Interpretable Automatic Fine-grained Inconsistency Detection in Text Summarization

Hou Pong Chan[1] **Qi Zeng**[2] **Heng Ji**[2]
[1]University of Macau [2]University of Illinois Urbana-Champaign
hpchan@um.edu.mo, {qizeng2, hengji}@illinois.edu

Abstract

Existing factual consistency evaluation approaches for text summarization provide binary predictions and limited insights into the weakness of summarization systems. Therefore, we propose the task of fine-grained inconsistency detection, the goal of which is to predict the fine-grained types of factual errors in a summary. Motivated by how humans inspect factual inconsistency in summaries, we propose an interpretable fine-grained inconsistency detection model, FINEGRAINFACT, which explicitly represents the facts in the documents and summaries with semantic frames extracted by semantic role labeling, and highlights the related semantic frames to predict inconsistency. The highlighted semantic frames help verify predicted error types and correct inconsistent summaries. Experiment results demonstrate that our model outperforms strong baselines and provides evidence to support or refute the summary.[1]

1 Introduction

Prior work (Fabbri et al., 2022b; Goyal and Durrett, 2020; Laban et al., 2022) formulates the problem of factual inconsistency detection as a binary classification task, which predicts whether a summary is consistent with the source document. However, these approaches have two drawbacks. First, they cannot predict the types of factual errors made by a summary and thus provide limited insights into the weakness of summarization systems. Although recent studies (Pagnoni et al., 2021; Tang et al., 2022; Goyal and Durrett, 2021a) have manually inspected the types of factual errors in summaries, there is *no existing work on automatic detection of fine-grained factual inconsistency.*

Second, existing models typically cannot explain which portions of the document are used to detect the inconsistency in the input summary. In order to verify and correct an inconsistent summary, humans still need to read the entire source document to find the supporting evidence. Kryscinski et al. (2020) introduce an auxiliary task to extract the supporting spans in the document for inconsistency detection, which *requires expensive ground-truth labels of supporting spans*.

To address the first limitation, we propose the **fine-grained factual inconsistency detection** task. The goal is to predict the types of factual inconsistency in a summary. We show examples of different factual error types in Table 1.

To solve the second challenge, we further introduce an **interpretable fine-grained inconsistency detection model** (FINEGRAINFACT) that does not require any label of supporting text spans, inspired by how humans verify the consistency of a summary. When humans annotate the factual error types of a summary, they first identify facts in the document that are relevant to the summary and then determine the factual error types in the summary. Following this intuition, our model first extracts facts from the document and summary using Semantic Role Labeling (SRL). We consider each extracted semantic frame as a fact since a semantic frame captures a predicate and its associated arguments to answer the question of "who did what to whom". After fact extraction, a document fact attention module enables the classifier to focus on the facts in the document that are most related to the facts in the summary. By highlighting the facts in the document with the highest attention scores, our model can explain which facts in the document are most pertinent to inconsistency detection.

Experiment results show that our model outperforms strong baselines in detecting factual error types. Moreover, the document facts highlighted by our model can provide evidence to support or refute the input summary, which can potentially help users to verify the predicted error types and correct an inconsistent summary.

[1]Code and data are available at https://github.com/kenchan0226/fineGrainedFact

Findings of the Association for Computational Linguistics: ACL 2023, pages 6433–6444
July 9-14, 2023

Source text	
Marcy Smith was woken up by her son David to find their house in Glovertown, Newfoundland and Labrador, completely engulfed in flames ... Mrs Smith said if it wasn't for her son, she and her daughter probably wouldn't have survived. **David was on FaceTime to his father at the time**, so was the only one awake and **saw the flames out of the corner of his eye** ...	
Error type	**Example summary**
Extrinsic noun phrase error: Errors that add new object(s), subject(s), or prepositional object(s) that cannot be inferred from the source article.	David was using FaceTime with *Maggie Smith* and saw the flames.
Intrinsic noun phrase error: Errors that misrepresent object(s), subject(s), or prepositional object(s) from the source article.	David was using FaceTime with *Marcy Smith* and saw the flames.
Extrinsic predicate error: Errors that add new main verb(s) or adverb(s) that cannot be inferred from the source article.	David was *eating* and saw the flames.
Intrinsic predicate error: Errors that misrepresent main verb(s) or adverb(s) from the source article.	David was *engulfed* and saw the flames.

Table 1: A text document and example summaries with different factual error types according to the typology defined by Tang et al. (2022). The errors in the sample summaries are in red color and italicized. We bold the text spans from the document that refute the sample summaries.

2 Task Definition

The goal of the fine-grained inconsistency detection task is to predict the types of factual errors in a summary. We frame it as a multi-label classification problem as follows. Given a pre-defined set of l factual error types $\{e_1, \ldots, e_l\}$, a document $\mathbf{d}$, and a summary $\mathbf{s}$, the goal is to predict a binary vector $\mathbf{y} \in \{0, 1\}^l$ where each element y_i indicates the presence of one type of factual errors.

We follow the typology of factual error types proposed by (Tang et al., 2022), which include *intrinsic noun phrase error*, *extrinsic noun phrase error*, *intrinsic predicate error*, and *extrinsic predicate error*. The definitions and examples of these error types are presented in Table 1.

3 Our FINEGRAINFACT Model

The model architecture is illustrated in Figure 1.

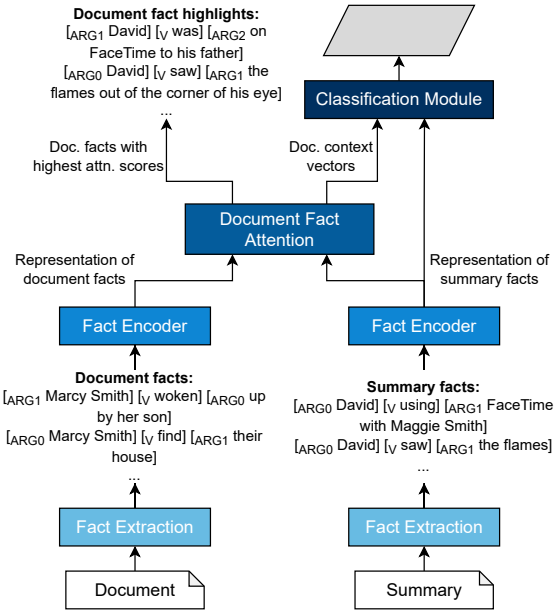

Figure 1: The architecture of FINEGRAINFACT. The fact extraction module represents facts from the input document and summary with semantic frames. The document fact attention module queries the document facts with summary facts and highlights those with the highest attention scores. Based on the retrieved highlighted document context and summary facts, the classification module predicts the factual error types.

Fact extraction. To represent facts from the input document and summary, we extract semantic frames with a BERT-based semantic role labeling (SRL) tool (Shi and Lin, 2019). A semantic frame contains a predicate and its arguments, e.g., [ARG0David][Vsaw][ARG1the flame]. We use f_i^{doc} and f_i^{sum} to denote the i-th fact in the document and summary, respectively.

Fact encoder. We first represent tokens in the concatenated sequence of the input document and summary by fusing hidden states across all layers in Adapter-BERT (Houlsby et al., 2019) with max pooling. To represent facts, we apply attentive pooling to all tokens in the semantic frame under the assumption that different tokens in a fact should contribute differently to the fact representation. Given the token representations $\mathbf{t}_j$, we calculate the attention scores $\alpha_j = \exp(\phi(\mathbf{t}_j)) / \sum_{j=1}^{m} \exp(\phi(\mathbf{t}_j))$, and represent each document or summary fact as $\mathbf{f}_i = \sum_{j=1}^{m} \alpha_j (\phi(\mathbf{t}_j))$, where m is the number of tokens in the fact and ϕ is a two-layer fully-connected network.

Document Fact Attention module. This module aims to retrieve the facts in the document that are related to the facts in the summary. We first concatenate the document fact representations into a document fact matrix $\mathbf{F}^{doc}$. We attend each summary fact f_i^{sum} to the document fact matrix to compute a **document context vector**: $\mathbf{c}_i = \text{MULTIHEADATT}(\mathbf{f}_i^{sum}, \mathbf{F}^{doc}, \mathbf{F}^{doc})$, where $\mathbf{f}_i^{sum}$ acts as the query, $\mathbf{F}^{doc}$ is used as the key and value. The document context vector $\mathbf{c}_i$ captures the information of the facts in the document that are related to the summary fact f_i^{sum}.

For each document fact, we sum up its attention scores received from all summary facts as its importance score. Concretely, we use $\alpha_{j \to i}$ to denote the sum of attention scores injected from the j-th summary fact to the i-th document fact over all attention heads. The importance score of a document fact f_i^{doc} is defined as $\sum_{j=1}^{n} \alpha_{j \to i}$, where n is the total number of facts in the summary. Then, we return the top k document facts with the highest importance scores as the **document fact highlights**, where k is a hyper-parameter.

Classification module. A linear classifier predicts the probability of each factual error type based on the concatenation of the representations of summary facts and document context vectors. Specifically, we first use mean pooling to fuse all summary fact representation vectors and all document context vectors into two fixed-size vectors: $\bar{\mathbf{f}}^{sum} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{f}_i^{sum}$, $\bar{\mathbf{c}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{c}_i$. These two vectors contain the information of all facts in the summary and the information of all document facts that are related to the summary. Next, we feed the concatenation of $\bar{\mathbf{f}}^{sum}$ and $\bar{\mathbf{c}}$ to a linear classification layer to predict the probability of each factual error type: $p(\mathbf{y}) = \sigma(\mathbf{W}[\bar{\mathbf{f}}^{sum}; \bar{\mathbf{c}}] + b)$, where $\mathbf{W} \in \mathbb{R}^{d \times l}, b \in \mathbb{R}$, d is the hidden size of Adapter-BERT, σ denotes the sigmoid function.

Training objective. We train our model with weighted binary cross-entropy (BCE) loss, The technical details are in Appendix A.

4 Experiments

4.1 Setup

Dataset. We conduct experiments on the Aggrefact-Unified dataset (Tang et al., 2022), which collects samples and unifies factual error types from four manually annotated datasets (Maynez et al., 2020; Pagnoni et al., 2021; Goyal and Durrett, 2021b; Cao and Wang, 2021). We remove the duplicated samples (i.e., duplicated document-summary pairs) in the Aggrefact-Unified dataset (Tang et al., 2022) and obtain 4,489 samples. We randomly split data samples into train/validation/test sets of size 3,689/300/500. The statistics of the error type labels are in Appendix B.1.

Evaluation metrics. We adopt the macro-averaged F1 score and balanced accuracy (**BACC**) as the evaluation metrics. BACC is an extension of accuracy for class-imbalanced datasets and is widely adopted by previous literature on inconsistency detection (Kryscinski et al., 2020; Laban et al., 2022). All experiment results are averaged across four random runs.

Baselines. We adapt the following baselines[2] for the new task. **FACTCC-MULTI**: FactCC (Kryscinski et al., 2020) is originally trained on synthetic data for binary inconsistency detection. We replace the binary classifier with a multi-label classifier and finetune the model on Aggrefact. **FACTGRAPH-MULTI**: FactGraph (Ribeiro et al., 2022) parses each sentence into an AMR graph and uses a graph neural network to encode the document. We replace the binary classifier with a multi-label classifier. We also fine-tune the **BERT** (Devlin et al., 2019) and **ADAPTERBERT** (Houlsby et al., 2019).

4.2 Performance of Error Type Detection

Following (Tang et al., 2022), we detect error types in summaries from different models: **SOTA** includes the pre-trained language models published in or after 2020. **XFORMER** contains the Transformer-based models published before 2020. **OLD** includes earlier RNN- or CNN-based models. **REF** represents reference summaries. From Table 2, we observe that: (1) *Representing facts with semantic frames improves factual error type prediction.*. We observe that in most of the cases, our model outperforms other baselines that do not use semantic frames to represent facts. (2) *The performance of our model drops after we remove the document fact attention module*. The results show that our document fact attention module *not only improves the interpretability, but also boost*

[2]We do not use QA-based metrics (Scialom et al., 2021) as our baselines. It is because both noun phrase errors and predicate errors in the summary can cause a QA model to predict incorrect answers. Hence, we cannot decide the types of factual errors based on the outputs of QA-based metrics.

Model	SOTA		XFORMER		OLD		REF		All	
	F1	BACC	F1	BACC	F1	BACC	F1	BACC	F1	BACC
BERT	32.15	62.45	45.79	59.79	47.48	65.13	41.70	57.08	45.14	63.59
ADAPTERBERT	33.87	62.95	46.01	59.21	46.87	63.72	42.42	57.57	45.06	63.05
FACTCC-MULTI	34.35	64.04	45.20	**60.28**	47.43	64.47	36.52	48.90	44.59	63.05
FACTGRAPH-MULTI	34.24	63.62	37.03	56.89	38.12	59.76	35.66	52.63	37.47	59.61
FINEGRAINFACT	**35.10**	**64.08**	**46.02**	59.42	**48.63**	**65.48**	**46.44**	**61.81**	**46.43**	**64.31**
– Doc. Fact Attention	34.77	63.12	45.61	59.36	47.43	64.63	46.35	60.67	45.96	63.99

Table 2: Performance of fine-grained consistency detection models in summaries generated by different systems (%). "– Doc. Fact Attention" indicates that we remove the document fact attention module and use mean pooling to fuse all document semantic representation vectors.

Model	R@3	R@4	R@5
BERT	36.76	46.18	53.34
ADAPTERBERT	36.34	46.14	53.80
FACTCCMULTI	41.11	50.95	58.41
FACTGRAPHMULTI	42.25	52.10	60.24
FINEGRAINFACT	**49.99**	**59.91**	**67.92**

Table 3: The recall@3,4,5 scores of document fact highlights (%).

the performance of factual error type detection. (3) *All detection models perform better in summaries generated by OLD systems*. It suggests that the factual errors made by OLD systems are relatively easier to recognize than the errors made by more advanced systems.

4.3 Evaluation of Document Fact Highlights

Since ground-truth document fact highlights are not available, we apply a fact verification dataset to evaluate whether the predicted document fact highlights provide evidence for inconsistency detection. Specifically, we adopt the FEVER 2.0 dataset (Thorne et al., 2018), which consists of claims written by humans and evidence sentences from Wikipedia that can support or refute the claims. We first extract facts from the evidence sentences via SRL and use them as the *ground-truth document fact highlights*. We then consider each claim as the input summary and the section of a Wikipedia article that contains the evidence sentences as the input document.

We devise the following method to compute document fact highlights for the baseline models. Since all baselines utilize the CLS token to predict the factual error types, we use the attention scores received from the CLS token to compute an importance score for each document fact. We then return the facts that obtain the highest importance scores as the document fact highlights for each baseline. More details are in Appendix B.2.

Table 3 presents the recall scores of document fact highlights predicted by different models. We observe that our model obtains substantially higher recall scores, which demonstrates that our model provides more evidence to support the inconsistency prediction. Thus, compared with the baselines, our model allows users to verify the predicted error types and correct inconsistent summaries.

Source text: Children in P6 and P7 will learn how to cope with change under the Healthy Me programme developed by Northern Ireland charity , Action Mental Health ... The charity is now hoping the programme will be rolled out in schools across Northern Ireland ... ...
Summary generated by an OLD model: *a school in northern ireland* has launched a programme to help children with mental health problems in northern ireland .
Ground-truth factual error type: Intrinsic Noun Phrase Error
Factual error type predicted by FINEGRAINFACT: Intrinsic Noun Phrase Error
Document fact highlight predicted by FINEGRAINFACT ($k = 1$): 1. [ARG1 the Healthy Me programme] [V developed] [ARG0 by Northern Ireland charity , Action Mental Health]

Table 4: Sample outputs of our FINEGRAINFACT model in the Aggrefact-Unified dataset. The error in the sample summary is in red color and italicized.

4.4 Case Study

Table 4 shows a sample summary generated by an OLD model with an intrinsic noun phrase error, where the "a school in northern ireland" in the summary contradicts with "Northern Ireland charity" in the document. Our model accurately predicts the error type with evidence in the form of document fact highlight, which helps users verify the error and correct the summary.

In Table 5, we present an error analysis on a sample summary generated by a SOTA model. According to the source text, the word "West" in the summary is incorrect and should be removed since the statement in the summary is made by "Sussex PPC" instead of "West Sussex PCC". In order to

Source text: The move is part of national fire service reforms unveiled by Home Secretary Theresa May last week . **Sussex PCC Katy Bourne** said emergency services would have an increased duty to collaborate under the new bill . But West Sussex County Council (WSCC) said it already had an excellent model . East Sussex ' s fire authority said it would co - operate with the PCC but it believed collaboration could be achieved without elaborate structural change . **Ms Bourne said she had written to WSCC leader Louise Goldsmith and Phil Howson , East Sussex Fire Authority chairman , to request they begin to look at the feasibility of bringing both fire services under her authority .** ...
Summary generated by a SOTA model: *West* Sussex 's police and crime commissioner (PCC) has said she wants to look at the feasibility of bringing East Sussex 's fire service under her authority .
Ground-truth factual error type: Intrinsic Noun Phrase Error
Factual error type predicted by FINEGRAINFACT: No Error
Document fact highlights predicted by FINEGRAINFACT ($k = 5$): 1. [ARG1 collaboration] [ARGM-MOD could] [V achieved] [ARGM-MNR without elaborate structural change] 2. [V bringing] [ARG1 both fire services] [ARG3 under her authority] 3. [ARG0 they] [V begin] [ARG1 to look at the feasibility of bringing both fire services under her authority] 4. [ARG0 they] [V look] [ARG1 at the feasibility of bringing both fire services under her authority] 5. [ARG0 she] [V request] [ARG1 they begin to look at the feasibility of bringing both fire services under her authority]

Table 5: Incorrect output sample of our FINEGRAINFACT model in the Aggrefact-Unified dataset (Tang et al., 2022). The error in the sample summary is in red color and italicized. We bold the text spans from the document that refute the sample summary.

detect this error, a model needs to understand that the expressions "Sussex PCC Katy Bourne", "Ms Borune", and "she" in the document refer to the same entity. This sample illustrates that the errors generated by a SOTA model are more subtle and more difficult to be detected. Our model fails to predict the correct error type for this sample. Since the top five document fact highlights returned by our model do not contain the entity "Sussex PCC Katy Bourne", we suspect that our model fails to recognize the co-referential relations among "Sussex PCC Katy Bourne", "Ms Borune", and "she" for this sample. Thus, improving the co-reference resolution ability of fine-grained inconsistency detection models is a potential future direction.

5 Related Work

Factual consistency metrics. QA-based consistency metrics (Durmus et al., 2020; Scialom et al., 2021; Fabbri et al., 2022b) involve generating questions from the given document and its summary, and then comparing the corresponding answers to compute a factual consistency score. Entailment-based consistency metrics (Laban et al., 2022; Kryscinski et al., 2020; Ribeiro et al., 2022; Goyal and Durrett, 2020) utilize a binary classifier to determine whether the contents in a system summary are entailed by the source article. In contrast, our model is a multi-label classifier that detects the types of factual errors in a summary. Moreover, our model leverages SRL to encode the facts in the input document and summary, enabling users to interpret which facts in the document are most relevant to the inconsistency detection.

Fact-based evaluation methods. To evaluate the informativeness of a summary, the Pyramid human evaluation protocol (Nenkova and Passonneau, 2004) asks annotators to extract semantic content units (SCUs) from the system summary and reference summary, respectively, and then compute their overlap. Each SCU contains a single fact. Xu et al. (2020) approximate the Pyramid method by using SRL to extract facts. They then compute the embedding similarity between the facts extracted from the system summary and those from the reference summary. Fischer et al. (2022) also use SRL to extract facts, but they measure the similarity between the facts extracted from the system summary and those from the source document to compute a faithfulness score. On the other hand, our model integrates SRL with a multi-label classifier to predict the factual error types of a summary.

6 Conclusion

In this paper, we present a new task of fine-grained inconsistency detection, which aims to predict the types of factual inconsistencies in a summary. Compared to the previous binary inconsistency detection task, our new task can provide more insights into the weakness of summarization systems. Moreover, we propose an interpretable fine-grained inconsistency detection model, which represents facts from documents and summaries with semantic frames and highlights highly relevant document facts. Experiments on the Aggrefact-Unified dataset show that our model can better identify factual error types than strong baselines. Furthermore, results on the FEVER 2.0 dataset validate that the highlighted document facts provide evidence to support the inconsistency prediction.

7 Limitations

Although our model allows users to interpret which parts of the input document are most relevant to the model's prediction, our model does not allow users to interpret which text spans of the input summary contain errors. We use the summary in Table 4 as an example. If the model can indicate the text span "a school in northern ireland" contains errors, it will be easier for users to correct the summary, potentially benefiting factual error correction systems (Fabbri et al., 2022a; Huang et al., 2023). Kryscinski et al. (2020) introduced an auxiliary task to extract erroneous text spans in summaries, but their method requires expensive text span ground-truth labels. Locating incorrect text spans in the summaries without requiring span-level training labels remains unexplored. Another limitation of our model is that it does not allow users to interpret the uncertainty of the prediction results (Deutsch et al., 2021).

8 Ethical Considerations

The factual error types and document fact highlights predicted by our model can help users correct factually inconsistent summaries. Since factually inconsistent summaries often convey misinformation, our model can potentially help users combat misinformation. However, the factual error types predicted by our model may be incorrect. For example, it is possible that an input summary contains extrinsic noun phrase errors, but our model predicts the error type of intrinsic predicate error. Hence, users still need to be cautious when using our model to detect and correct inconsistent summaries. The Aggrefact-Unified dataset contains public news articles from CNN, DailyMail, and BBC. Hence, the data that we used does not have privacy issues.

Acknowledgement

We thank the anonymous reviewers for their insightful comments on our work. This research is based upon work supported by U.S. DARPA AIDA Program No. FA8750-18-2-0014, DARPA INCAS Program No. HR001121C0165, NSF under award No. 2034562, the Molecule Maker Lab Institute: an AI research institute program supported by NSF under award No. 2019897 and No. 2034562, and the AI Research Institutes program by National Science Foundation and the Institute of Education Sciences, U.S. Department of Education through Award # 2229873 - AI Institute for Transforming Education for Children with Speech and Language Processing Challenges. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government, the National Science Foundation, the Institute of Education Sciences, or the U.S. Department of Education. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. Hou Pong Chan was supported in part by the Science and Technology Development Fund, Macau SAR (Grant Nos. FDCT/060/2022/AFJ, FDCT/0070/2022/AMJ) and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST).

References

Shuyang Cao and Lu Wang. 2021. CLIFF: contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6633–6649. Association for Computational Linguistics.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Trans. Assoc. Comput. Linguistics*, 9:1132–1146.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Esin Durmus, He He, and Mona T. 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, ACL 2020, Online, July 5-10, 2020*, pages 5055–5070. Association for Computational Linguistics.

Alexander R. Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022a. Improving factual consistency in summarization with compression-based post-editing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages

9149–9156. Association for Computational Linguistics.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022b. Qafacteval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2587–2601. Association for Computational Linguistics.

Tim Fischer, Steffen Remus, and Chris Biemann. 2022. Measuring faithfulness of abstractive summaries. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 63–73.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3592–3603. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021a. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1449–1462. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021b. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1449–1462. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Kung-Hsiang Huang, Hou Pong Chan, and Heng Ji. 2023. Zero-shot faithful factual error correction. *CoRR*, abs/2305.07982.

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 2020, Online, November 16-20, 2020*, pages 9332–9346. 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. *Trans. Assoc. Comput. Linguistics*, 10:163–177.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, 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, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 145–152. The 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, NAACL-HLT 2021, Online, June*

6-11, 2021, pages 4812–4829. Association for Computational Linguistics.

Martha Palmer, Paul R. Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguistics*, 31(1):71–106.

Revanth Gangi Reddy, Heba Elfardy, Hou Pong Chan, Kevin Small, and Heng Ji. 2022. Sumren: Summarizing reported speech about events in news. *CoRR*, abs/2212.01146.

Leonardo 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3238–3253. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6594–6604. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yahvuz, Wojciech Kryscinski, Justin F. Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *CoRR*, abs/2205.12854.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*.

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

Xinnuo Xu, Ondrej Dusek, Jingyi Li, Verena Rieser, and Ioannis Konstas. 2020. Fact-based content weighting for evaluating abstractive summarisation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5071–5081. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11765–11773. AAAI Press.

A Details of Training Objective

Since some error types may have an imbalanced distribution of positive and negative samples, we apply sampling weighting to the training objective. We first weigh the loss for the positive samples according to their proportion in the training set. Then we sum up the binary cross-entropy loss of each error type as the training objective. The weighted binary cross-entropy (BCE) loss of our model is formally defined as follows:

$$L_i = \beta_i y_i^* \log p(y_i) + (1 - y_i^*) \log(1 - p(y_i)), \tag{1}$$

$$L = \sum_{i=1}^{K} L_i, \tag{2}$$

where β_i is the weight for positive samples of the i-th error type. We set β_i to be the ratio of the number of positive samples to the number of negative samples of the i-th error type in the training data.

B Experiment Details

B.1 Aggrefact-Unified Dataset

This dataset contains news documents from CNN/DM (Nallapati et al., 2016) and XSum (Narayan et al., 2018). In addition to the four factual error types presented in Table 1, the Aggrefact-Unified dataset also provides the labels of *intrinsic entire-sentence error*, *extrinsic entire-sentence error*, and *entire-sentence error*. We map intrinsic (extrinsic) entire-sentence errors to intrinsic (extrinsic) noun phrases and intrinsic (extrinsic) predicate errors. We also map the entire-sentence error to all four types of factual errors. Statistics of the factual error type labels are shown in Table 6. Table 7 presents the statistics of summaries generated by different systems.

B.2 Extraction of Document Fact Highlights for Baseline Models

Given a baseline model and a sample output from the baseline model, we first extract all the facts from the input document by SRL. Then for each extracted document fact, we compute the average attention score injected from the CLS token to the tokens in the semantic frame in the last layer of the baseline model. This average attention score is treated as the importance score of the document fact. Concretely, we use $\alpha'_{\text{CLS}\to i}$ to denote the total attention score injected from the CLS token to the i-th token of the semantic frame in the last layer of the baseline model over all attention heads. Then we compute the importance score as follows: $\sum_{i=1}^{m} \alpha'_{\text{CLS}\to i}$, where m is the number of words in the fact. Finally, we return the document facts with the highest importance scores as the document fact highlights.

Source	Ex. NP	In. NP	Ex. Pred.	In. Pred.
CNNDM	348	200	280	111
XSum	1,812	1,114	540	327

Table 6: Statistics of fine-grained error types in the AggreFact-Unified dataset.

Source	SOTA	XFORMER	OLD	REF
CNNDM	550	249	800	0
XSum	400	994	997	499

Table 7: Statistics of summaries generated by different systems in the AggreFact-Unified dataset.

B.3 Hyper-parameter Settings

To compute F1 and BACC scores, we set the classification threshold to be 0.5. The dimension of the adapter in the Adapter-BERT model is set to 32. The number of attention heads in our document fact attention module is set to 16. We search the optimal number of attention heads from $\{1, 4, 8, 16\}$ that obtains the highest BACC score in the validation set. We train our models for 40 epochs and select the checkpoint that obtains the highest BACC score in the validation set. We set the learning rate to be 1e-5. The training batch size is 12 with a gradient accumulation steps of 2. The AdapterBERT, BERT, and FineGrainFact models receive the same amount of hyperparameter tuning.

B.4 Hardware and Software Configurations

We run all the experiments using a single NVIDIA V100 GPU. It takes around 1 hour and 50 minutes to train our model for 40 epochs. Our model contains 113.1M of parameters in total. We only need to train 3.6M of the model parameters since most of the parameters are frozen by the Adapter-BERT model. We obtain the BERT-base-uncased checkpoint from Huggingface (Wolf et al., 2019). We adopt the implementation of the BERT-based SRL model (Shi and Lin, 2019) provided by AllenNLP (Gardner et al., 2018) to conduct semantic role labeling (Palmer et al., 2005).

Error Type	XSum	CNN/DM
Extrinsic NP	64.58	52.39
Extrinsic Pred.	64.26	52.15
Intrinsic NP	46.48	63.01
Intrinsic Pred.	42.61	51.53

Table 8: The F1 score results of the FINEGRAINFACT model in each summarization dataset and factual error type (%).

C Results on Different Summarization Datasets and Error Types

In Table 8, we separate the F1 scores obtained by our FINEGRAINFACT model according to the summarization dataset and the type of factual errors. It is observed that our model has relatively low performance ($< 50\%$) on detecting intrinsic errors (intrinsic noun phrase and intrinsic predicate errors) in the XSum dataset. We analyze the reason as follows. According to previous studies (Durmus et al., 2020), system summaries generated in the XSum dataset tend to have a high abstractiveness (low textual overlapping with the source document). We suspect that our FINEGRAINFACT model learns a spurious correlation that suggests an inconsistent summary with high abstractiveness contains extrinsic errors rather than intrinsic errors. A critical future direction is to address this spurious correlation of our model.

D Generalization Ability Analysis

To more robustly evaluate the generalization ability of inconsistency detection models, we further construct a challenging data split in which there are no overlapped systems and documents between the test set and the training set. We first gather all the samples that contain a summary generated by the BART model (Lewis et al., 2020) to construct the test set. We choose BART since it is a common baseline in recent summarization literature (Reddy et al., 2022; Zhong et al., 2022). After that, we randomly split the remaining data samples into training and validation sets. Finally, we remove the duplicated documents between the training set and the test set. This data split contains 3,839/550/100 samples for train/validation/test sets. The results of different inconsistency detection models are shown in Table 9. We observe that our FINEGRAINFACT model outperforms all the baselines, which demonstrates the strong generalization ability of our model.

Model	F1	BACC
BERT	38.83	59.27
ADAPTERBERT	39.88	61.20
FACTCCMULTI	32.53	58.24
FACTGRAPHMULTI	25.83	57.55
FINEGRAINFACT	**40.71**	**62.19**

Table 9: Performance of fine-grained inconsistency detection models in the challenging data split (%).

E Scientific Artifacts

We list the licenses of the scientific artifacts used in this paper: AllenNLP (Apache License 2.0), Huggingface Transformers (Apache License 2.0), and FACTCC (BSD-3-Clause License). We apply the above artifacts according to their official documentation. We will release an API of our model for research purposes. Our API can be applied to detect the fine-grained factual error types in summaries written in the English language.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
8

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

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

B ☑ Did you use or create scientific artifacts?

3, 4

☑ B1. Did you cite the creators of artifacts you used?
4.1, B.4, E

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

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

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
8

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

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4.1, B.1

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
B.4

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
B.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
B.4

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.