Annotating and Detecting Fine-grained Factual Errors for Dialogue Summarization

Rongxin Zhu Jianzhong Qi Jey Han Lau
School of Computing and Information Systems
The University of Melbourne
rongxinz1@student.unimelb.edu.au, {jianzhong.qi,laujh}@unimelb.edu.au

Abstract

A series of datasets and models have been proposed for summaries generated for well-formatted documents such as news articles. Dialogue summaries, however, have been under explored. In this paper, we present the first dataset with fine-grained factual error annotations named DIASUMFACT. We define fine-grained factual error detection as a sentence-level multi-label classification problem, and we evaluate two state-of-the-art (SOTA) models on our dataset. Both models yield sub-optimal results, with a macro-averaged F1 score of around 0.25 over 6 error classes. We further propose an unsupervised model ENDERANKER via candidate ranking using pretrained encoder-decoder models. Our model performs on par with the SOTA models while requiring fewer resources. These observations confirm the challenges in detecting factual errors from dialogue summaries, which call for further studies, for which our dataset and results offer a solid foundation.[1]

1 Introduction

Factual inconsistency in abstractive summarization — a phenomenon where model-generated summaries contain facts that are inconsistent with the source document — is a widely known problem and has been studied extensively in the document summarization community. An example is shown in Figure 1, where the source document is a dialogue — the type of documents that this paper focuses on.

Existing work covers topics on factual inconsistency including error typology and factuality annotations of state-of-the-art neural summarization models (Maynez et al., 2020; Huang et al., 2020; Pagnoni et al., 2021; Goyal and Durrett, 2021; Fabbri et al., 2021; Gao and Wan, 2022; Tang et al., 2022a), automatic factual error detectors (Wang et al., 2020; Goyal and Durrett, 2020; Kryscinski et al., 2020; Durmus et al., 2020; Zeng et al., 2021; Scialom et al., 2021), methods to correct factual errors in summaries (Cao et al., 2020; Dong et al., 2020; Chen et al., 2021a) and methods to produce factually more consistent summaries (Zhao et al., 2020; Cao and Wang, 2021; Tang et al., 2022b; Zhu et al., 2021; Aralikatte et al., 2021; Chen et al., 2021b; Balachandran et al., 2022). Almost all of these works focus on news summarization based on two datasets: CNN/DAILYMAIL (Hermann et al., 2015; Nallapati et al., 2016) and XSUM (Narayan et al., 2018).

Source Dialogue
Lilly: Wanna go out tonight?
Marshall: can't :(money's low
Lilly: my treat :)
Marshall: I wouldn't let a woman pay for me.

Factually Consistent Summary
Lilly offered to treat Marshall and he rejected.

Factually Inconsistent Summary
Lilly offered to treat Marshall and he accepted.

Figure 1: Example summaries that are factually consistent and inconsistent with a source dialogue.

Dialogue summarization (cf Figure 1), which aims to produce a condensed version of a dialogue while maintaining its salient information, is equally important due to its application to summarizing meeting transcripts (Li et al., 2019; Zhu et al., 2020; Zhong et al., 2022), daily conversations (Chen and Yang, 2020; Liu and Chen, 2021; Feng et al., 2021), customer service dialogues (Liu et al., 2019; Zou et al., 2021) and medical dialogues (Joshi et al., 2020; Krishna et al., 2021). However, factual consistency in dialogue summarization is under explored as there are currently no benchmark datasets that contain fine-grained error categories. This paper aims to fill in this gap.

To investigate factual consistency in dialogue

[1]The dataset and code are available at https://github.com/731935354/Dia-Sum-Fact

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6825–6845
July 9-14, 2023

summarization, we release DIASUMFACT with fine-grained sentence-level annotations regarding factual consistency for 475 model summaries (1,340 sentences) from six neural dialogue summarization models on two popular datasets: SAMSUM (Gliwa et al., 2019) and QMSUM (Zhong et al., 2021). We adopt a two-dimensional typology that considers the semantic roles and verifiability of error spans separately.

We formulate factual error detection as a sentence-level multi-label classification task and use DIASUMFACT to evaluate two state-of-the-art factual error detection models designed for document summarization. As there are no existing error detection model for fine-grained error categories, we adapt the two binary classification models to fit to our task. Empirical results show that they don't work well on the task, indicating its difficulty and the domain gap between document summarization and dialogue summarization.

We then propose two models: BERTMULTI and ENDERANKER. BERTMULTI is a multi-class classification model trained on synthetic data, which is created by corrupting sentences from reference summaries (Kryscinski et al., 2020). ENDERANKER is a simple unsupervised model that can leverage any pretrained encoder-decoder model to detect factual errors. Given a model-generated summary sentence containing a span of interest for error detection, ENDERANKER computes log likelihood scores for the sentence and its variants containing replacement spans fetched from the source dialogue. The scores are computed as BARTSCORE (Yuan et al., 2021), which will be explained in 4.2. We compare the scores of the sentences to determine if the span of interest and hence the summary sentence contains a factual error. We run experiments with T5 (Raffel et al., 2020), BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020), fine-tuned either on news summarization or dialogue summarization, as the encoder-decoder for ENDERANKER. The results show that BERTMULTI and ENDERANKER performs on par with the adapted state-of-the-art models in terms of macro-averaged F1.

Motivated by the strong complementarity between models, we further present two ensemble models combining the four models above. The results, while exceeding those of the individual models, are still far from indicating a practical model for factual error detection over dialogue summaries. This calls for further studies, for which our dataset and results form a solid foundation.

To summarise, this paper makes the following contributions:

- We annotate and present DIASUMFACT, the first dataset with fine-grained sentence-level factual errors for dialogue summarization, providing rich annotation including error classes, erroneous spans and explanation.
- We investigate the effectiveness of adapting state-of-the-art factual error detection models for document summarization on model-generated dialogue summaries, demonstrating the difficulty of the task.
- We propose BERTMULTI, a weakly-supervised multi-class classifier and ENDERANKER, an unsupervised factual error detector that requires no human labeled data for training and can leverage existing pre-trained encoder-decoder models. Both models perform on par with adapted SOTA factual error detection models for document summarization.
- Our experiments and analyses reveal the strengths and weaknesses of different factual error detection models, and point out future directions to improve them.

2 Related Work

Error typology and datasets. There are a few existing datasets on factual errors. Some of them use binary (factually consistent or inconsistent) labels (Kryscinski et al., 2020; Wang et al., 2020) and 5-point Likert Scale labels (Fabbri et al., 2021; Gao and Wan, 2022), which require lower efforts to annotate, but they do not provide information on how and where factual errors were made. To support fine-grained analysis, multi-class and multi-dimensional typologies are designed. Pagnoni et al. (2021) propose a linguistically motivated annotation framework that covers semantic frame errors, discourse errors and content verifiability errors. Goyal and Durrett (2021) use a 2-dimensional typology, where content verifiability and semantic error types are considered separately. Cao et al. (2022) focus on hallucinations and consider both factual and non-factual hallucination. Tang et al. (2022a) unify different error types from previous

works into a hierarchical taxonomy. These datasets mostly focus on news summaries.

DialSummEval (Gao and Wan, 2022) is another popular dataset that contains annotation on factual consistency of model-generated dialogue summaries. The core difference of our work is that we consider fine-grained error categories and the text span (i.e., starting and ending position) of an error. Thus it provides a more elaborate, diagnostic assessment as to what and where goes wrong when a summary is not factually consistent. In comparison, DialSummEval only considers coarse-grained assessment of factuality using 5-point Likert Scale (Joshi et al., 2015), without specifying the actual error type (e.g., entity error).

Factual error detection models. Most popular factual error detectors are based on either textual-entailment or question-answering (QA).

Textual-entailment-based models are generally binary classifiers that take as input the source document and a model-generated summary. For example, Kryscinski et al. (2020) train binary factual error classifiers using synthetic data. Zeng et al. (2021) use a gradient-based adversarial method to improve model accuracy. Goyal and Durrett (2020) leverage dependency-level entailment achieving better performance and interpretability.

QA-based models first generate questions from a model-generated summary (or source dialogue), and then answer those questions based on its source dialogue (or a model-generated summary). The factual consistency is decided by the similarity between the ground truth answer and the predicted answer. For example, Wang et al. (2020); Durmus et al. (2020) use a precision-oriented method that generates questions from model-generated summaries and answer them using the source document. Scialom et al. (2019) instead generate questions from a source document and answer them using the summary, making it a recall-oriented method. Scialom et al. (2021) combine recall and precision-oriented techniques into a single framework. Fabbri et al. (2022) refine the model component design and obtain a QA-based method that outperforms textual-entailment-based methods.

Our unsupervised method ENDERANKER compares a span (e.g., a person name) in a model-generated sentence with candidates (e.g., other people's names in the dialogue) and decide the factual consistency of the span based on its rank among candidates. It achieves comparable macro F1 with adapted SOTA factual error detectors for document summarization but requires no labelled resources.

3 The DIASUMFACT Dataset

This section presents our DIASUMFACT dataset and procedures to construct the dataset.

3.1 Data Source

To cover dialogues from different domains, we selected two popular datasets SAMSUM (Gliwa et al., 2019) and QMSUM (Zhong et al., 2021). SAMSUM contains daily conversations and gold summaries. QMSUM comes with queries and answers based on meeting transcripts. The answers to each query can be seen as a summary to an aspect of the meeting transcript.

For both SAMSUM and QMSUM, we randomly sampled 60 dialogues and their summaries in its test split.[2] For QMSUM, we only chose queries whose gold utterances contain no more than 700 tokens according to Bert tokenizer. [3] We manually filtered out dialogues with sensitive contents (e.g., dirty words and potential bias on gender or race). More statistics on the dataset can be found in Appendix Table 5 and Table 6.

3.2 Summary Generation Models

We generally choose models with publicly accessible pretrained model checkpoints or generated outputs instead of training models ourselves.

On SAMSUM, we use five models: **BART** (Lewis et al., 2020), **PEGASUS** (Zhang et al., 2020), **S-BART** (Chen and Yang, 2021), **CONDIGSUM** (Liu et al., 2021) and **GPT-3** (Brown et al., 2020). For **S-BART** and **CONDIGSUM**, we obtain model outputs from the original papers. For **BART** and **PEGASUS**, we generate output by running their pre-trained models.[4] For **GPT-3**, we fine-tune *curie* over SAMSum dataset and generate summaries using the official API.[5]

On QMSUM, we use three models: **PEGASUS**, **BART** and **DialogLM** (Zhong et al., 2022). Since we only focus on specific queries (i.e., queries that

[2]For QMSUM we also have the queries, in addition to the dialogues and summaries.

[3]50% of the queries on aspects of meeting transcripts satisfy this constraint.

[4]We use *linydub/bart-large-samsum* for BART and *transformersbook/pegasus-samsum* for PEGASUS. Both are from https://huggingface.co/models.

[5]We fine-tuned it on May 27th, 2022 following https://beta.openai.com/docs/guides/fine-tuning.

Dialogue	Lucas: Where r u? I'm waiting at the airport. Vanessa: There was a foul-up with the flight. I'm trying to get another ticket. Lucas: OMG. How come? Vanessa: No bloody idea. All of the flights are booked cos students are returning from holidays. Lucas: I've called the airport and they said there's a flight to New York at 9:45 p. m. Vanessa: Great, I'll book it now.

Error	**Description**	**Example Summary**	**In/Ex**
EntE	The core arguments or their attributes in a semantic frame are wrong, such as the subjects and objects.	*Vanessa is waiting at the airport.*	In
PredE	The predicate, which is usually a verb, of a semantic frame is wrong.	*Lucas has emailed the airport and got some information about the flight to New York.*	Ex
CirE	The non-core arguments, such as location modifiers, temporal modifiers are wrong.	*Lucas is waiting at the train station.*	Ex
CorefE	A pronoun or a reference (e.g., `this picture`) has a wrong antecedent or has no antecedents.	*Vanessa is trying to get another ticket for themselves.*	N/A
LinkE	The relationship, e.g., a causal relationship, between statements is wrong.	*Vanessa will book the flight to New York at 9:45 pm because students are returning from holidays.*	N/A
Others	This class covers the errors that do not fall into the above classes.	/	N/A

Table 1: Factual error type descriptions and examples. **In/Ex** refers to Intrinsic Error (In) and Extrinsic Error (Ex).

only ask about an aspect of a meeting, instead of summarizing the whole meeting), which is a subset of the original dataset, we fine-tuned them using specific queries only. The fine-tuned models achieve ROUGE scores that are better or comparable to state-of-the-art models on the complete dataset.[6]

3.3 Typology of Factual Errors

Motivated by Goyal and Durrett (2021); Pagnoni et al. (2021), we adopt a 2-dimensional typology that treats semantic role and content verifiability of error spans separately.

On the semantic role dimension, we consider six error classes **Entity Error (EntE)**, **Predicate Error (PredE)**, **Circumstance Error (CirE)**, **Coreference Error (CorefE)**, **Link Error (LinkE)** and **Others**, with definitions and examples shown in Table 1. EntE, PredE, CirE are semantic frame errors, and CorefE, LinkE are discourse errors. When a sentence in the summary does not contain any factual error, we label it as **No Error**.

For content verifiability, we consider **Intrinsic Error** (i.e., the error span consists of tokens from the source dialogue) and **Extrinsic Error** (i.e., the error span consists of tokens not mentioned in the source dialogue), a.k.a. hallucinations. This dimension is only defined for EntE, PredE and CirE.

3.4 Annotation Procedure

We recruited 12 workers for the annotation task, including nine PhD students majored in natural language processing and three Master's students majoring in linguistics and information technology. All annotators are fluent English speakers. We take an in-house annotation approach because a trial on Amazon Mechanical Turk did not yield meaningful results, even though high-quality crowd-sourced workers were sourced through strict constraints. The 12 annotators form six pairs randomly where each pair annotates 10 dialogues from each dataset.

The annotation is done in three stages: pilot study, full annotation and annotation adjudication.

An annotation task involves analysing a dialogue and the summaries generated by all corresponding models. During the pilot study, annotators are required to go through the definition and examples for each error class to learn the labelling typology. Then, they will work on two pilot tasks, which are the same for all workers. For each task, a source di-

[6]The ROUGE scores of the fine-tuned models are shown in Appendix A.2. We also tried 2-shot GPT-3 but found that it didn't work well in preliminary experiments and for that reason didn't include GPT-3.

alogue and a model-generated summary are shown at the same time, and the annotator needs to label any factual errors in each individual sentence in the summary. When all sentences in the summary are done, another summary generated by a different model will be shown. Models are anonymized and their generations are shown in random order.

During the full annotation stage, we assign each annotator 10 tasks from each dataset, which are different from the tasks in pilot study. The annotations are only done for the semantic role dimension.

In the adjudication stage, the two annotators of a pair along with an annotation quality controller (one of the authors of this paper) go through the annotations to resolve any disagreements, and detailed notes were taken for reaching the final decisions (which is released as part of the dataset as it can be useful for future analysis). Annotation mistakes are also corrected in this process. In the end, a total of 1340 sentences (99.7%) with agreed annotations were obtained, while the rest of the sentences were discarded because no agreement can be made.

Note that the annotations on the content verifiability dimension are manually created by the annotation quality controller based on the detailed meeting notes of the last stage. It is a product of a post-annotation process because the original annotators did not explicitly label the error type as extrinsic or intrinsic. Instead, the annotators mark an **Extrinsic Error** for all error spans that are not mentioned in the source dialogue. The annotation quality controller takes this information and further split them into EntE, PredE and CirE based on the semantic role of an error span, and assign **Intrinsic Error** to all original EntE, PredE and CirE, thus obtaining a 2-dimensional annotation.

3.5 Inter-annotator Agreement

We use Cohen's Kappa (McHugh, 2012) to evaluate the inter-annotator agreement. The scores in each group before adjudication are as follows. We first evaluate the agreement for binary label by merging all error types into a single negative class. The scores are 0.39, 0.44, 0.57, 0.59, 0.43, 0.51. For multi-class label, the scores are 0.34, 0.33, 0.44, 0.31, 0.31, 0.25. After adjudication we have full agreement for all instances (as explained in Section 3.4).

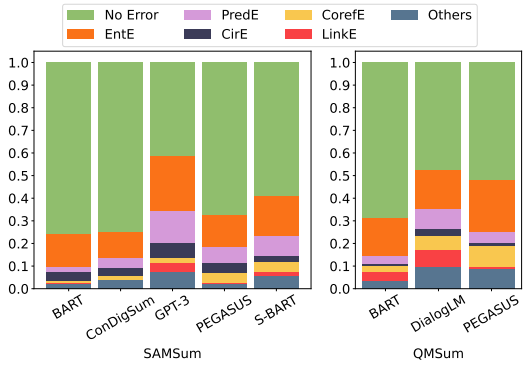

Figure 2: Semantic factual error distribution of different summarization models on SAMSUM and QMSUM.

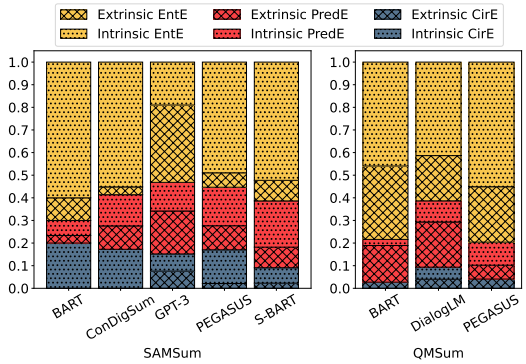

Figure 3: Intrinsic and Extrinsic error distribution for EntE, PredE and CirE of different summarization models on SAMSUM and QMSUM.

3.6 Results on the Summarization Models

We summarize the performance results of the summarization models as derived from the annotations in this subsection. Figure 2 and Figure 3 show the factual error class distribution of the summarization models evaluated on SAMSUM and QMSUM.

Overall, 33.3% and 41.9% sentences in model-generated summaries contain one or more factual errors in SAMSUM and QMSUM, respectively. The average number of errors for a factually inconsistent sentence is 1.14. This indicates a broad existence of factual errors in the model-generated summaries, thus emphasizing the importance to resolve factual errors in dialogue summarization.

Semantic frame errors (i.e., EntE, PredE and CirE) are more frequent than discourse errors (i.e., CorefE and LinkE) overall, while their distributions are not the same on both datasets. SAMSAM has a higher portion of factually inconsistent sentences caused by semantic frame errors (76.9%) than QMSUM has (58.9%), while QMSUM has a higher portion of discourse errors (24.0%) than SAMSAM (11.3%). We observe two main reasons for this

discrepancy. First, the sentences in QMSUM are longer and exhibit more complex discourse structures, especially causal relations, which can be challenging for models to summarize. Second, models fine-tuned on QMSUM tend to copy large chunks of the input dialogue. Many pronouns are directly copied from the source dialogue without proper context, causing Coreference Errors (CorefE).

Among the different summarization models, BART and PEGASUS have been evaluated on both datasets where BART generates summaries with fewer factual errors consistently. On SAMSUM, 24.0% of the sentences generated by BART contain factual errors, which is the fewest, while the highest portion is reported by GPT-3, i.e., 58.7%. CONDIGSUM and S-BART are variants of BART that achieve better ROUGE scores than BART using contrastive learning and dialogue structure information, respectively. Our results reveal that both models produced more sentences with factual errors than BART did, indicating that improvement in ROUGE may not help with the factual consistency of summaries. This result emphasizes the importance of more benchmark datasets for dialogue summarization model evaluation. On QMSUM, BART is still the best, while DIALOGLM produced the highest proportion of sentences with factual errors.

On the content verifiability dimension, models on QMSUM produce more extrinsic errors than on SAMSum. A potential reason is that reference summaries in QMSUM contain more tokens outside the source dialogue. For SAMSUM, all models are mainly dominated by intrinsic errors, while GPT-3 produces more extrinsic errors than intrinsic ones.

4 Detecting Factual Errors

In this section, we automate factual error detection in model-generated summaries. We first adapt two state-of-the-art factual error detection models from document summarization. We then propose a weakly supervised multi-class classifier and a simple yet effective unsupervised model that can utilize any pretrained encoder-decoder model to identify factual errors. Finally, we present ensemble-based models combining all techniques above.

Problem statement. We formulate factual error detection as a sentence-level multi-label classification task, i.e., given an input dialogue and a sentence from a model-generated summary, we classify whether the sentence contains any (semantic role) factual errors as outlined in Section 3.3.

Dependency Arc Types	Error Class
nsubj, obj, obl:agent, iobj, dobj, nmod, vocative, appos, nummod, compound, amod, det, clf, flat	EntE
obl:tmod, advmod	CirE
aux	PredE
other arc types	Others

Table 2: Rules to map from dependency arc types to our factual error classes.

4.1 Adapted State-of-the-Art Models

DAE (Goyal and Durrett, 2020) is based on dependency-level entailment, which predicts whether a dependency arc in a model-generated sentence is entailed by the input document (e.g., a dialogue in our problem). To adapt it to our problem, we design rules to map from dependency arc types to our factual error classes, as shown in Table 2. Given a summary sentence, we use the trained DAE provided by the authors to predict dependency arcs in the sentence. The union of all factual error classes corresponding to the types of the predicted erroneous dependency arcs will be used as our factual error predictions. Note that not all factual error classes have corresponding dependency arc types and hence not all error classes can be detected by this model.

QAFactEval (Fabbri et al., 2022) is a QA-based factual error detector. Given a question generation model (QG) and a question answering model (QA), which are trained on existing datasets for the question answering task, it works as follows: (1) Question-worthy spans (s), which are noun phrases and named entities, are extracted from a model-generated summary. (2) For each s, a question is generated by QG based on s and the summary. (3) The QA model predicts an answer a based on the question and the source document. (4) The similarity between s and a is measured by some metric. (5) The factual consistency of the summary is made based on the similarity scores for all s in it.

We use the learned metric `LERC (QuIP)` mentioned in the paper and report a factual error if the similarity score between s and a is smaller than a threshold T_{qa} (a hyper-parameter). Question-worthy spans of different semantic roles correspond

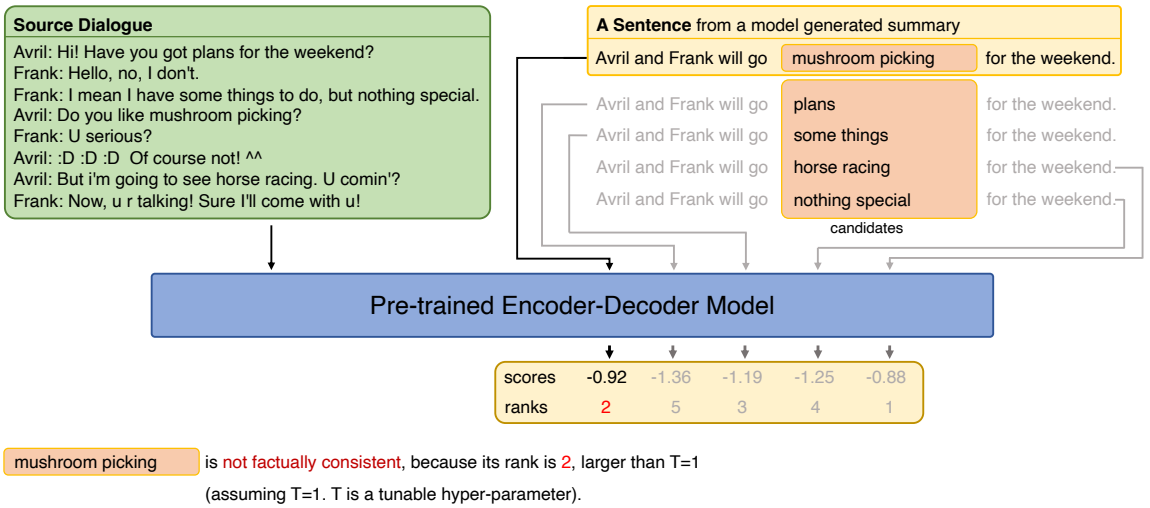

Figure 4: The workflow of our ENDERANKER model.

to our semantic role-based factual error classes, as outlined in Algorithm 1 in Appendix. We obtain the semantic role of a question-worthy span by a pre-trained structured prediction model in AllenNLP 2.9.3.[7]

WEAKLY-SUPERVISED-CLASSIFIER is a multi-class classifier that we construct. It takes as input a source dialogue and a generated summary sentence to predict factual error classes in the sentence, motivated by Kryscinski et al. (2020). We create synthetic training data by corrupting sentences in reference summaries as follows.

For Entity Error, Circumstance Error and Coreference Error, we replace named entities or pronouns with those randomly picked from the same category. For Predicate Error, we replace verbs with other randomly chosen verbs. We match the form (e.g., tense) of the selected verbs to the original one. Negative replacements for all above classes are extracted from either the source dialogue or the whole dataset. For Link Error, we replace a discourse marker corresponding to causal relation (e.g., because) with another one indicating a reversed causal relation (e.g., so). More details on our synthetic data generation are in Appendix A.3.1.

We use cross entropy loss to train the classifier, which is based on BERT (Devlin et al., 2019) with a linear layer on top of [CLS] representation for classification. We concatenate the source dialogue and a sentence, delimited by [SEP], as input.

[7]We use *structured-prediction-srl-bert* and choose the semantic role of the shortest span containing s.

4.2 ENDERANKER

Here, we present our proposed unsupervised model, ENDERANKER. Given a generated summary sentence, it first identifies a set of *spans of interest* (SOI) which may correspond to factual errors. For each SOI, ENDERANKER replaces it with different candidate spans and calculates a score for each span including the SOI. The factuality of the SOI is then decided based on its score among the scores of all candidate spans. Figure 4 summarizes the workflow of ENDERANKER. Below we detail core steps of ENDERANKER: (1) *SOI identification*, (2) *candidate span generation*, (3) *span scoring* and (4) *ranking-based factual error detection*.

Span of interest identification. An SOI is a snippet in a sentence for factual error classification. We consider noun phrases, named entities and verbs as SOIs, which are obtained using spaCy 3.1.4.[8] We obtain the semantic roles of the SOIs like for QAFACTEVAL, which will be used to decide the error class of an SOI later.

Candidate span generation. For each SOI, we create a set of candidate spans that can potentially replace it in the model generated summary sentence. For a named entity SOI, the candidate spans are entities of the same named entity class (e.g., **PERSON**) of the SOI extracted from the input dialogue. For the **PERSON** class, in particular, we include all speaker names on top of all other **PERSON** named entities extracted. For a verb SOI, we extract all verbs from the input dialogue according

[8]https://spacy.io/

Model	NoE	EntE	CirE	PredE	CorefE	Others	Micro Avg	Macro Avg
Adapted state-of-the-art models								
QAFACTEVAL	$0.68_{0.04}$	$\underline{0.45}_{0.03}$	$\underline{0.23}_{0.11}$	$0.00_{0.00}$	$0.11_{0.06}$	$0.00_{0.00}$	$0.51_{0.03}$	$0.25_{0.02}$
DAE	$0.77_{0.02}$	$0.32_{0.05}$	$0.03_{0.06}$	$0.00_{0.00}$	$0.00_{0.00}$	$\underline{0.34}_{0.11}$	$0.59_{0.02}$	$0.24_{0.02}$
Weakly Supervised multi-class classifier								
BERTMULTI	$0.72_{0.00}$	$0.20_{0.00}$	$0.08_{0.00}$	$0.09_{0.00}$	$\underline{0.29}_{0.00}$	$0.08_{0.00}$	$0.54_{0.00}$	$0.24_{0.00}$
ENDERANKER (ours)								
BART-LARGE-CNN	$0.67_{0.06}$	$0.34_{0.07}$	$0.04_{0.06}$	$0.15_{0.04}$	$0.12_{0.10}$	$0.00_{0.00}$	$0.47_{0.07}$	$0.22_{0.01}$
BART-LARGE-SAMSUM	$0.67_{0.06}$	$0.35_{0.08}$	$0.03_{0.04}$	$0.21_{0.06}$	$0.21_{0.13}$	$0.00_{0.00}$	$0.47_{0.05}$	$0.24_{0.02}$
PEGASUS-CNN	$0.71_{0.03}$	$0.37_{0.08}$	$0.04_{0.05}$	$0.18_{0.05}$	$0.14_{0.09}$	$0.00_{0.00}$	$0.52_{0.04}$	$0.24_{0.01}$
PEGASUS-SAMSUM	$0.67_{0.04}$	$0.37_{0.09}$	$0.06_{0.07}$	$0.19_{0.06}$	$0.16_{0.11}$	$0.01_{0.02}$	$0.46_{0.05}$	$0.24_{0.01}$
T5-LARGE-CNN	$0.68_{0.04}$	$0.35_{0.09}$	$0.03_{0.04}$	$0.15_{0.04}$	$0.06_{0.03}$	$0.01_{0.02}$	$0.47_{0.05}$	$0.21_{0.02}$
T5-LARGE-SAMSUM	$0.70_{0.08}$	$0.35_{0.10}$	$0.04_{0.05}$	$\underline{0.22}_{0.08}$	$0.14_{0.03}$	$0.00_{0.00}$	$0.51_{0.09}$	$0.24_{0.03}$
Ensemble learning (**including our ENDERANKER model**)								
FREQVOTING	$0.79_{0.03}$	$0.40_{0.05}$	$0.05_{0.11}$	$0.10_{0.08}$	$0.12_{0.10}$	$0.01_{0.02}$	$\underline{0.62}_{0.03}$	$0.24_{0.03}$
LOGISTIC	$\underline{0.80}_{0.03}$	$0.44_{0.05}$	$0.20_{0.13}$	$0.00_{0.00}$	$0.11_{0.10}$	$0.03_{0.03}$	$0.61_{0.03}$	$\underline{0.26}_{0.04}$

Table 3: F1 scores for factual error detection models with a break down on each error class based on our annotated dataset DIASUMFACT. We report the average score and standard deviation over 5-fold cross validation. **Link Error (LinkE)** is merged into **Others** because almost no model can detect it. The best score for each column is underlined.

to the Part-of-Speech tags and match the form (e.g., tense) with the SOI. For a noun phrase SOI, all noun phrases from the input dialogue are considered as candidate spans. All candidate spans are extracted using spaCy 3.1.4.

Span scoring. Let D be an input dialogue and S be a generated summary sentence with n tokens $\{w_1, w_2, \cdots, w_{n-1}, w_n\}$, which includes a candidate span or an SOI, denoted by c. We adopt a encoder-decoder model $\mathbb{M}$ to calculate a sentence score for S conditioned on D as follows, which is used as the score of span c, denoted by score_c. $\mathbb{M}$ can be any pre-trained encoder-decoder model, such as a summarization model.

$$\text{score}_c = \frac{1}{n}\sum_{i=1}^{n} \log p(w_i | w_{<i}, D) \qquad (1)$$

Intuitively, the score is the average log likelihood of each token w_i in S, conditioning on the previous tokens in S (i.e., $w_{<i}$) and D. Here, w_0 is the starting token of the decoder.

Ranking-based factual error detection. Given a set of candidate spans $C = \{c_1, c_2, \cdots, c_{|C|}\}$ of an SOI, we form $|C|$ sentences by replacing the SOI with each of the candidate spans. We calculate span scores for the SOI and the candidate spans, and rank the spans by their scores in descending order. If the SOI has a rank larger than a threshold T (a hyper-parameter), we report it as erroneous and determine its error class based on its semantic role, as summarized in Algorithm 1 (cf. Appendix). The same process is repeated for all SOIs in S. The union of all error classes detected for the SOIs is the final factual error classes predicted for S.

4.3 Ensemble Modeling

We further build two simple ensemble models based on the four models above: Most **Freq**uent **Voting** (FREQVOTING) and **Logistic regression** (LOGISTIC). FREQVOTING takes all predicted error classes from the four models above and uses the class(es) with the largest frequency as the final prediction. For LOGISTIC, we train a logistic regression model for each factual error class that takes the binary outputs from the four models above as features. We use the union of all factual error classes predicted by the different logistic regression models as the final prediction.

4.4 Experiments

To evaluate the models described in the last section, we perform 5-fold cross validation (Stone, 1978) using DIASUMFACT.[9] Implementation details and parameter settings are discussed in Appendix A.3. We record the F1 scores (mean and standard deviation) of the models on each error class in Table 3.

[9] As it gives more reliable results considering the size of our dataset, compared to a usual train/test split.

Results: All models can detect EntE significantly and consistently better than the other classes. Different models show advantage on different error classes, while no model can outperform all the others on all error classes.

QAFACTEVAL performs the best on EntE (0.45) and CirE (0.23) but poorly on the other error classes. The reason is that only named entities and noun phrases are treated as question-worthy spans. Future work may consider question-worthy spans of different types, such as verbs and discourse markers, to cover more error classes.

DAE performs well on EntE and Others, while it suffers on CirE, PredE and CorefE. The main reason is that not all error classes are covered in the rules mapping from dependency arc to error class. Since a dependency arc is related to two words, rule designing is not easy. Future work may leverage learned models to predict error class automatically.

BERTMULTI shows the best results on CorefE (0.29) but poor performance on CirE, PredE and Others, despite its high performance on synthetic validation dataset (0.98 accuracy). It indicates the difference between synthetic and real factual errors.

Our proposed model ENDERANKER using different pretrained encoder-decoder models generally exhibits strong results on EntE, PredE and CorefE, while more improvements need to be done on CirE and Others. Among all variants of ENDERANKER, PEGASUS-CNN performs on par with QAFACTEVAL in terms of macro-averaged F1 score, while it does not require question generation and question answering models.

The two ensemble models improve on the micro and macro-averaged F1, indicating complementarity among the models. For most error classes, the ensemble models usually have the best or second best performance.

Overall, none of the models yielded a particularly high F1 score for any error class. It shows that fine-grained factual error detection in dialogue summaries is a challenging problem which calls for further studies, for which our results and dataset will serve as a solid foundation.

5 Conclusions

We created a fine-grained multi-faceted dataset named DIASUMFACT on factual consistency of dialogue summarization. DIASUMFACT offers insights into how and where current neural summarization models fail when they produce factually inconsistent details in dialogue summaries. It can also serve as a testbed for automating factual error detection. Our proposed error detection method, ENDERANKER, is shown to perform on par with state-of-the-art models even though it requires no labelled training data. That said, we ultimately found that even ensembling several error detection methods do not produce results that are good enough for practical use, indicating opportunities for future research in this area.

6 Limitations

ENDERANKER is only tested on DIASUMFACT. Further tests on more datasets are required to establish its general applicability.

7 Ethics Statement

This study is conducted under the guidance of the ACL code of Ethics. We manually filtered out potential offensive content and removed all information related to the identification of annotators. The annotators are all fairly paid based on the Australian minimum wage. The annotation protocol is approved under Human Ethics LNR Application with reference number 2022-24233-30104-3.

Acknowledgements

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200. We want to thank Gisela Vallejo, Han Sun, Miao Li, Rui Xing, Wei Gao, Yanchuan Chang, Yulia Otmakhova, Zheng Wei Lim, Zhexi Li, Zhuohan Xie for their help in the annotation.

References

Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan McDonald. 2021. Focus attention: Promoting faithfulness and diversity in summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6078–6095, Online. Association for Computational Linguistics.

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. *arXiv preprint arXiv:2210.12378*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Meng Cao, Yue Dong, and Jackie Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021a. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. *arXiv preprint arXiv:2104.09061*.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021b. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. 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*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

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

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1479–1491, Online. Association for Computational Linguistics.

Mingqi Gao and Xiaojun Wan. 2022. DialSummEval: Revisiting summarization evaluation for dialogues. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5693–5709, Seattle, United States. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment.

In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. 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*, pages 1449–1462, 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. *Advances in neural information processing systems*, 28.

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

Anirudh Joshi, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2020. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3755–3763, Online. Association for Computational Linguistics.

Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396.

Kundan Krishna, Sopan Khosla, Jeffrey Bigham, and Zachary C. Lipton. 2021. Generating SOAP notes from doctor-patient conversations using modular summarization techniques. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4958–4972, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1957–1965.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. Topic-aware contrastive learning for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1229–1243, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengyuan Liu and Nancy Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, 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, Berlin, Germany. Association for Computational Linguistics.

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

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

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

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

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Mervyn Stone. 1978. Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics*, 9(1):127–139.

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

Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022b. CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5657–5668, Seattle, United States. Association for Computational Linguistics.

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

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Zhiyuan Zeng, Jiaze Chen, Weiran Xu, and Lei Li. 2021. Gradient-based adversarial factual consistency evaluation for abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4102–4108, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249, Online. 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Yicheng Zou, Jun Lin, Lujun Zhao, Yangyang Kang, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021. Unsupervised summarization for chat logs with topic-oriented ranking and context-aware auto-encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14674–14682.

A Implementation Details

A.1 Cross Validation Settings

We randomly split DIASUMFACT into 5 portions with equal number of examples and keep the splits

consistent across all models. Each time we take one portion as the test set and combine the other four portions for training or validation, or both. The details for the evaluation of each model are described below.

- BERTMULTI, DAE and FREQVOTING: there is no hyper-parameter to tune. The model is only evaluated on different test sets for 5 times.

- QAFACTEVAL and ENDERANKER: they are unsupervised models so no training is needed. Each time the four portions are combined as validation set for hyper-parameter tuning.

- LOGISTIC: since this model requires supervised training, we combine the four portions, shuffle it and further split it into training set and validation set, following a ratio of 7:3. The validation set is used for hyper-parameter tuning.

A.2 Summary Generation Models

GPT-3: we use a batch size of 64 and fine-tune it for 2 epochs. During inference, `temperature` is set to 1.0 and `max_tokens` is set to 100. The finetuned model achieves 41.7 and 15.9 on ROUGE-1 and ROUGE-2.

DIALOGLM: we finetune *MingZhong/DialogLED-large-5120* proposed in the original paper[10]. We finetune it for 5 epochs using a batch size of 32 (per-device batch size is 2, gradient accumulation is 16) and learning rate 3×10^{-5}. The fine-tuning takes 30 minutes. The finetuned model achieves 38.48 and 13.70 on ROUGE-1 and ROUGE-2, which are higher than 34.50 and 9.92 reported in the original paper.

PEGASUS: we finetune *google/pegasus-cnn_dailymail* for 5 epochs using a batch size of 32 (per-device batch size is 2, gradient accumulation is 16) and learning rate 3×10^{-5}. The fine-tuning takes 15 minutes. The finetuned model achieves 33.56 and 11.35 on ROUGE-1 and ROUGE-2.

BART: we finetune *facebook/bart-large-cnn* for 5 epochs using a batch size of 32 (per-device batch size is 2, gradient accumulation is 16) and learning rate 3×10^{-5}. The fine-tuning takes 25 minutes. The finetuned model achieves 40.46 and 14.93 on ROUGE-1 and ROUGE-2.

[10] https://github.com/microsoft/DialogLM

All original models come from huggingface model hub[11]. The fine-tuning for BART, PEGASUS and DIALOGLM is conducted using *run_summarization.py* from Transformers [12] 4.14.0.

During training, the input is the concatenation of the query and its relevant utterances, which is a subset of the whole meeting transcript. Utterances are concatenated as a long string, the query and utterances are delimited by "||".

A.3 Error Detection Models

A.3.1 WEAKLY-SUPERVISED-CLASSIFIER

To obtain corrupted reference sentences with Entity Error, Coreference Error and Predicate Error, we first extract named entities, noun phrases and verbs using spaCy 3.1.4, then get their semantic roles like for QAFACTEVAL in Section 4.1. We finally map from semantic role to factual error class according to Algorithm 1.

We generate 80k negative examples for each error class, among which 75k are used for training and 5k for validation. For EntE, PredE and CirE, the negative replacements for half of the data come from the same dialogue, while another half of the data uses negative replacements extracted from the whole dataset excluding the dialogue corresponding to the sentence. In this case we include both intrinsic and extrinsic negative replacements. Sentences from reference summaries are used for No Error.

We use *run_glue.py* from Transformers 4.14.0 for model training. The pretrained model we use for BERT is *bert-base-uncased*.

We tune batch size among 16, 32, 64 and 128. The best value is 64 according to the accuracy on validation set (98.24%). The model is trained for 8 epochs and evaluated every 500 steps. The learning rate we use is 3×10^{-5}. The training takes 8 hours on a Tesla V100 GPU with 32GM RAM.

A.3.2 ENDERANKER

The details of the pretrained models that we use are as follows:

BART-LARGE-CNN: *facebook/bart-large-cnn*

BART-LARGE-SAMSUM: *lidiya/bart-large-xsum-samsum*

PEGASUS-CNN: *google/pegasus-cnn_dailymail*

[11] https://huggingface.co/models

[12] https://huggingface.co/docs/transformers/index

PEGASUS-SAMSUM: *transformersbook/pegasus-samsum*
T5-LARGE-CNN: *sysresearch101/t5-large-finetuned-xsum-cnn*
T5-LARGE-SAMSUM: We fine-tune it using *run_summarization.py* from Transformers 4.14.0 based on *sysresearch101/t5-large-finetuned-xsum-cnn*. The final batch size is 2 with a gradient accumulation steps of 16 (i.e., the conceptual batch size is $2 \times 16 = 32$). The model is trained for 8 epochs on a single NVIDIA A100 (40G) GPU, taking 5 hours. We choose the batch size 32 among [8, 16, 32] because it produces the highest ROUGE-1 and ROUGE-2 on validation set.

A.3.3 DAE

We use the trained classifier provided by the authors of the DAE model[13] and process each sentence in a model-generated summary separately. A dependency arc is considered as erroneous if the predicted probability for the positive class is less than 0.5.

A.3.4 QAFACTEVAL

We use the model provided by the authors[14] and retrieve the similarity score between ground truth answers and predicted answers from logs, given by the learned model `LERC (QuIP)`. We tune the threshold T_{qa} among $[0.5, 1.0, 1.5, 2.0]$ and choose 0.5 as the final value, as it produces the highest macro-averaged F1 score.

The process to map from semantic role to factual error class is outlined in Algorithm 1.

A.3.5 ENDERANKER

We tune T (i.e., the rank threshold) among [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and choose the smallest value that achieves the highest macro-averaged F1 on the validation set. The best T values for different pre-trained models are as follows:

- BART-LARGE-CNN: 2
- BART-LARGE-SAMSUM: 2
- PEGASUS-CNN: 3
- PEGASUS-SAMSUM: 2
- T5-LARGE-CNN: 3
- T5-LARGE-SAMSUM: 3

[13] https://github.com/tagoyal/dae-factuality
[14] https://github.com/salesforce/QAFactEval

Algorithm 1 Semantic Role to Factual Error Class. arg0 to arg5 are core semantic roles such as subject and object. 'ARGM' is the prefix for non-core semantic roles such as ARGM-TMP (temporal modifier). V represents 'verb'.

Require: s ▷ a Span-of-Interest
Require: sr ▷ the semantic role of s
$pronouns \leftarrow$ [i, we, us, you, he, him, she, her, it, they, them, this, that, these, those, myself, yourself, himself, herself, ourselves, yourselves, themselves]
if sr in [arg0, arg1, arg2, arg3, arg4, arg5] **then**
  if $s \in pronouns$ **then**
    Return CorefE
  else
    Return EntE
  end if
else if sr contains 'ARGM' **then**
  Return CirE
else if $sr =$ 'V' **then**
  Return PredE
else
  Return Others
end if

To avoid repeated encoding for the same dialogue, which corresponds to multiple sentences for factual error detection, we cache the encoded representation in encoder and reuse them to improve inference speed.

The experiments are conducted on a single Nvidia V100 GPU with 16GM RAM. The inference over a full pass of our dataset takes around 40 hours with a batch size of 1. The computaional overhead can be reduced by (1) reducing the number of Span of Interest (SOI) in a sentence, and (2) reducing the number of candidates, especially for noun phrases. We also tried distilled encoder-decoder models, but the results are sub-optimal.

A.3.6 Ensemble Learning

For ensemble models (i.e., FREQVOTING and LOGISTIC), the best ENDERANKER is chosen based on the performance on the validation set, which is 30% of the four portions combined except the test set, as introduced in A.1.

During the training of LOGISTIC, we upsample the minority class to match the number of the majority class for each logistic regression model corresponding to different factual error types.

Dataset	#Mod	#Summ	#Sen	Domain	Annotation Typology
FactCC (Kryscinski et al., 2020)	10	/	1,434	news	binary (consistent, inconsistent)
QAGS (Wang et al., 2020)	2	474	/	news	binary (consistent, inconsistent)
SummEval (Fabbri et al., 2021)	44	12,800	/	news	5-point Likert scale
Polytope (Huang et al., 2020)	10	1,500	/	news	multi-class
Cao'22 (Cao et al., 2022)	1	800	/	news	multi-class
Maynez'20 (Maynez et al., 2020)	5	500	/	news	binary (intrinsic, extrinsic)
Frank (Pagnoni et al., 2021)	8	2,250	4,942	news	multi-class
Goyal'21 (Goyal and Durrett, 2021)	3	50	/	news	multi-dimensional, multi-class
CLIFF (Cao and Wang, 2021)	2	600	/	news	multi-class
ConFIT (Tang et al., 2022b)	4	76	/	dialogue	multi-class
DialSummEval (Gao and Wan, 2022)	13	4,200		dialogue	5-point Likert Scale
DIASUMFACT (ours)	6	475	1,340	dialogue	multi-dimensional, multi-class

Table 4: Datasets that focus on or include factual consistency for summarization. #Mod: the number of summarization models covered. #Summ: the number of model-generated summaries covered. #Sen: the total number of sentences in model-generated summaries.

B Data Annotation

B.1 Error Typology

For CorefE, if a reference comes without antecedents in the input dialogue, we ignore the error in the summary.

B.2 Annotation Tool

We modify a web application developed originally for FRANK (Pagnoni et al., 2021)[15] to fit to our task. Specifically, we replace the example article and model summaries with an example dialogue and manually composed summaries to help explain different error types. We also add an input field for error span annotation in the main page. Screenshots are shown in Figures 6, 7, 8 and 9.

For in-house annotation, we deploy the web application on Firebase[16] and provide with annotators URLs to the tasks directly.

[15] https://github.com/artidoro/frank-annotation-platform

[16] https://firebase.google.com/

B.3 Annotation Procedure

The initial annotation by all annotators follows the typology proposed by Pagnoni et al. (2021), which includes two additional classes: Out-of-Article Error (i.e., Extrinsic Error in our paper) and Grammar Error. We merge Grammar Error to Others, and treat Extrinsic Error as a separate dimension, as outlined in 3.3.

B.4 Payments to Annotators

All our annotators are volunteers. We pay 100 AUD to each annotator. The annotation task begins after they agree to the amount of payment.

B.5 Demographic Characteristics of Annotators

1 annotator come from Colombia, 1 annotator comes from Russia, 1 annotator come from Malaysia, 9 annotators come from China. There are 6 female and 6 male annotators.

B.6 Consent from Annotators

We show the consent form in the annotation web application. Annotation can only begin after consent form is received from annotators.

C Case Study

As shown in Figure 5, our ENDERANKER successfully identifies an error of the span "The team" because its rank is larger than the threshold $T = 3$. Since the semantic role of the span is *arg0*, the model predicts Entity Error according to Algorithm 1. On the right-side example, ENDERANKER fails to report the error of "muchroom picking", although the factual consistent span "horse racing" is ranked at the top among candidates. The reason is that T is too large. For future work, we may design error identification methods using SOI-specific thresholds rather than a universal threshold for all SOIs.

D Potential Risks

The factual error detection models we propose, which are BERTMULTI and ENDERANKER, do not produce satisfactory performance to be used for real applications. We do not advise people to use them directly in real applications as factual error detectors for dialogue summarization without further improvements.

E Intended Use of Existing Artifacts

The SAMSum (Gliwa et al., 2019) dataset is shared on terms of the AttributionNonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license. We provide additional information (i.e., model-generated summaries and human annotations) without modifying the original data (i.e., dialogues and reference summaries).

Data source	SAMSUM	QMSUM
#Exs	757	583
T_D	148.4	355.7
U_D	12.3	16.2
T_{Sen}	11.3	25.7
S_{Summ}	2.6	3.2
T_Q	/	14.9

Table 5: Statistics of our dataset. #Exs: the number of (dialogue, sentence) pairs. T_D: the average number of tokens in a dialogue. U_D: the average number of utterances in a dialogue. T_{sen}: the average number of tokens in a summary sentence. S_{Summ}: the average number of sentences in a model-generated summary. T_Q: the average number of tokens in a query.

Error Type	Frequency
No Error	853
EntE	256
PredE	106
CirE	48
CorefE	62
LinkE	41
Others	42

Table 6: The number of human-detected errors in each error type along semantic role dimension.

The team agreed to have wood for the bottom and plastic for the base, but it was not exactly right for the spongy point of view.

Rank	Candidate	Score
1	industrial designer	-2.16
2	the design	-2.23
3	wood	-3.00
	...	
12	participant	-3.17
13	the team	-3.18
	...	

label: Entity Error
prediction: Entity Error
correctness: ✓

Frank will go mushroom picking with Avril.

Rank	Candidate	Score
1	horse racing	-2.16
2	mushroom picking	-2.23
3	the weekend	-3.51
4	plans	-3.73
	...	
7	nothing	-4.36
8	avril	-4.74

label: Entity Error
prediction: No Error
correctness: ✗

Figure 5: Case study for ENDERANKER where it identifies an error correctly in the example on the left, but fails in the right-side example. The rank threshold T=3. The SOIs are highlighted both in the original sentence and in the candidates list sorted by score in descending order.

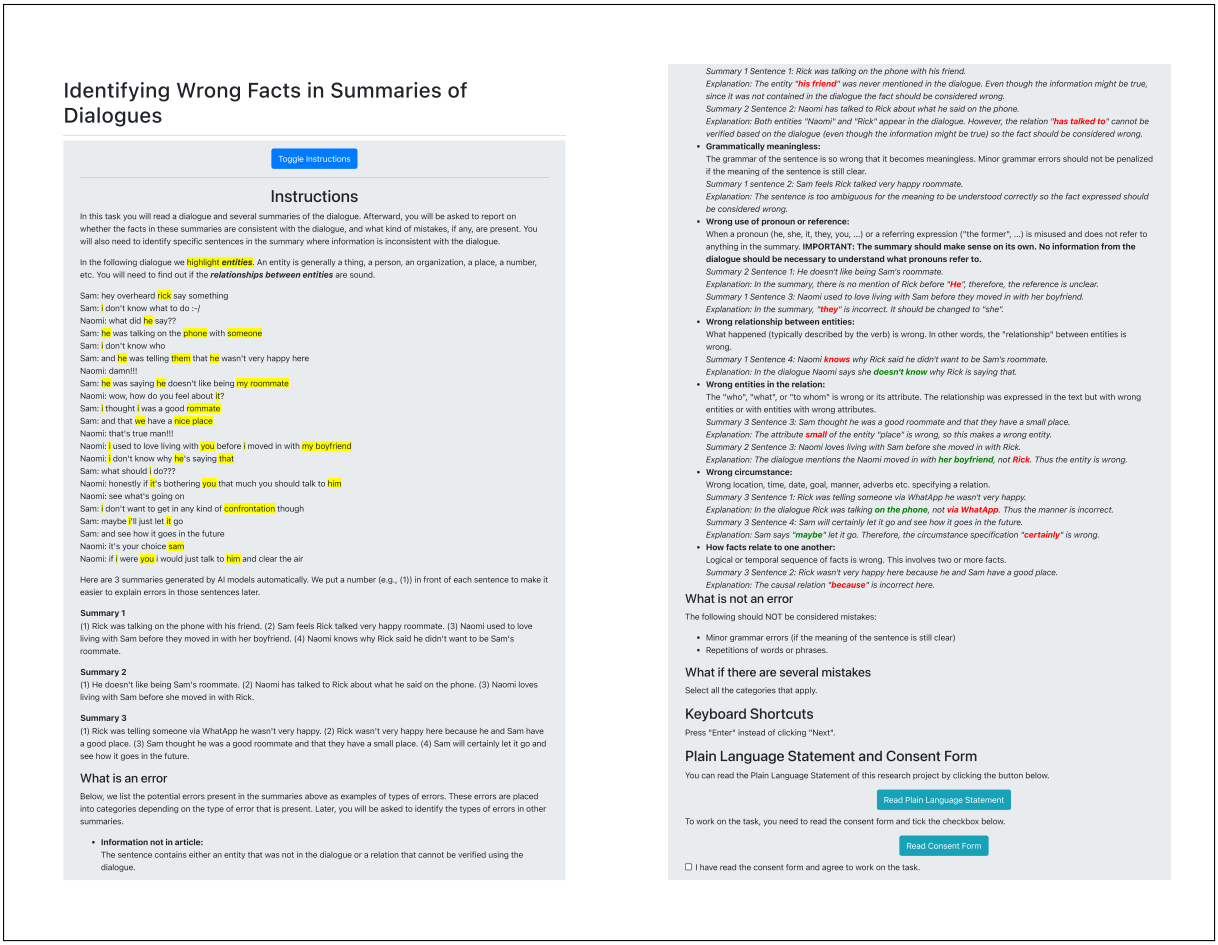

Figure 6: The instruction page of our annotation tool, where an example dialogue, definitions and examples of different types of factual errors are shown. The plain language statement and consent form are at the bottom.

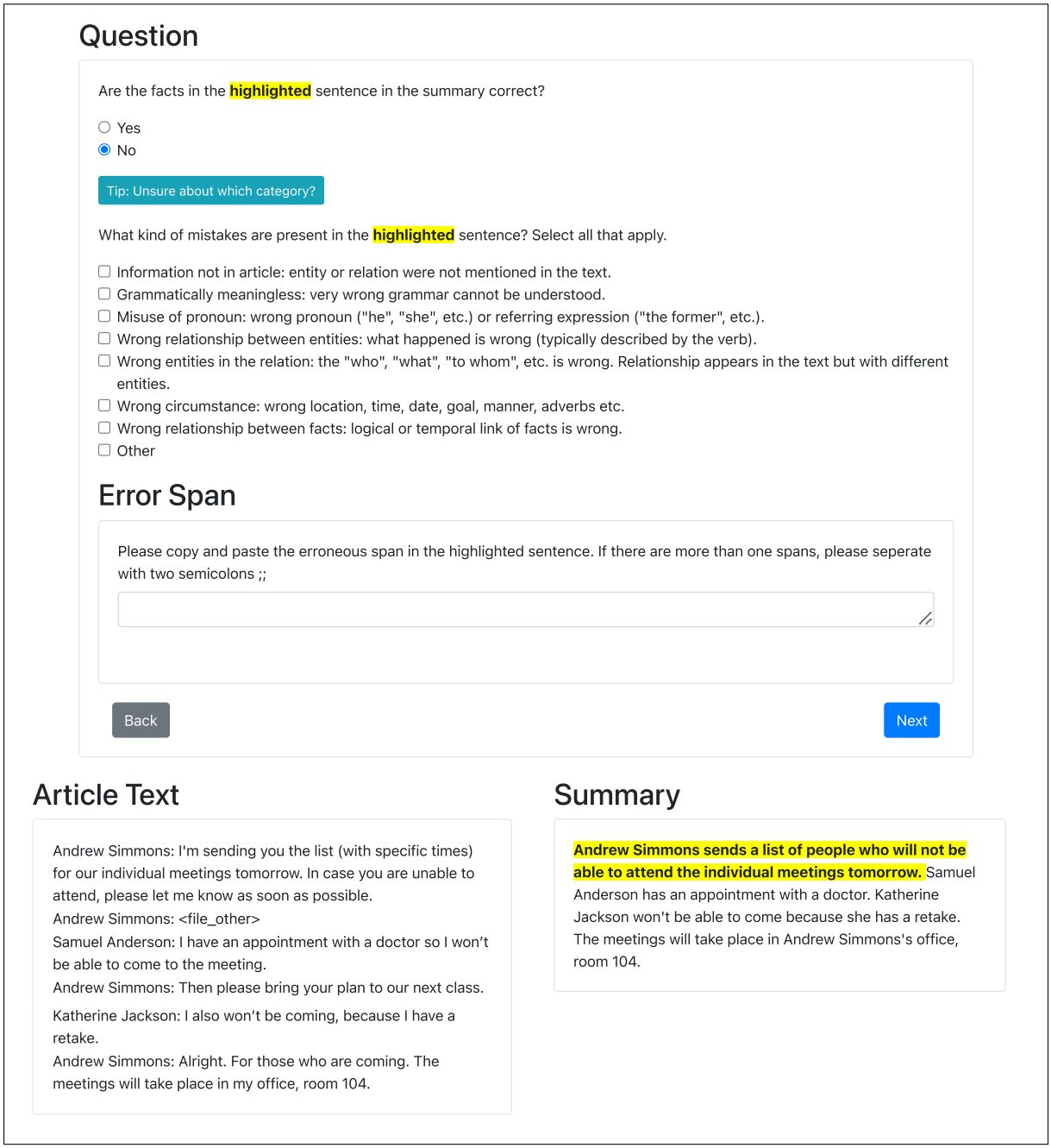

Figure 7: The main page of our annotation tool.

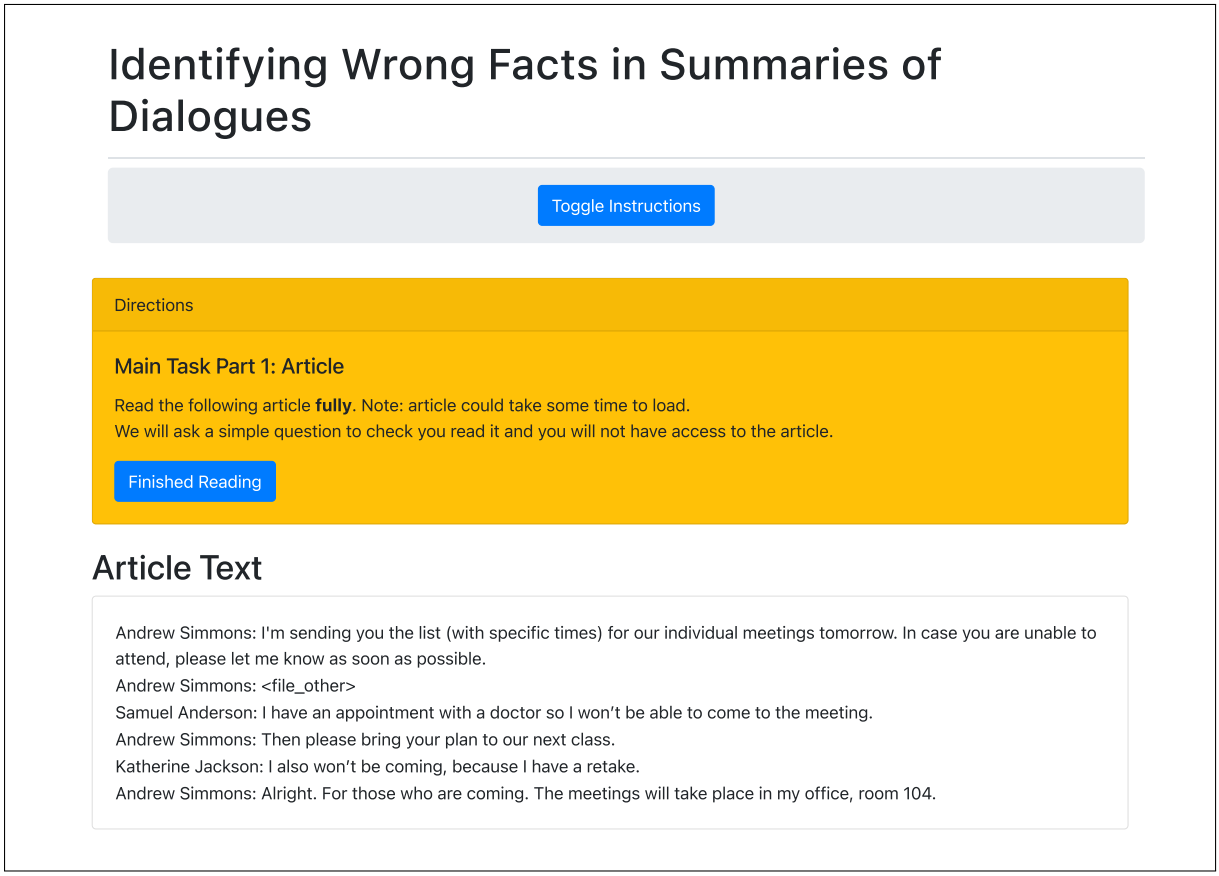

Figure 8: The entity question page (part 1) of our annotation tool. Annotators are required to answer the entity question first to make sure they read the dialogue carefully.

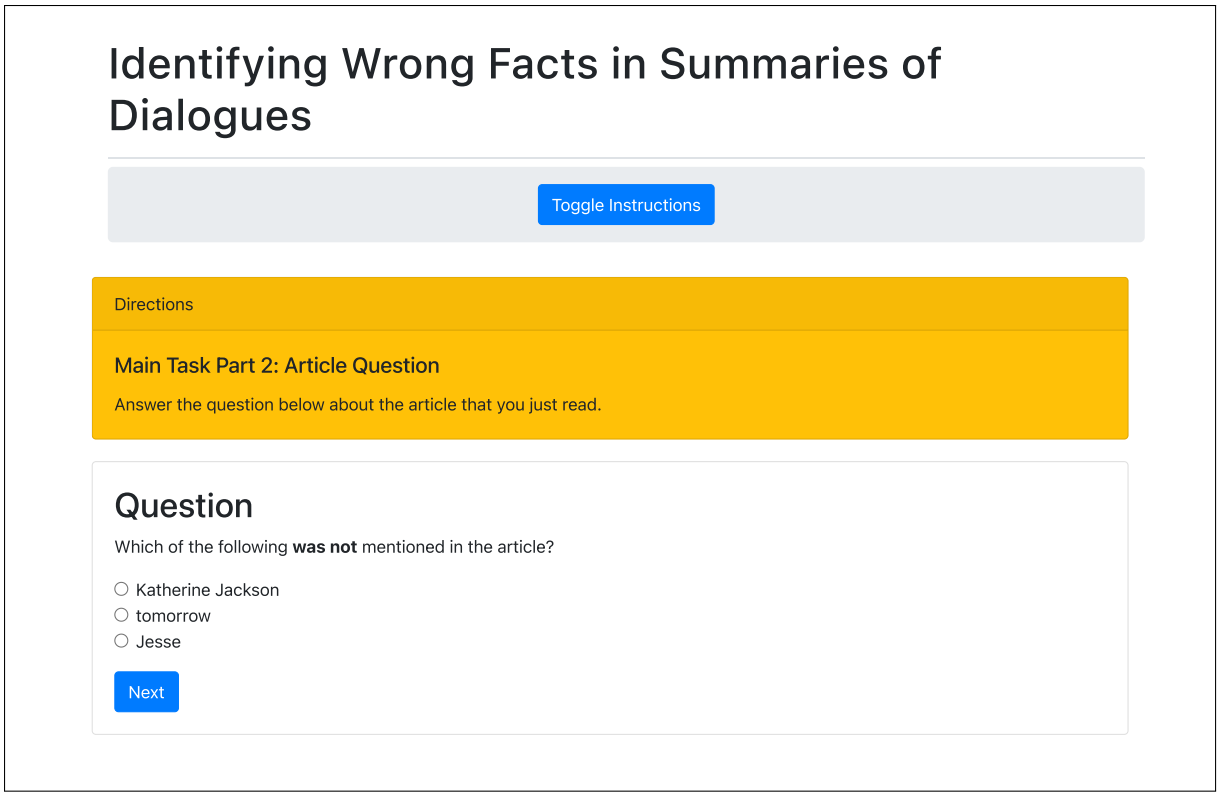

Figure 9: The entity question page (part 2) of our annotation tool. Annotators are required to answer the entity question first to make sure they read the dialogue carefully.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section Limitations between conclusion and References

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

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

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

B ☑ Did you use or create scientific artifacts?

Our created artifacts are provided in an anonymous github repository. The artifacts we use are mentioned in Section 3.1; Section 4.1; Appendix A.2 and A.3.

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1; Section 4.1; Appendix A.2 and A.3.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license is included in the anonymous github repositories.

☑ 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)?
Appendix 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?
The end of Section 3.1 mentions how we filter out offensive contents. The Ethics Statments section mentions that we protect annotators privacy.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The documentation of our data and code are provided in the anonymous github repositories, mentioned in the footnote of abstract.

☑ 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.
Introduction, Table 5, Table 4, Appendix A.3.1

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

C ☑ Did you run computational experiments?

Section 4.4, Appendix A

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.2, A.3

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix A.2, A.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?
Figure 2, Figure 3, Table 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.2, Appendix A.3.1

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

Section 3.4

☑ 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.?
Figure 6, 7, 8 and 9 in Appendix.

☑ 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)?
Appendix B.4, Ethics Statement.

☑ 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?
Appendix B.6.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Ethics Statement

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix B.5.