summarization models (Cao et al., 2018; Falke et al., 2019). This has been established for both abstractive summarization models (Maynez et al., 2020) and extractive summarization models (Kryscinski et al., 2020; Falke et al., 2019). Past work has explored using off-the-shelf frameworks such as entailment models (Falke et al., 2019) or QA systems (Durmus et al., 2020; Wang et al., 2020) to detect and sometimes correct errors in generated summaries. Another line of recent work has used synthetically generated data to specifically train models on the factuality detection task (Kryscinski et al., 2020; Zhao et al., 2020; Goyal and Durrett, 2020a). However, these efforts have focused on different datasets, summarization systems, and error types, often shedding little light on what errors state-of-the-art systems are actually making and how to fix them.

In this paper, we aim to answer two main questions. First, while synthetic data generation approaches are specifically designed for factuality evaluation, **do these align with actual errors made by generation models? We find the answer is no:** techniques using surface-level data corruption (Kryscinski et al., 2020; Zhao et al., 2020; Cao et al., 2020) or paraphrasing (Goyal and Durrett, 2020a) target inherently different error distributions than those seen in actual model generations, and factuality models trained on these datasets perform poorly in practice. Furthermore, we show that different summarization domains, CNN/Daily Mail (Hermann et al., 2015; Nallapati et al., 2016) and XSum (Narayan et al., 2018) (which differ in the style of summaries and degree of abstraction), exhibit substantially different error distributions in generated summaries, and the same dataset creation approach cannot be used across the board.

Second, we investigate the best approach for modeling and learning factuality, particularly for highly abstractive summarization settings (Narayan et al., 2018). Specifically, we compare the utility of fine-grained human annotations (such as error highlighting at the word- or span-level) with sentence-level factuality annotations. We use a prior factuality detection model capable of leveraging such fine-grained annotations (Goyal and Durrett, 2020a) and show that these allow us to more reliably detect errors as well as localize those errors within generated texts. In fact, fine-grained

[1] Code and data available at https://github.com/tagoyal/factuality-datasets

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1449–1462
June 6–11, 2021.

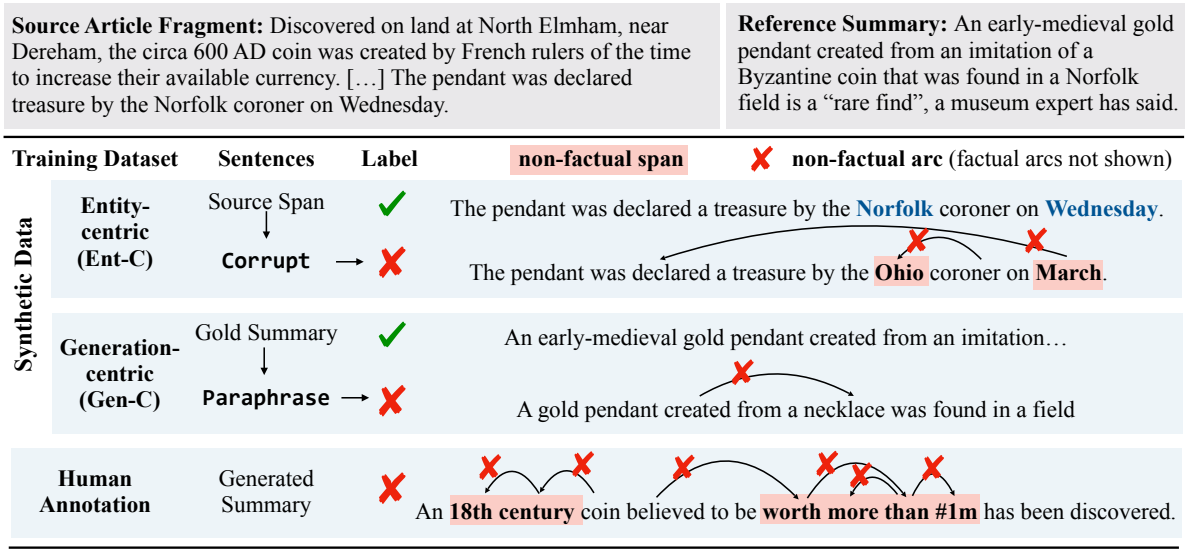

Figure 1: Examples from the synthetic and human annotated factuality datasets. The entity-centric and generation-centric approaches produce bad summaries from processes which can label their errors. All models can be adapted to give word-level, dependency-level, or sentence-level highlights, except for Gen-C.

human annotations are almost essential for any of our techniques to work well with high-performing summarizers in the challenging XSUM setting.

Finally, we demonstrate a practical application for such error localization capabilities beyond interpretibility. Given noisy training data for summarization, we employ a modified training objective that leverages information about error spans in gold summaries, derived from factuality models, to train the summarizer. Our results show that models trained using this approach are inherently more factual than standard training objectives when dealing with error-prone gold datasets.

2 Training Datasets to Compare

We first seek to answer how well synthetic training data can help address factuality errors observed in real summarization datasets. Figure 1 shows a summary of the approaches we consider, which we describe in detail in Section 2.1 and 2.2.

The summarization models we analyse are trained on two English-language domains: (1) **XSUM**, an "extreme" summarization dataset from British Broadcasting Corporation (BBC) articles, where the first sentence of the article is treated as a summary of the article. These summaries are highly abstractive in nature: summarization models trained on this dataset have to learn to model long-range dependencies and may still be unable to recover all information in the gold summary. (2) **CNN/DAILYMAIL**, a multi-sentence abstractive summary dataset. The level of abstraction in this dataset is considerably lower and reference summaries exhibit high overlap with source articles (Zhang et al., 2018).

For both of these domains, we compare the distribution of factuality errors from *synthetic training data* with the distribution of *observed factuality errors* from models trained on that data. In Section 4, we further dive into factuality models' performance in these settings.

2.1 Entity-centric Synthetic Data (Ent-C)

A recent thread of work has focused on leveraging synthetic data transformations for evaluating factuality (Kryscinski et al., 2020), imposing decoding-time constraints (Zhao et al., 2020), or post-correction of summaries (Cao et al., 2020). Each of these approaches assumes that corruption strategies will yield useful non-factual summaries, while gold summaries are treated as factual. Figure 1 illustrates this process: these approaches apply transformations to either the source article (shown) or a reference summary to obtain a corrupted summary (*Ohio* instead of *Norfolk*).

We call this set of approaches **entity-centric** because the transformations largely focus on perturbing entities and noun phrases and addressing these types of hallucinations. The approach from Kryscinski et al. (2020) has the broadest set of transformations out of this line of prior work, so we follow them to generate training examples representative of this class of techniques. The data

Gold: Apple has been accused of misleading customers over its new iPad 3.0 version.	
Entity Swap	Apple has been accused of … over its new **iPhone** 3.0 version.
Number Swap	Apple has been accused of … over its new iPhone **10** version.
Pronoun Swap	Apple has been accused of … over **her** new iPhone version.
Negation	Apple has **not** been accused of … over its new iPhone version.
Noise Injection	Apple has **has** been accused of … over its ~~**new**~~ iPhone version.
Paraphrase	Customers have accused Apple of misinformation over its new iPad 3.0 version.

Figure 2: Set of transformations/data corruption techniques from Kryscinski et al. (2020) used to generate training data for the entity-centric approach.

corruptions or transformations included are entity and number swapping, pronoun swapping, sentence negation, and arbitrary noise injection. Additionally, backtranslation is used to paraphrase summaries and further augment the dataset. Figure 2 illustrates the complete set of transformations applied to the reference summary to construct the synthetic dataset.

For CNN/DM, we use a dataset of 50k labeled pairs that is a subset of the data distributed by Kryscinski et al. (2020); this subset is sufficient to reproduce the performance of their factuality classifier. We generate a similarly-sized dataset for XSUM. Note that although the data creation procedure produces sentence-level annotations, since data corruptions are introduced in a rule-based manner, we can highlights spans within the summaries where the error was actually introduced to get span-level factuality annotations as well. Figure 1 illustrates these spans in red. The figure also demonstrates how to obtain dependency-level factuality judgements from error span highlights; what these mean and how these are derived is explained in Section 2.2.

2.2 Generation-centric Synthetic Data (Gen-C)

Goyal and Durrett (2020a) introduce a different method for obtaining factuality annotations that more closely align with errors made by generation models. The core assumption of that **generation-centric** approach (see Figure 1) is that generated paraphrases at the bottom of a paraphrasing model's beam (the 10th-best paraphrase) are more likely to contain factual errors than 1-best generations, and new information in these generations can be labeled non-factual. Moreover, these generations align with realistic errors made by generation models, unlike purely synthetic entity swaps. In addition to sentence-level annotations, this approach also extracts **factuality labels corresponding to each dependency arc of the generated summary**. According to the definition given in Goyal and Durrett (2020a), an arc is factual (or entailed) if the semantic relationship described by that particular dependency arc is entailed by the source article. Figure 1 shows a non-factual *created* → *necklace* collapsed dependency arc.

To adapt this data creation approach for our current experimental setting, we generated paraphrases of gold summaries using the paraphrase generation model of Goyal and Durrett (2020b). We use the 10th-best generated summaries to generate both sentence-level and dependency-level annotations automatically. See Figure 1 for an example of this process. We generate 40k training examples for both CNN/DM and XSUM domains.

2.3 Types of supervision

The two techniques, Ent-C and Gen-C, naturally generate annotations at different levels. We take steps to unify these formats to enable apples-to-apples comparison of them.

For Ent-C as well as human-labeled data (discussed later), we have access to span highlights within the summary that are non-factual with respect to the source article. From these, we can derive dependency-level annotations in the following way: for each arc in the summary, if either the head word or the child word is highlighted as non-factual, the dependency arc is annotated as non-factual. Otherwise, the arc is factual. This process is demonstrated in Figure 1.

Table 1 gives a summary of the type of annotations available for the 3 types of training datasets. Mapping Gen-C dependency-level annotations to word-level classification decisions is less well-defined, so we do not attempt to do this. Our focus in this work will be on training sentence-level and dependency-level classification models, which is possible on all our sources of data.

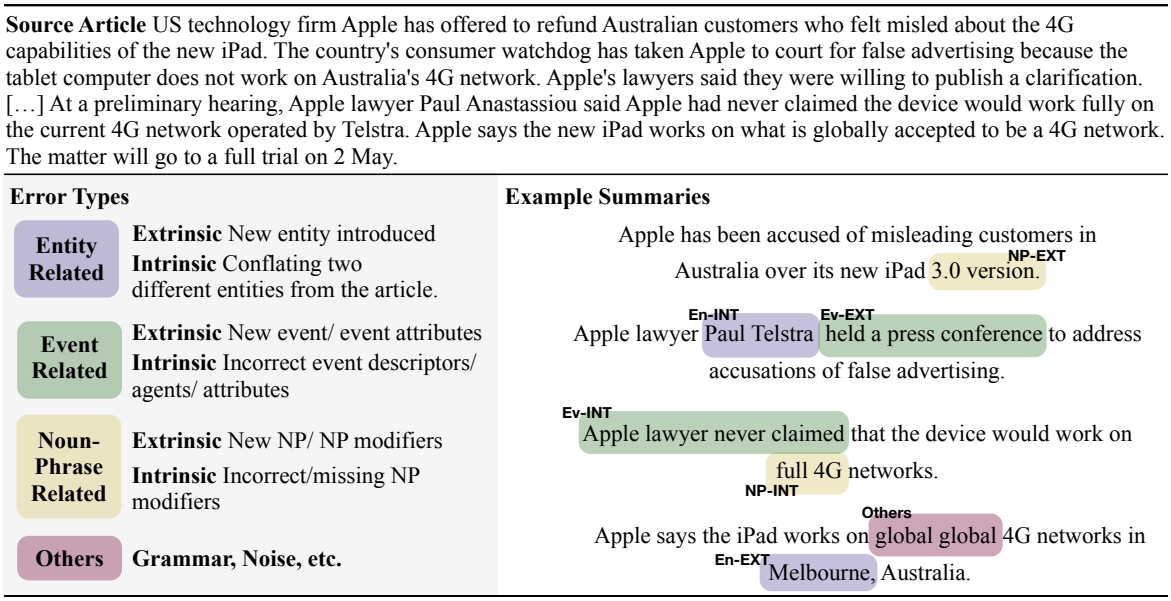

Figure 3: Taxonomy of error types considered in our manual annotation. On the right are example summaries with highlighted spans corresponding to the error types; the first summary is an actual BART generated summary while others are manually constructed representative examples.

Dataset Source	Sent-level	Word-level	Dep-level
Ent-C	✓	✓	✓^{d}
Gen-C	✓		✓
HUMAN-XSUM	✓	✓	✓^{d}

Table 1: Summary of the annotations available for each training dataset. ✓ indicates that annotations at that granularity can be directly obtained from the data creation process. ✓^{d} indicates that annotations can be derived.

3 Analysis of Error Types

Past work using synthetic training data implicitly assumes that training a factuality model on such data will allow it to transfer to realistic settings. We start by qualitatively analyzing the actual errors produced by summarization models to see how these align with the synthetic data, which helps us better understand this assumption.

We identify four broad categories of errors (see Figure 3) that we will identify through manual inspection. Each of these categories is further divided into **Intrinsic** (errors that arise as a result of misinterpreting information from the source article) and **Extrinsic** (errors that hallucinate *new* information or facts not present in the source article), following the characterization from Maynez et al. (2020).

1. **Entity-Related**: errors specifically related to surface realization of named entities, quantities, dates, etc. Hallucination of new entities is an extrinsic error; incorrectly combining distinct entities from the source article is an intrinsic error (*Paul Telstra* in Figure 3).

2. **Event-Related**: errors with incorrect claims about events in the summary, such as predicates with arguments filled by incorrect entities. Hallucinations of new events (*held a press conference* in Figure 3) are extrinsic; mixed-up attributes from within the source article are intrinsic (*apple lawyer never claimed* in Figure 3, incorrect agent).

3. **Noun Phrase-Related**: errors related to noun phrases other than the entity-specific errors. Examples include hallucinating new NP modifiers (extrinsic) or combining with a wrong modifier from the article (intrinsic).

4. **Other Errors**: errors such as ungrammatical text, repeated words, highly erroneous spans, etc. that don't fall into one of the above categories. These are not broken down by intrinsic/extrinsic.

Our taxonomy of summarization errors differs from that of Lux et al. (2020): theirs is targeted at the effects on the reader, whereas ours is more directly tied to the grammatical role of the error, which we believe is more useful to improve our

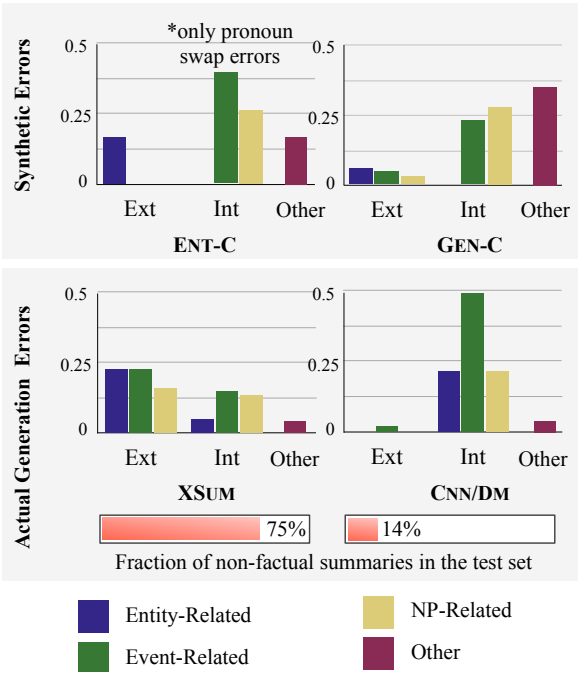

Figure 4: Fractions of examples in each dataset exhibiting different error types (note a single example may have multiple errors). The graphs show a significant mismatch between the error distributions of actual generation models and synthetic data corruptions.

data and our systems. We use the above taxonomy to annotate examples from both summarization domains. For XSUM, we use the state-of-the-art BART model (Lewis et al., 2020) to generate summaries followed by manual annotation (100 examples). For CNN/DM, annotation was done on the 50 summaries across 10 different models collected by Kryscinski et al. (2020). We additionally do this annotation for the artificially introduced errors in Ent-C and Gen-C.[2]

Results Figure 4 shows the distribution of errors for these different settings. First, we see that **summarization models from different domains make substantially different types of errors**. Models trained on XSUM learn to hallucinate new content and consequently produce extrinsic errors: 60% of the errors made by BART models are extrinsic. One reason for this is that the XSUM data was automatically constructed and contains gold summaries that are noisy or non-factual (75% of gold summaries, according to Maynez et al. (2020)). In addition to this, the gold summaries are also highly abstractive, and **XSum-trained summarization models learn to combine information from different parts of an article**, leading to models making long-range dependency errors. This misinterpretation of content is largely responsible for the 40% of the errors which are intrinsic.

On the other hand, the CNN/DM summarization datasets contain human written gold summaries and are therefore generally much more reliable. The models trained on this dataset reflects that. Only 14% of the generated summaries contains errors in the CNN/DM validation set from (Kryscinski et al., 2020). Of these 14%, **the bulk of the errors produced are intrinsic errors**, primarily event-related caused by sentence compression or fusion, which is common in this dataset (Lebanoff et al., 2019). For example, *the two Delaware boys are in critical condition at the U.S. Virgin Islands* should instead be *...at the hospital after a trip to the U.S. Virgin Islands.* The generation models rarely makes extrinsic hallucinations, and we observed that these are even less common in recent systems like PEGASUS (Zhang et al., 2020a). This aligns with the findings from prior work analysing summarization models (Fabbri et al., 2021).

Comparing these with synthetic error distributions, we can see that **synthetic datasets do not reflect the error distributions of actual generation models**. To the extent that Ent-C covers intrinsic event-related errors, these are almost exclusively from pronoun swaps. Moreover, because CNN/DM and XSUM feature such different errors, a synthetic dataset inspired by observed errors on one setting is not likely to be effective on the other. Later (in Section 5.1), we provide further evidence of this mismatch for both datasets: models trained on this synthetic data perform poorly when evaluated on actual generation errors. Also, models trained on human annotated XSUM training data do not transfer to the CNN/DM domain.

4 Factuality Models to Compare

Next, we investigate how factuality models trained on these synthetic datasets perform on real generation errors. Given a document D, a factuality model predicts whether all the information in a generated summary S is supported by the source document D.[3] We consider two factuality modeling formulations: (1) a **Sentence-Factuality** model that

[2]Discussion of inter-annotator agreement is included in Appendix A.

[3]Factuality is ill-defined: whether inferences, world knowledge, implicatures, etc. are viewed as factual is not standardized and is dependent on human annotators for each dataset or task. However, existing generation models only rarely exhibit tricky cases along these dimensions.

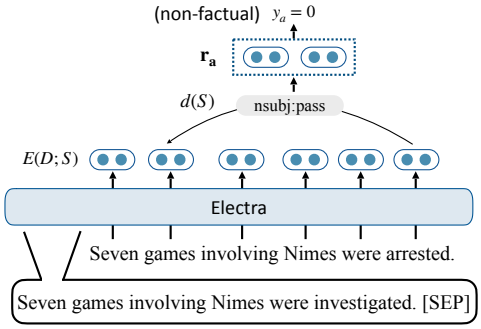

Figure 5: The dependency arc entailment (DAE) model from (Goyal and Durrett, 2020a). A pre-trained encoder is used to obtain arc representations; these are used to predict arc-level factuality decisions.

makes a factuality judgment at the entire summary-level, and (2) an **Arc-Factuality** model (Goyal and Durrett, 2020a) that makes independent factuality judgments for dependency arcs of the generated summary, which are then combined to obtain a sentence-level decision. This helps in localizing factuality errors and was shown to be more effective than sentence-level models in prior work.[4]

4.1 Sentence-Factuality Model

Prior work (Kryscinski et al., 2020) used a BERT-based sequence-pair classification model (Devlin et al., 2019) as follows: the source document D and the generated summary S are concatenated and fed into a pre-trained transformer encoder model (BERT, ELECTRA, etc.). The representation of the [CLS] token is fed into a linear and softmax layer that outputs a probability distribution over the output labels ($y = \{\text{Factual, Non-Factual}\}$). This model can be trained on any data with summary-level factuality labels.

4.2 Arc-Factuality model

The **Dependency Arc Entailment (DAE)** model (Goyal and Durrett, 2020a) evaluates factuality at the dependency arc level. Let $d(S)$ be the dependency-parse of the generated summary S. For each arc $a \in d(S)$, the DAE model predicts whether the relationship described by the arc is entailed by the input document. Note that these factuality judgements are made *independently* for each arc in the summary, and can differ across arcs within the same summary. For instance, in the example in Figure 5, the arc $arrested \leftarrow games$ is non-factual: in context, it is not the case that the games are being arrested. However, the arc $seven \leftarrow games$ is supported by the input (there are seven games) and hence, entailed.

The model architecture is detailed in Figure 5. First, the document D and summary S are concatenated and fed through a pre-trained encoder E. Arc representations $\mathbf{r}_a$ are derived for each dependency arc $a \in d(S)$: $\mathbf{r}_a = [E(D;S)_{a_h}; E(D;S)_{a_c}]$. Here, a_h and a_c correspond to the head and child words of arc a respectively. The arc representation $\mathbf{r}_a$ is fed into a classification layer that outputs a probability distribution over the output labels ($y_a = \{\text{Factual, Non-Factual}\}$). Finally, **summary-level** judgments are extracted from these arc-level decisions: if *any* dependency arc is non-factual, the generated summary is labeled as non-factual.

The DAE model is trained from arc-labeled examples of the form $(D, S, \{y_a\}_{a \in d(S)})$. These are derived from either synthetic or human-labeled data, as described in Section 2.

DAE with weak supervision (DAE-Weak) DAE training requires gold annotations at the dependency-level; however, such fine-grained annotations may not always be available. We extend the DAE framework to address this. The core idea behind our approach is that the sentence-level labels naturally impose loose constraints on the arc-level labels.

The constraints are as follows: for a factual example, *all* individual arcs in the summary must be factual. For a non-factual example, *at least one arc* must be non-factual, and this arc should be one *not* present in the source document. The DAE-Weak model is trained to maximize the marginal likelihood of all labelings that obey these constraints.

Let F be the set of all arcs that should be factual (contains all arcs with sent-label = 1 and arcs common with the source article for sent-label = 0). The above constraints are formulated as the following training objective:

$$\mathcal{L} = \log \left[\prod_{a \in F} P(y_a = 1 \mid D, S) \right] + \log \left[1 - \prod_{a \in D(S) \setminus F} P(y_a = 1 \mid D, S) \right]$$

The second term in the above equation is the probability of predicting at least one non-factual arc in

[4] We describe models for single-sentence summaries to align with the available human-annotated test set (described later in Section 5.1). However, it is straightforward to extend these frameworks for multi-sentence summaries.

	Ent-C	Gen-C
Majority Label	50	50
Kryscinski et al. (2020)	74.1	-
Sent-Factuality	72.3	64.4
DAE	**76.7**	**72.1**
DAE-Weak	75.2	71.1

Table 2: Label-balanced accuracy of factuality models when trained on synthetic factuality training datasets in the CNN/DAILYMAIL domain. Performance is reported on the human-annotated test set from Kryscinski et al. (2020).

a non-factual summary.[5]

5 Experiments

5.1 Evaluation of Synthetic Training Datasets

CNN/DAILYMAIL First, we compare the performance of the three models (Sent-Factuality, DAE and DAE-Weak) trained on the two synthetic factuality datasets (outlined in Section 2) on the CNN/DAILYMAIL domain. We compare their performance on the human-annotated test dataset from Kryscinski et al. (2020). The test set contains human-annotated sentence-level factuality judgements for 503 (article, summary) pairs for summaries generated using 10 different generation models. We use the validation set provided by the authors to choose the best model checkpoint across all settings. Similar to the original paper, we report **class-balanced accuracy values**.

Table 7 outlines our results. The results show that models trained on Ent-C perform slightly better than those trained on Gen-C, but many of the systems are in the same range, with accuracy values of around 75%. However, the reported accuracy values on held-out Ent-C/Gen-C examples are consistently over 90% (results included in Appendix B). This demonstrates that while models trained on these factuality datasets are able to fit the synthetic data distributions well, these are inherently different from actual generation errors. The Appendix also includes graphs of how the human annotated dev set performance varies with training iterations, showing that constant performance on the held-out training set corresponds with highly fluctuating performance on the human annotated data, further indicating that these settings are not identical.

[5]This techniques resembles posterior regularization (Ganchev et al., 2010); however, these constraints are enforced in a hard way on individual examples rather than in expectation at the corpus level. It can also be viewed as an instance of constraint-driven learning (Chang et al., 2007).

Train Data	Majority	Sent-Fact	DAE	DAE-Weak
Ent-C	50	50.9	51.2	53.6
Gen-C	50	54.2	53.0	51.6

Table 3: Performance of factuality models trained on synthetic factuality datasets in the XSUM domain. Label-balanced accuracy is reported on 500 examples from the human-annotated test set from Maynez et al. (2020).

XSUM Next, we similarly evaluate the synthetic datasets and factuality models on the more challenging XSUM domain. Again, we evaluate on a human annotated dataset collected by prior work (Maynez et al., 2020). The dataset contains span highlights indicating hallucinated/incorrect content or information with respect to the source article for 4 different summarization models trained on the XSUM domain (as well as for gold summaries). Figure 1 illustrates this. Similar to prior work, if any word in a summary is marked as hallucinated, we mark the sentence as non-factual. Therefore, for XSUM-HUMAN, the annotation is available at both the sentence-level and span-level.

In total, this dataset contains 2500 (A, S) pairs (along with their factuality labels). We use 500 examples from these to construct our test dataset. The remaining 2000 examples are used to train models, explained in Section 5.2.

Table 3 outlines the results. Unlike on CNN/DM, we see that all models trained on synthetic factuality datasets perform very poorly, achieving close to the majority label baseline. Again, the performance on the held-out synthetic datasets was observed to be very high (see Appendix B). **There is a fundamental difference between the errors that are produced by XSUM summarization models and those introduced by artificial data corruption mechanisms.** Other data that more closely resembles the generation errors is needed to train factuality models in this setting.

5.2 Human Annotated Dataset Evaluation

To investigate whether human annotated data is useful to train factuality models, we train our 3 factuality models on the remaining 2000 human annotated examples from XSUM-HUMAN. In order to train DAE model on this dataset, we use the span highlights to derive dependency-level gold annotations, using the same strategy from 2.3 (illustrated

Model	Balanced-Acc
Sent-Factuality	65.6
DAE	**78.7**
DAE-Weak	70.9

Table 4: Comparison of different factuality models when trained on human annotated data and evaluated on XSUM (compare to Table 3). Fine-grained annotations provide a big boost in performance.

Model	Precision	Recall	F1
Dependency-level			
DAE	69.7	78.2	73.7
DAE-Weak	54.9	76.6	63.9
Word-level			
DAE	57.5	74.7	65.0
DAE-Weak	56.2	62.3	59.1
DAE (best-ckpt)	62.0	83.9	71.3

Table 5: Error localization comparison of the different factuality models. The DAE model achieves high recall for both word-level and dependency-level factuality.

in Figure 1).

The results are shown in Table 4. Comparing these with results from Table 3, we see that a small number of human annotated examples can outperform large auto-generated training datasets by a large margin. Notably, we see that **availability of fine-grained factuality annotations significantly boosts performance**, with models that leverage that information (DAE) significantly outperforming sentence-level models. Even in the absence of fine-grained annotations, we see that the DAE-Weak model that decomposes the error computation and explicitly tries to localize errors is better than the sentence-level model.

However, **these factuality models do not transfer to CNN/DM**: the best model achieves an accuracy of 55.9, substantially lower than 76.7% in Table 7. This demonstrates that summarization models make different types of errors on different domains, and data collection and modelling efforts for factuality should account for these differences.

6 Localization of errors

Our evaluation so far has focused on the sentence-level performance of factuality models. Next, we evaluate the models' ability to localize errors within the generated summary as well as show how such a capability can be leveraged to train less error-prone summarization models.

6.1 Localizing Factuality on XSUM

We evaluate the error localization performance of the models at two granularity levels: (1) **Dependency arc-level** and (2) **Word-level**.[6] Table 5 outlines the results of our experiments.

The DAE model outperforms the DAE-Weak model at both levels of granularity. This reiterates our earlier claim that **fine-grained annotations lead to better factuality models with more reliable localization**. However, the DAE-Weak model is able to achieve comparable recall at the dependency-level; both models are more recall-oriented, which is desirable for certain applications.

For Section 6.2, we select our DAE model's best checkpoint on the test data (**best-ckpt**), which achieves a recall of 83.9, a significant gain if we directly optimize for this metric.

6.2 Downstream Applications

Localizing errors potentially allows for post-hoc correction (Zhao et al., 2020; Cao et al., 2020); however, repairing a summary to be fully factual is a very hard problem and past work has focused on a subset of errors as a result. Instead, we show that even our imperfect error localization techniques can be used to meaningfully improve the *training* data for summarization. We use our DAE model to identify unsupported facts in the XSUM training data and ignore the corresponding tokens when training our summarization model.

Training on a subset of tokens Summarization models are trained to maximize the log likelihood of the summary given the source article: $\mathcal{L} = \sum_{i=1:|S|} \log p(S_i|D, S_{1:i-1})$. When a word in the summary is non-factual, training on it encourages the model to hallucinate new content. In our approach, we modify the training objective to only maximize the likelihood of *factual* words in the summary, factuality being determined by the DAE model from the previous sections: $\mathcal{L} = \sum_{i=1:|S|} M_i \log p(S_i|D, S_{1:i-1})$ where $M_i = 1$ if the word w_i is factual, otherwise $M_i = 0$. A similar objective has been used by prior work (Song et al., 2020b) to encourage the model to copy words present in the source.

We compare our approach with two systems: a baseline model trained without this masking and

[6] We can approximately extract word-level decision from the dependency-level predictions: if any arc containing word w is non-factual, then w is non-factual; otherwise, it is factual.

Model	Word-ER ↓	Sent-ER ↓	Human ↑
Baseline	32.2	74.0	37.4
Loss trunc	31.1	70.9	39.1
DAE-based (ours)	**23.7**	**61.4**	**46.5**

Table 6: Comparison of the different summarization models. Our proposed approach achieves significantly lower word error rates, sentence error rates and are rated higher by human annotators.

a model using the loss truncation technique well-suited for noisy datasets from Kang and Hashimoto (2020). All models are trained on 50k examples using BART summarization model initialized from the BART-XSUM-LARGE checkpoint. For all these approaches, summaries generated on the original XSUM test set (11k examples) are compared.[7]

Evaluation First, we use our trained DAE model to evaluate the performance of our summarization models. That is, we generate summaries for all examples in the test set using the three models; the DAE model is then used to compute the word error rate (fraction of words determined to be non-factual according to the DAE model) and the sentence error rate (fraction of sentences determined to be non-factual). Table 6 outlines the results, which show that our DAE-masked training leads to better factuality performance.

Next, we perform human evaluation to compare the factuality of summaries generated by the three models using Amazon Mechanical Turk. We randomly sampled 50 articles from the test set and generated summaries corresponding to the 3 models.[8]. We asked 7 human annotators to classify each (article, summary) pair as either factual (score = 1) or non-factual (score = 0). An average score is computed for each summary by aggregating the 7 annotator scores. Table 6 reports the average summary scores for the 50 (article, summary) pairs across the 3 summarization models. The results show that the proposed approach outperforms both the baseline model and the loss truncation approach. This demonstrates that **factuality models trained on a small number of annotated examples can be used to train *factual* summarization models, even when the underlying summarization dataset is noisy**.

[7]To ensure fair comparison between the different models, we removed the examples from XSUM-HUMAN used to train the factuality models from our test set.

[8]See Appendix E for more details about the task design.

7 Related Work

Earlier work on abstraction (Barzilay et al., 1999; Carenini and Cheung, 2008) and compression (Knight and Marcu, 2000; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Durrett et al., 2016) in summarization has typically focused evaluation on content selection and grammaticality, with little heed paid to factuality. Human evaluation similarly focused on content selection (Gillick and Liu, 2010). Methods such as Pyramid (Nenkova and Passonneau, 2004) that could have in principle been used to evaluate factuality were primarily used to understand content selection.

Recent work has explored different methods for enforcing factuality: modifying the model, such as encoding SRL structures in the input (Cao et al., 2018), post-hoc correction (Dong et al., 2020), or constrained decoding (Song et al., 2020a; Mao et al., 2020). However, these techniques fundamentally struggle to handle the whole range of factual errors; factuality is a fuzzy notion and cannot be easily encapsulated into a set of discrete rules.

Faithfulness and factuality have also been tackled in related tasks, including summarizing radiology reports (Zhang et al., 2020b) and data-to-text generation tasks (Tian et al., 2019). Another recent line of work has looked at fact verification (Thorne et al., 2018; Nie et al., 2019; Atanasova et al., 2020). In this literature, the claims are usually human-authored and a straightforward statement of a fact, whereas generated summaries might feature claims buried in nominal modifiers like *two-time winner*.

8 Conclusion

In this work, we showed that existing synthetic datasets are not well-suited to factuality evaluation of recent summarization models (like BART) in challenging domain (like XSUM). Models trained on human-annotated data, especially those that leverage fine-grained annotations, can enable training of more factual summarization models. We hope future work will explore better modeling and data creation to address the pressing issues in current systems.

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522, a gift from Salesforce Inc, and an equipment grant from NVIDIA. Thanks as well to Jiacheng Xu and the anonymous reviewers for their helpful comments.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA. Association for Computational Linguistics.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly Learning to Extract and Compress. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 481–490.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Giuseppe Carenini and Jackie C. K. Cheung. 2008. Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 33–41, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 280–287.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331.

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.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1998–2008.

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 (TACL)*.

Tobias Falke, Leonardo FR 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.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151, Los Angeles. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020a. Evaluating factuality in generation with dependency-level entailment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3592–3603.

Tanya Goyal and Greg Durrett. 2020b. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Kevin Knight and Daniel Marcu. 2000. Statistics-Based Summarization—Step One: Sentence Compression. In *Proceedings of the National Conference on Artificial Intelligence (AAAI) and Confer-*

ence on Innovative Applications of Artificial Intelligence (IAAI), pages 703–710.

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.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Hong Kong, China. 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*, Online. Association for Computational Linguistics.

Klaus-Michael Lux, Maya Sappelli, and Martha Larson. 2020. Truth or Error? Towards systematic analysis of factual errors in abstractive summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 1–10, Online. Association for Computational Linguistics.

Yuning Mao, Xiang Ren, Heng Ji, and Jiawei Han. 2020. Constrained Abstractive Summarization: Preserving Factual Consistency with Constrained Generation. In *arXiv preprint 2010.12723*.

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

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

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.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Kaiqiang Song, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, Xiangyang Xue, Chen Li, Dong Yu, and Fei Liu. 2020a. Joint parsing and generation for abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8894–8901.

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020b. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8902–8909.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics.

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. 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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Kristian Woodsend and Mirella Lapata. 2012. Multiple Aspect Summarization Using Integer Linear Programming. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*.

Fang-Fang Zhang, Jin-ge Yao, and Rui Yan. 2018. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790.

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

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Zheng Zhao, Shay B Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of EMNLP*.

A Manual Annotation of Errors

In Section 3, we outline the error distributions of multiple factuality datasets. The distributions were obtained by combing manual annotations from two authors of this paper. On a common set of 50 summaries annotated by both authors, we observe the following: (1) Both authors agreed on what spans/hallucinations within a summary constitute an error 74% of the times. (2) In cases when both authors marked a common span as erroneous, they agreed on the error category 84% of the time.

B Synthetic Dataset Performance on held-out samples

Section 5.1 evaluates the performance of models trained on the synthetic datasets on human annotated test sets for two summarization domains. Here, we report the model performance on held-out tests datasets that are constructed in the same way as the training datasets. Table 7 presents these results. For both domains, we see that the models report very high performance indicating that they are able to fit the distribution of the synthetic domain. However, we see in Section 5.1 that the performance is significantly lower on actual generation outputs, with close to majority label baseline performance on the more challenging XSUM domain. This means that the two datasets have inherently different error distributions.

Figure 6 shows the balanced accuracy values reported by the model at different points during its training, on both the synthetic and human-annotated test sets. The graph clearly shows that performance on the human annotated dataset (CNN/DM) has high variance, compared to the held-out dataset accuracies which has a steadily increasing performance. This behavior was observed for both ENT-C and GEN-C domains; however, ENT-C exhibited more variance. This indicates that the synthetic datasets are targeting a different error distribution, and optimizing for the synthetic distribution does not necessarily improve results on the actual generation errors.

	Ent-C	Gen-C
CNN/DM		
Sent-Factuality	96.4	91.2
DAE	95.4	97.3
DAE-Weak	94.8	97.8
XSUM		
Sent-Factuality	96.1	97.9
DAE	94.3	97.1
DAE-Weak	95.3	95.9

Table 7: Performance of factuality models when trained on synthetic factuality training datasets on their held-out test sets.

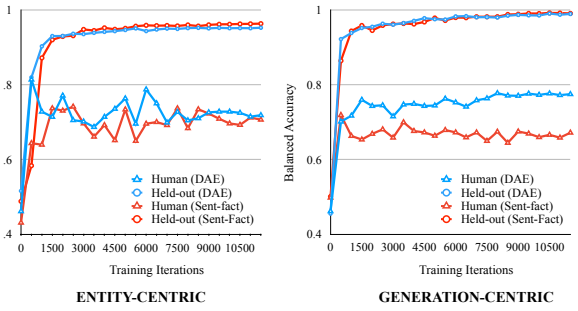

Figure 6: Performance of different train checkpoints on a held-out dataset and on the human annotated dev set for models trained on the synthetic data in the CNN/DM domain.

C Transferability of human annotations across generation models within the same domain

In Section 5.2, we demonstrate that for highly abstractive domains like XSUM, we require human annotated data to train factuality models. However, even within the same summarization domain (say XSUM), it is prohibitively expensive to collect human annotations for each summarization model that we may wish to evaluate. Here, we investigate whether the factuality annotations collected for one summarization model be used to identify factuality errors in summaries generated by other models. These experiments are done within the same domain (XSUM)

We create new training and test sets from the XSUM-HUMAN dataset. We create 2 types of train-

ing datasets for each of the 5 models annotated in that dataset: (1) **All-models** train set: This contains (A,S) pairs from all models, including the models being evaluated (2000 pairs from other models, 200 pairs from same model) and (2) **Other-models** train set: This contains (A, S) pairs from the rest of the models (2000 pairs). Evaluation is done on the remaining 300 (A,S) pairs for each summarization model. We train the best performing factuality model, i.e. the DAE model for all these settings.

	BERTS2S	PTGEN	TCONVS2S	TRANS2S
All	79.6	75.8	76.7	84.5
Others	82.3	77.0	74.1	85.3

Table 8: Performance of models trained on All-models dataset vs Other-models dataset.

Results are outlined in table 8. These show that the performance is similar for both All-models and Other-models settings for all models considered. This indicates that for the given set of summarization models considered (all trained on the same summarization training dataset), human annotations from one generation model can be used to evaluate factuality for other models.

D Implementation Details

We use the Huggingface Library (Wolf et al., 2019) for all our experiments. All our factuality models are trained by fine-tuning the pre-trained ELECTRA (electra-base-discriminator, 110M parameters) model. We perform 5 hyper parameter trials to select the best set of hyper parameters, varying the learning rate. The final hyper-parameters are:

Computing Infrastructure	32GB NVIDIA V100 GPU
Max Seq Length	512
Optimizer	Adam
Optimizer Params	$\beta = (0.9, 0.999), \epsilon = 10^{-8}$
Learning Rate Decay	Linear
Learning rate	2e-5
Weight Decay	0
Warmup Steps	0
Max Gradient Norm	1
Batch size	8
Epochs	3

Table 9: Hyperparameters used for fine-tuning the factuality models.

For models with high variance (sent-factuality model from section 5.2), we report average of 3 runs by initializing with a random seed.

The hyperparameters for training the BART summarization models are given in Table 10. Parameter settings used during decoding to generate summaries on the test set are also included

For training	
Computing Infrastructure	32GB NVIDIA V100 GPU
Max Input Seq Length	512
Max Output Seq Length	128
Optimizer	Adam
Optimizer Params	$\beta = (0.9, 0.999), \epsilon = 10^{-8}$
Learning Rate Decay	Linear
Learning rate	2e-5
Weight Decay	0
Warmup Steps	0
Max Gradient Norm	1
Batch size	8
Epochs	10
For decoding	
Num beams	6
Length Penalty	2
No repetition size	3-grams
Min-Length	10
Max Length	60

Table 10: Hyperparameters used for fine-tuning and decoding using the BART-based summarization models.

E Human Study

Figure 7 provides an screenshot of the Amazon Mechanical Turk tasks used to obtain human judgements for factuality of generated summaries, as outlined in Section 6.2. Workers were presented with a source article and 3 corresponding summaries. Each of these summaries were marked as Factual or Non-Factual. Additionally, they were asked to highlight the span within the summary that was erroneous.

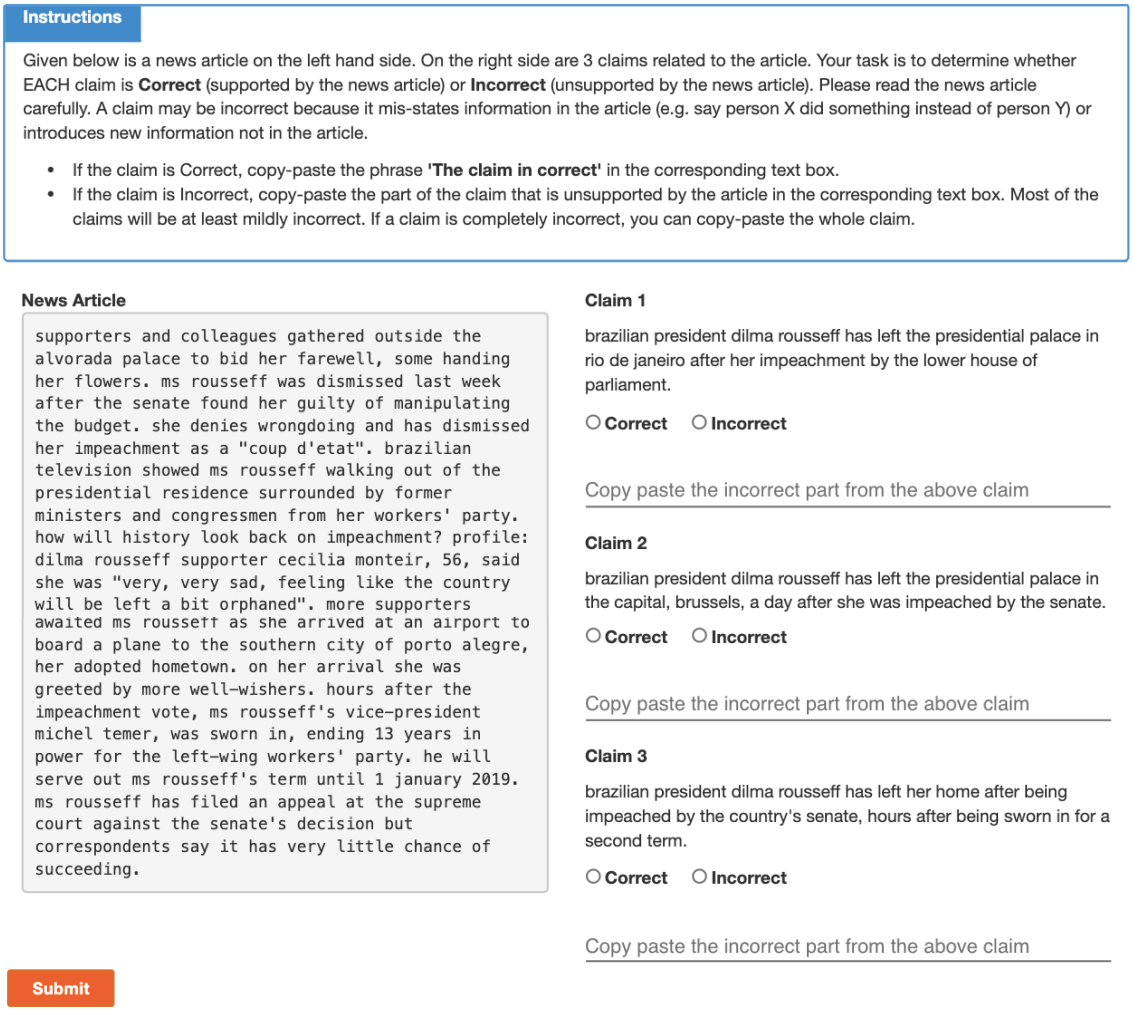

Figure 7: Screenshot of the Mechanical Turk experiment. Given an input articles, the annotators were tasked with evaluating the factuality of 3 model generated summaries on a binary scale.