Towards Understanding Omission in Dialogue Summarization

Anonymous ACL submission

Abstract

Dialogue summarization aims to condense the lengthy dialogue into a concise summary, and has recently achieved significant progress. However, the result of existing methods is still far from satisfactory. Previous works indicated that omission is a major factor in affecting the quality of summarization, but few of them have further explored the omission problem, such as how omission affects summarization results and how to detect omission, which is critical for reducing omission and improving summarization quality. Moreover, analyzing and detecting omission relies on summarization datasets with omission labels (i.e., which dialogue utterances are omitted in the summarization), which are not available in the current literature. In this paper, we propose the OLDS dataset, which provides high-quality Omission Labels for Dialogue Summarization. By analyzing this dataset, we find that a large improvement in summarization quality can be achieved by providing ground-truth omission labels for the summarization model to recover omission information, which demonstrates the importance of omission detection for omission mitigation in dialogue summarization. Therefore, we formulate an omission detection task and demonstrate our proposed dataset can support the training and evaluation of this task well. We also call for research action on omission detection based on our proposed datasets. Our dataset and codes are publicly available [1].

1 Introduction

With the exponential increase in the volume of conversational messages from daily life, there is a growing demand for dialogue summarization (Murray and Carenini, 2008; Gliwa et al., 2019; Chen and Yang, 2020; Zhong et al., 2021; Zou et al., 2021c), which compresses lengthy interactions into a more concise and structured piece of text while preserving the most important and relevant information. Recent years have witnessed significant progress in abstractive dialogue summarization, especially using large-scale pre-trained language models (Lewis et al., 2020; Raffel et al., 2020). Despite the advances in a high level of fluency and coherence, existing models are still prone to generate defective summaries (Kryściński et al., 2019; Maynez et al., 2020; Tang et al., 2022) that limit their practical usage. Previous works (Chen and Yang, 2020; Liu et al., 2021; Tang et al., 2022) have investigated the taxonomy of errors involved in output summaries, and human evaluations revealed that the majority of errors fall into the category of omission, which often leads to incomplete summaries where critical facts are lost. However, few of these works have further analyzed the omission problem, let alone addressing this problem.

To reduce omission rate and improve summarization quality, a comprehensive analysis on omission problem (e.g., how omission affects summary results) and a precise omission detection (i.e., to locate which dialogue utterances are omitted in the summarization) is important. However, there are no omission related datasets in dialogue summarization literature to support such analysis and detection. Hence, in this work, we construct the OLDS dataset, which provides high-quality Omission Labels for Dialogue Summarization. Our dataset is built upon five existing benchmarks covering different domains. For each dialogue, we use different abstractive models to generate diverse candidates and propose a reference-based strategy to automatically label omissions for these candidates. The human evaluation indicates that our OLDS dataset presents a high quality of omission labels.

Based on the curated OLDS dataset, we comprehensively investigate the omission problem in dialogue summarization from multiple aspects. **First**, we analyze the proportion of candidates with omission errors and the position distribution of omitted

[1] The link is omitted due to the review version.

information in dialogues. The results reveal that omission is a severe problem that frequently occurs in dialogue summarization. **Second**, we measure the correlation between the omission rate and multiple reference-based metrics (e.g., ROUGE and BERTScore), discovering that omission is one of the decisive factors influencing the summary evaluation results. **Third**, we explore the potential performance improvement brought by utilizing the omission information in a post-editing manner. The analyses probe that candidate summaries could be effectively improved as long as the model is provided with the omitted dialogue utterances. Hence, how to accurately locate omission information in dialogue naturally becomes a critical question.

To pave the way to omission mitigation and summary improvement, we formulate the task of omission detection, which aims to identify the omitted utterance given the whole dialogue utterances and the generated summary with potential omission. In addition, we present three different frameworks as baselines for the omission detection task, including pair-wise classification, sequence labeling, and pointer network extraction. Experimental analyses on the OLDS dataset reveal that omission detection, as a promising direction to assessment and improvement for dialogue summarization, poses significant values and challenges.

The contributions of our paper are as follows:

- We propose OLDS, a dataset with high-quality omission labels for dialogue summarization, to facilitate the research on the omission problem.
- Based on OLDS, we systematically analyze the omission problem and demonstrate the significance of omission in dialogue summarization.
- We introduce the omission detection task that paves the way to omission mitigation and summary improvement. We design 3 frameworks as baselines and conduct comprehensive analyses to provide possible directions for solving this task.

2 The OLDS Dataset

In this section, we first define what is omission. Then, we introduce OLDS, a dataset that contains Omission Labels for Dialogue Summarization that facilitates the analysis of omission problem and the exploration of how to identify omission content. Finally, we conduct human assessment that demonstrates the high quality of OLDS.

Dialogue:
(01) **Adam:** Have you talked to May?
(02) **Karen:** Yes, yesterday, why?
(03) Adam: I just talked to her and I must admit I **worry** about her.
(04) **Karen:** Me too, I suggested she should see a specialist, but she wasn't very happy about it.
(05) **Adam:** No wonder...
(06) **Karen:** I know, but I think this is serious. She's saying she's depressed, like everyone around, but in her case it may be true.
(07) **Adam:** She was telling me she doesn't feel like doing anything, she's bored all the time, she never feels happy. It sounds like a real, typical depression.
....
(12) **Adam:** Yes, but she doesn't want to see a specialist. Basically, she doesn't want to see anyone.
(13) **Karen:** Hm... I don't know... How about I call someone for advice? So we could know what to do.
(14) **Adam:** Sounds rational, do you know anyone you could call? Don't mention her name.
(15) Karen: Of course I won't! I have a friend who's a **psychologist**, we can trust her. I'll let you know.
(16) **Adam:** Thank you Karen!

Reference summary:
Adam and Karen are **worried** that May suffers from depression. Karen will call her friend who is a **psychologist** and ask for advice.

Candidate summary:
May is depressed. Karen suggested she should see a specialist, but she doesn't want to. Karen will call her friend for advice.

Omission utterances (Labels):
(03) (15)

Table 1: An example of the OLDS dataset. The dialogue is from SAMSum and the candidate summary is generated from $\text{BART}_{\text{large}}$. The salient words are underlined, and the omission information is highlighted in red.

2.1 The Definition of Omission

In summarization tasks, omission usually refers to the missing content in the candidates, which is presented in the gold reference. The definition of omission content is flexible, which could refer to either the omitted keywords, text spans, or utterances. In dialogues, an utterance could represent complete information compared to words or text spans and can be viewed as a basic semantic unit for identification and evaluation. Therefore, in this paper, we mainly focus on **utterance-level omission** and provide utterance-level labels. Table 1 shows an example of our OLDS dataset, which contains the original dialogue, reference summary, candidate summary, and omission labels. In this example, the candidate summary omits three key messages: the person "Adam", the attitude "worried" and the persona "psychologist", and thus the corresponding utterance-level omission labels are the 3rd and 15th utterances in the original dialogue.

2.2 Dataset Creation

OLDS is a dataset that collects multiple candidates for dialogue summarization and provides their cor-

Domain	Split	# of dialogs	Avg. turns	Len. of dialogs	Len. of summ.	# of candidate summaries (Avg. ROUGE-1 score)											
						BART_{L}		BART_{B}		T5_{B}		T5_{S}		Transformer		$\text{Pegasus}_{\text{L}}$	
SAMSum	Train	14,732	11.2	124.1	23.4	25,424	(50.6)	12,687	(48.2)	29,473	(47.2)	32,959	(41.0)	46,777	(37.7)	0	(-)
	Dev.	818	10.8	121.2	23.4	1,636	(54.4)	1,636	(51.1)	1,636	(51.0)	1,636	(44.2)	1,636	(39.2)	1,636	(51.1)
	Test	819	11.3	126.6	23.1	1,638	(52.6)	1,638	(49.2)	1,638	(49.3)	1,638	(43.5)	1,638	(37.9)	1,638	(50.4)
DialogSum	Train	12,460	9.5	187.5	31.0	20,766	(46.1)	10,132	(44.1)	26,897	(44.7)	37,056	(39.5)	29,749	(39.1)	0	(-)
	Dev.	500	9.4	185.0	29.0	1,000	(49.5)	1,000	(46.8)	1,000	(46.2)	1,000	(40.3)	1,000	(40.1)	1,000	(48.4)
	Test	500	9.7	192.5	28.4	1,000	(46.9)	1,000	(44.3)	1,000	(44.7)	1,000	(39.1)	1,000	(36.8)	1,000	(45.8)
EmailSum	Train	1,800	6.5	231.3	26.9	2,581	(33.2)	730	(33.8)	3,939	(33.0)	3,203	(29.7)	7,547	(24.4)	0	(-)
	Dev.	249	6.5	227.2	26.2	498	(37.8)	498	(36.6)	498	(36.1)	498	(34.0)	498	(24.8)	498	(35.9)
	Test	500	6.5	243.0	28.2	1,000	(37.0)	1,000	(36.2)	1,000	(35.3)	1,000	(32.4)	1,000	(25.7)	1,000	(35.2)
QMSum	Train	1,095	52.6	1,137.4	71.2	2,973	(38.3)	624	(37.2)	2,197	(31.7)	2,617	(29.9)	2,539	(29.5)	0	(-)
	Dev.	237	57.7	1,145.4	71.4	474	(36.0)	474	(33.9)	474	(33.0)	474	(28.1)	474	(29.5)	474	(24.9)
	Test	244	55.6	1,152.2	63.9	488	(37.4)	488	(35.0)	488	(33.8)	488	(29.4)	488	(29.1)	488	(24.6)
TweetSumm	Train	879	10.5	244.0	48.2	678	(47.3)	649	(47.2)	919	(43.2)	3,901	(30.7)	2,643	(34.9)	0	(-)
	Dev.	110	10.2	226.1	48.4	220	(52.6)	220	(50.0)	220	(48.7)	220	(34.8)	220	(35.4)	220	(49.0)
	Test	110	10.6	258.2	47.8	220	(48.4)	220	(46.9)	220	(44.4)	220	(32.6)	220	(36.1)	220	(45.4)

Table 2: Statistics of the OLDS dataset. OLDS is built upon five dialogue summarization benchmarks that cover different domains. **Len.** stands for the average length (number of words). L, B, S in the subscript of model names stand for *large*, *base*, and *small* model sizes.

responding omission labels at the utterance level. This dataset first collects multiple public benchmarks, including SAMSum (Gliwa et al., 2019), DialogSum (Chen et al., 2021), EmailSum (Zhang et al., 2021), QMSum (Zhong et al., 2021) and TweetSumm (Feigenblat et al., 2021), to cover different dialogue domains.

Then, in order to collect samples with omission contents, we still need to generate candidate summaries for each dialogue. To gain deeper insights into the omission problem induced by models with different capacities, we select 6 different model settings [2], including $\text{BART}_{\text{large/base}}$ (Lewis et al., 2020), $\text{T5}_{\text{base/small}}$ (Raffel et al., 2020), Transformers (Vaswani et al., 2017), and $\text{Pegasus}_{\text{large}}$ (Zhang et al., 2020), to generate candidate summaries [3].

Finally, based on the collected candidate summaries, we need to identify which salient information is omitted in these candidates. Therefore, we elaborately design a strategy to label omission automatically and the details are described in the next subsection. As a result, our OLDS is able to obtain multiple candidates and their corresponding omission label for each dialogue. More details about the dataset creation can refer to Appendix A.

2.3 The Automatic Labeling Strategy

It is generally a non-trivial task to identify the missing critical content in candidate summary. Fortunately, the existing datasets provide reference summaries as ground truths. We could locate the omitted information in dialogue by directly comparing candidates with references. Thus, we design a pipeline strategy for automatic omission labeling, which is composed of three steps: oracle extraction, omission identification, and redundancy removal. Appendix A.1 shows an example of the complete process of automatic omission labeling.

Oracle Extraction The first step is to match summaries to the corresponding utterances in the dialogue. Following Nallapati et al. (2017), we use a greedy algorithm to select utterances from the dialogue that maximizes the Rouge score (Lin, 2004) with respect to the summary. We return this subset of utterances as oracle labels, representing their membership in the summary. We define the extracted oracle labels for reference summaries and candidate summaries as *Gold Oracle* and *Candidate Oracle*, denoted as G and C respectively.

Omission Identification The goal of this step is to find out the omission set O. An intuitive solution is to calculate the complement of candidate oracle in gold oracle as $G - C = \{u | u \in G, u \notin C\}$. Nevertheless, it is an imperfect solution because the utterances in C might still contain omitted words or phrases. For instance, in Table 1, the 15th utterance with a phrase *"I have a friend who's a psychologist"* matches the key information "*friend*" in both reference and candidate, and this utterance would be included in both G and C. However, the keyword "*psychologist*" is actually omitted in the candidate, so the 15th utterance should be labeled

[2] We do not use extractive models because dialogue summaries are very abstractive. There is a huge gap in the format and style between summary sentences and dialogue utterances.

[3] $\text{Pegasus}_{\text{large}}$ is only used to generate summaries for dialogues in the validation/test sets. The purpose is to conduct the robustness evaluation on candidates from unseen sources.

Domain	Avg. Accept Num. (Rate)	*kappa*
SAMSum	$182.3_{\pm 3.6}$ (91.2%)	0.689
DialogSum	$188.0_{\pm 4.4}$ (94.0%)	0.616
EmailSum	$192.3_{\pm 2.5}$ (96.2%)	0.633
QMSum	$197.0_{\pm 1.0}$ (98.5%)	0.549
TweetSumm	$194.0_{\pm 1.7}$ (97.0%)	0.656
Overall	$953.7_{\pm 4.2}$ (95.4%)	0.653

Table 3: Quality assessment based on human evaluation. We randomly sampled 200 examples for each domain and asked 3 annotators to rate *Accept* or *Reject*.

as an omission. In other words, some utterances in the intersection of G and C may also be omissions. To further discover the potential omission utterances from $G \cap C = \{u | u \in G, u \in C\}$, we empirically adopt a word-level comparison approach. Specifically, for each utterance u in $G \cap C$, we further extract the overlapping words W_G^u / W_C^u [4] between u and reference/candidate summary. If $W_G^u \not\subseteq W_C^u$, we deem this corresponding utterance includes some key messages that are omitted in the candidate, and thus it should be labeled as an omission. During this process, we could obtain the omission words of utterance u, which is denoted as $W^u = \{w | w \in W_G^u, w \notin W_C^u\}$.

Redundancy Removal After the omission identification, we can obtain the omission set O. However, some utterances in O can be redundant since they could share the identical missing content. For example, for utterance u_1 and u_2, their omission words W^{u_1} and W^{u_2} can be equal so that we can argue these two utterances share similar omission information. To reduce this redundancy, we only keep the utterance with the front position if multiple utterances have the same omission words.

2.4 Quality Assessment

To assess the quality of the extracted omission labels for the OLDS dataset, we also conducted human evaluation to validate the correctness of the labeled utterances. We recruited three annotators with NLP backgrounds and each annotator is required to answer the question whether the set of labeled omission utterances is *Accept* or *Reject*. The set should be marked as *Reject* as long as it misses any critical utterance, or includes any redundant or uninformative utterance. Otherwise, it should be marked as *Accept*. To this end, we randomly sampled 200 dialogue-candidate pairs from each domain for assessment. Table 3 reports the results of the human evaluation for quality assessment. The acceptance rate of human evaluation ranges between 91.2%-98.5%, which validates the effectiveness of our omission extraction strategy. Furthermore, in order to evaluate the reliability of this assessment, we measure the agreement between different annotators by reporting Fleiss' Kappa values (Fleiss et al., 1971) among the possible combinations of two annotators, as reported in Table 3. We find that the overall Kappa score is 0.653, which shows the substantial agreement between annotators. Overall, the results of human evaluation demonstrate that our omission extraction strategy is able to produce high-quality omission labels automatically. More details about human evaluation can refer to Appendix A.4.

2.5 Dataset Format and Statistics

An example of our OLDS dataset is shown in Table 1, which contains the basic information, such as dialogue, reference, candidate, and omission labels. In the released version of OLDS, we further provide some auxiliary information. The detailed dataset format and a complete example can be seen in Appendix A.5. Table 2 shows the statistics of the OLDS dataset. We can see that the dialogues are from different domains, with different lengths and turns. Besides, the lengths of summaries also differ from each other, and the employed abstractive models are able to produce candidates with different qualities. We expect that our dataset could pave the way for analyzing the omission problem across different domains and diverse candidate summaries.

3 Understanding the Omission Problem

In this section, we explore the omission problem in different aspects and analyze why we should pay attention to omission in dialogue summarization.

3.1 Distribution of Omission Information

To explain the importance of the omission problem, we answer the following two questions.

Q1: How serious is the omission problem? For each abstractive model used in OLDS, we calculate the percentage of candidates which include omission information (i.e., the omission set $O \neq \emptyset$). Generally, a lower percentage means the model's ability to identify the salient information in dialogue is more powerful. Figure 1 shows the sta-

[4] We process words in a case-insensitive setting. We keep the original form of words but perform word stemming for comparison. Besides, stop words are removed.

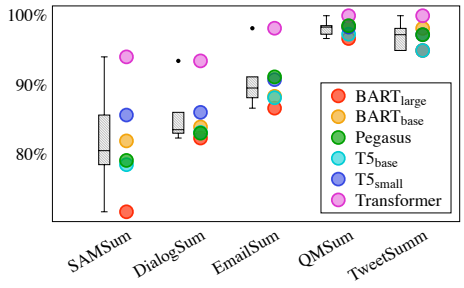

Figure 1: The percentage of candidate summaries with omission errors. We report the results of six adopted models on the test set of each dialogue domain.

Domains	SAM.	Dial.	Email.	QM.	Tweet.
RG-1	-0.563	-0.409	-0.445	-0.470	**-0.574**
RG-2	-0.448	-0.342	-0.394	-0.480	-0.524
RG-L	-0.510	-0.398	**-0.457**	-0.494	-0.547
BLEU	-0.332	-0.289	-0.231	-0.397	-0.467
BS$_{\mathrm{B}}$	-0.549	-0.502	-0.418	-0.463	-0.485
BS$_{\mathrm{L}}$	-0.562	**-0.504**	-0.445	**-0.521**	-0.546
BLEURT	**-0.567**	-0.461	-0.292	-0.410	-0.525

Table 4: Pearson correlations between *Omission Rate* and other reference-based metrics on the test set of five domains. **RG** denotes ROUGE. **BS$_{\mathrm{B}}$**, **BS$_{\mathrm{L}}$** stand for BERTScore using *Roberta-base* and *Roberta-large* as backbone models. For BLEURT, we use *BLEURT-20*.

tistical results of each model on different dialogue domains. We find that using pre-trained models always produces a lower ratio than the vanilla Transformer. Nevertheless, even using pre-trained models, we find it still reaches a high omission ratio of at least 70%. The omission phenomenon is worse in QMSum and TweetSumm, that almost 90% of their candidates have omission errors. From this perspective, we can conclude that omission is a general and grievous problem in dialogue summarization, and how to alleviate the omission problem is still intractable.

Q2: How is the omission information distributed in the dialogue? To answer this question, we investigate the position distribution of omissions in dialogues. Just as shown in Figure 2, we observe that the omitted utterances are randomly distributed in each position of the dialogue, regardless of its length and domain. This position distribution also indicates that dialogues are unstructured, and how to identify the dispersed key information precisely is still difficult for current models.

3.2 Correlation with Reference-Based Metrics

Since omission is defined by the difference between references and candidates, we thus investigate the correlation between the amount of omission content and a variety of reference-based metrics, to verify whether the omission rate of a candidate summary could affect these metrics. Here, we calculate the *omission rate* as follows:

$$\text{OmissionRate} = \frac{\sum_{u \in O} |W^u|}{\sum_{u \in G} |W^u_G|}, \qquad (1)$$

where W^u and W^u_G denote the set of omitted words and the set of gold oracle words shared across u and the reference, respectively. It directly measures the amount of key information omitted by a summary, and a lower rate indicates the candidate is of higher quality. Table 4 demonstrates the Pearson correlations between omission rate and other reference-based metrics, including n-gram based metrics ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002), embedding-based metric BERTScore (Zhang et al., 2019), and learning-based metric BLEURT (Sellam et al., 2020). The results indicate that most of the reference-based metrics moderately correlate with the omission rate, among which BERTScore$_{\mathrm{Large}}$ is the most stable metric that has a better correlation with the amount of omission content. By contrast, BLEU shows the least correlation because it is a precision-oriented metric. Empirical analyses indicate that the omission rate is strongly correlated with a wide range of evaluation metrics, and so how to mitigate the omission problem is one of the most important priorities to improve the quality of dialogue summaries.

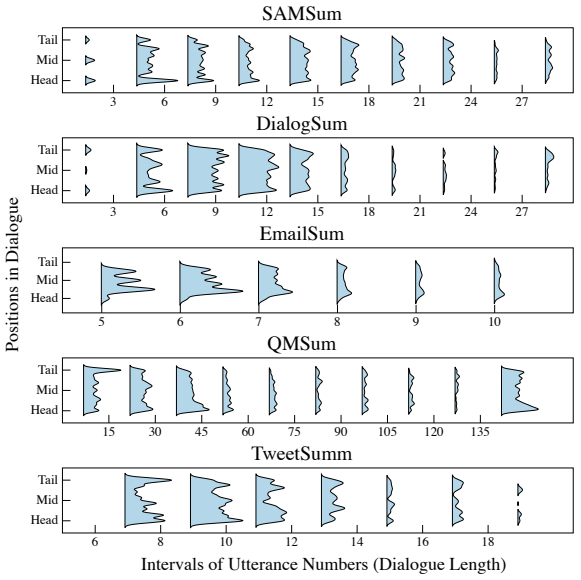

Figure 2: Position distribution of omissions in dialogues across different domains. The X-axis represents the intervals of untterance numbers.

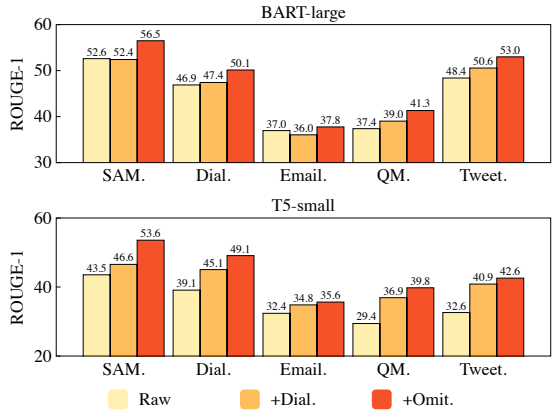

Figure 3: Post-editing results in different domains. *Raw* means the results of raw candidates. *+Dial.* and *+Omit.* mean using raw dialogue or omissions as the supplement information for refinement.

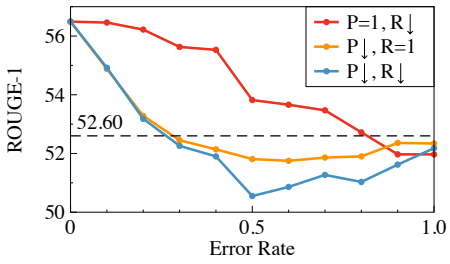

Figure 4: Change of the post-editing results by perturbing input omissions. The results are from the SAMSum test set using $\text{BERT}_{\text{large}}$. The dotted line shows the raw results before post-editing. P and R denote the precision and recall of omissions. ↓ stands for a decreasing trend.

3.3 Omission-based Summary Refinement

The above analyses demonstrate the importance of omission information. So we raise another question: what happens if we utilize the omissions to refine the summary quality? Hence, we adopt a post-editing method to investigate the potential of using omissions. Specifically, we formulate summary refinement as a seq2seq task to predict the gold summary. Instead of inputting raw dialogue, we use the concatenation of candidate summary, omission utterances, and non-omission utterances as the input: "*Candidate* <sep> *Omission* <sep> *Non-Omission*". By dividing dialogue utterances into the omission and non-omission groups, the model is able to distinguish omission information while perceiving the whole dialogue simultaneously. If the omission group is empty, it is identical to using candidate and raw dialogue for refinement, and we consider it as the baseline for comparison. We use $\text{BART}_{\text{large}}$ and T5_{small} as the backbone model, and the results are shown in Figure 3. The results show that performances are significantly enhanced by the refinement using omissions compared to that using raw dialogues, which indicates that omission-based refinement is a promising direction for quality improvement in dialogue summarization.

In addition, Figure 3 also shows an upper bound of performance boost by post-editing because we directly employ the gold omission utterances. However, in real situations, we may identify some incorrect omissions. To further explore the impact of wrong omissions on the post-editing results, we investigate three different perturbations by gradually injecting errors into the omission group: 1) we keep the precision as 1 and decrease the recall by moving utterances from the omission group to the non-omission group; 2) we keep the recall as 1 and decrease the precision by moving utterances from the non-omission group to the omission group; 3) we gradually exchange utterances in the two groups until they are swapped, and both the precision and recall decrease from 1 to 0. Figure 4 depicts the trend of performance degradation as the error rate increases. From the curves, we can find that the precision is relatively more important because the refinement model performs more robustly in the first type of perturbation and is sensitive to the addition of wrong omissions.

4 The Omission Detection Task

Since candidate summaries could be effectively improved given the gold omission information, how to accurately detect omission utterances in dialogue naturally becomes a critical question. In this section, we formulate the omission detection task in a reference-agnostic setting. Formally, given a dialogue $D = \{u_1, u_2, .., u_N\}$ along with a candidate summary c, a detection model is required to extract a set of omission utterances O from D without knowing the reference summary. In this section, we introduce three typical frameworks as baselines and conduct evaluations to see how this task could benefit from them.

4.1 Model Settings

To build a foundation for the omission detection task and explore what model architecture the task could benefit from, we investigate three frameworks as baselines, which have different input formats and structures. Their implementation and training details can be found in Appendix B.1.

Model	SAMSum				DialogSum				EmailSum				QMSum				TweetSumm			
	P	R	F1	WR	P	R	F1	WR	P	R	F1	WR	P	R	F1	WR	P	R	F1	WR
Pair-wise Classification																				
BERT	41.66	38.60	40.07	54.83	38.01	45.56	41.44	57.23	47.94	40.81	44.09	50.93	35.97	42.86	39.12	60.73	41.84	47.17	44.35	53.86
RoBERTa	42.45	43.27	42.85	58.94	38.42	44.93	41.43	57.56	48.13	50.02	49.05	59.04	32.92	**43.65**	37.53	**60.99**	41.37	49.66	45.14	57.08
Sequence Labeling																				
BERT	45.18	43.57	44.37	61.35	40.71	46.23	43.30	57.51	50.58	**50.41**	50.49	**61.11**	47.11	31.22	37.56	47.29	40.70	49.72	44.76	58.48
RoBERTa	47.34	**47.65**	**47.49**	**63.97**	42.63	**46.54**	44.50	58.51	**53.62**	48.65	**51.01**	59.04	**48.09**	36.27	**41.35**	52.82	48.26	48.85	48.55	59.27
Pointer Network																				
BERT	47.20	39.13	42.79	58.52	41.23	42.79	42.00	56.02	52.57	48.47	50.44	60.66	45.31	31.73	37.32	48.46	**48.69**	42.35	45.30	52.10
RoBERTa	**50.64**	41.90	45.86	60.25	**43.68**	45.90	**44.76**	**60.03**	53.61	46.04	49.54	56.92	44.80	35.16	39.40	53.21	47.23	**52.18**	**49.58**	**63.61**

Table 5: Experimental results of omission detection on OLDS dataset. WR means the word-level omission recall.

Pair-wise Classification A straightforward way is to model this task as an utterance-level classification problem. The input pattern for this paradigm is: <*s*> c </*s*> u_i </*s*>, where <*s*> and </*s*> denote the classification token and separation token, respectively. c is the candidate summary and u_i is the i-th utterance in the dialogue. The model would perform binary classification for the candidate-utterance pair as $y \in \{0, 1\}$, where $y = 1$ represents that the utterance is identified as an omission.

Sequence Labeling Inspired by BERTSum (Liu and Lapata, 2019) that formulates extractive summarization as a sequence labeling problem at the sentence level, we employ a similar strategy which assigns each utterance a label $y_i \in \{0, 1\}$ indicating whether the utterance is an omission. We append the candidate summary in front of the dialogue, as <*s*> c </*s*> <*s*> u_1 </*s*> <*s*> u_2 </*s*> ... <*s*> u_N </*s*>. The last hidden layer of each <*s*> token will be used as utterance representations for classification.

Pointer Network Pointer network is to select the omission utterance recurrently using glimpse operation (Vinyals et al., 2015) based on previous predictions. It is a widely-used strategy for sentence extraction in summarization (Chen and Bansal, 2018; Zhong et al., 2019; Zou et al., 2021b). Here, we use the same input format as in sequence labeling, and the pointer network outputs an extraction distribution based on the <*s*> representations.

4.2 Evaluation Metrics

We use the standard Precision (P), Recall (R), and F1-score (F1) metrics on the utterance level to evaluate omission detection models. Furthermore, we calculate the percentage of gold omission words that are hit in the detected utterances to measure the word-level omission recall:

$$WR = \frac{\#\text{hit omission words}}{\#\text{gold omission words}}. \quad (2)$$

$\#$ means the counted number. The closer the word-level omission recall is to 1, the more the omission information is collected by the detection model.

4.3 Main Results

Table 5 presents the experimental results on OLDS. All detection models are separately trained on the five domains. For each omission detection framework, we employ $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ as the backbone model to extract text features. Among these three frameworks, pair-wise classification performs the worst in most cases since it does not consider contextual information of dialogue. Meanwhile, sequence labeling is on par with the pointer network, which indicates that dialogue context is a crucial factor for models to detect the omitted content. However, although omission detection models only need to make a choice of whether the given utterance is an omission, the task is still very challenging. In Table 5, the best F1 score is around 50% in all five domains, while the recalled omission words in extracted utterances (WR) are around 60%. Besides, models in QMSum only achieve at most a F1-score of 41.35 and we guess it is due to the effect of longer dialogue in QMSum (over 1K tokens in Table 2). Intuitively, summarizers produce the candidates that have picked the low-hanging fruit, and the remaining omission information is a tough nut to crack. In other words, there exists some salient information omitted by the summarizer that is still difficult for detection models to capture.

4.4 Analysis and Discussion

To understand what factors may affect the performance of the detection model, we conduct the following explanatory experiments.

Label Imbalance We first calculate the percentage of omission utterances against non-omission utterances in five domains to investigate whether

Framework	Model	Overall		$\text{BART}_{\text{Large}}$		$\text{BART}_{\text{Base}}$		T5_{Base}		T5_{Small}		Transformer		$\text{Pegasus}_{\text{Large}}$	
		F1	WR	F1	WR	F1	WR	F1	WR	F1	WR	F1	WR	F1	WR
Pair-wise classification	BERT	40.07	54.83	31.98	54.77	38.40	55.14	33.80	51.75	42.83	55.51	48.60	56.69	38.25	55.11
	RoBERTa	42.85	58.94	35.52	58.32	41.28	57.86	38.75	58.03	44.68	59.80	51.43	61.29	39.39	58.36
Sequence Labeling	BERT	44.37	61.35	35.58	59.94	42.23	60.23	40.27	60.63	46.70	62.77	53.82	64.85	41.07	59.60
	RoBERTa	**47.49**	**63.97**	**38.37**	**61.23**	**45.66**	**62.51**	**43.50**	**62.42**	**50.41**	**66.81**	**55.94**	**67.43**	**44.67**	**63.32**
Pointer Network	BERT	42.79	58.52	35.75	58.53	39.82	57.65	38.86	57.83	44.91	59.32	52.04	60.55	39.57	57.22
	RoBERTa	45.86	60.25	37.12	58.27	43.54	58.81	40.88	58.01	48.06	62.50	54.78	63.90	44.03	60.03

Table 6: Omission detection results on the candidate summaries from SAMSum test set, which are categorized into multiple groups according to their source summarizer.

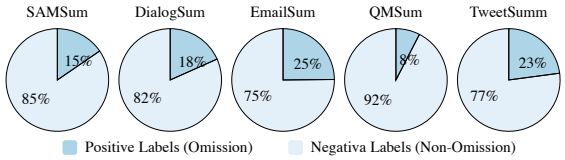

Figure 5: The proportion of positive labels (omission utterances) against negative ones (non-omission utterances) in five domains.

the label imbalance problem exists in the datasets. Figure 5 shows that the proportion of positive labels is always smaller than 25%, which indicates that label imbalance is a common problem in omission datasets. Besides, we observe that the degree of label imbalance is consistent with the performance of detection models, according to the results in Table 5. For example, the models achieve nearly 50% F1-score in EmailSum and TweetSumm, which have a ratio of 25% and 23% omission utterances. However, in QMSum, the detection models only achieve a 40% F1-score as the omission proportion of this dataset is only 8%. Hence, how to alleviate label imbalance is critical for omission detection and we leave it as future work.

Candidate Quality Furthermore, we evaluate the performance of detection models on the candidates produced by different abstractive summarizers to investigate whether the candidate quality may influence detection models. The results are shown in Table 6, and we find the result of omission detection is negatively correlated with the performance of summarizers. For instance, BART_{L} and $\text{Pegasus}_{\text{L}}$ produce candidates with higher quality, yet the detection model has difficulty obtaining their omissions. On the contrary, Transformer produces relatively low-quality candidates, while the detection model could produce better results (i.e., 55.94% F1-score). It indicates that capturing the remaining omissions for high-quality candidates is difficult, and how to address this issue is also valuable.

4.5 Future Research Opportunities

From the results in Table 5, we could observe that omission detection is a challenging task. Hence, we summarize some research directions as follows:

- One direction is to develop a more advanced model for omission detection. Based on the analysis of Section 3.3, we could focus on improving the precision of omission detection results because a high precision of detected omissions would bring benefit to the refinement model. An ideal detection model could serve as a model-based metric for reference-free summary evaluation. Besides, we could use the detected omission to improve the results of summarization.
- Another research direction is to develop a refinement model for summary improvement using the detected omissions. In this paper, we briefly touch on this by introducing a post-editing approach in Section 3.3. The approach is straightforward, and the whole summarization procedure becomes a summarize-then-refine pipeline. However, the results show that the model is sensitive to wrong omissions. Hence, how to design a robust refinement model is also noteworthy.

5 Conclusion

In this work, we systematically study the omission problem in dialogue summarization based on the curated OLDS dataset, which collects candidate summaries from multiple models and domains and provides high-quality omission labels for them. We discover that omission is a significant problem that directly affects the results of dialogue summarization, and the defective candidate summary could be largely improved by leveraging the omission information properly. We further introduce an omission detection task to identify omission content, which is a challenging and valuable task that paves the way to omission mitigation and summary improvement in dialogue summarization.

6 Limitations

The omission problem is critical in dialogue summarization, but even if this problem is solved, we still cannot guarantee a candidate is appropriate because it might bring hallucination content that is not presented by the source dialogue. Previous works (Tang et al., 2022; Maynez et al., 2020) also concluded that factual inconsistency is a critical problem in dialogue summarization, and it is not easy to distinguish. How to mitigate the omission problem while avoiding the occurrence of new errors is not discussed in this paper, and we hope to address this issue in future work.

References

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.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074.

Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, and Ranit Aharonov. 2021. Tweetsumm-a dialog summarization dataset for customer service. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 245–260.

Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2022. A survey on dialogue summarization: Recent advances and new frontiers. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5453–5460. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

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.

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.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740.

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.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021. Coreference-aware dialogue summarization. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 509–519.

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.

Gabriel Murray and Giuseppe Carenini. 2008. Summarizing spoken and written conversations. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 773–782.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 105–108.

Graham Russell. 1999. Errors of omission in translation. In *Proceedings of the 8th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, University College, Chester.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Vipin Sharma. 2015. The relevance of addition, omission and deletion (aod) in translation. *International Journal of Translation (IJT) (ISSN: 0940-9819)*, 27:1–13.

Yan Song, Yuanhe Tian, Nan Wang, and Fei Xia. 2020. Summarizing medical conversations via identifying important utterances. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 717–729.

Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. 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.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*.

Yue Weng, Sai Sumanth Miryala, Chandra Khatri, Runze Wang, Huaixiu Zheng, Piero Molino, Mahdi Namazifar, Alexandros Papangelis, Hugh Williams, Franziska Bell, and Gökhan Tür. 2020. Joint contextual modeling for ASR correction and language understanding. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 6349–6353. IEEE.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy.

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.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. Emailsum: Abstractive email thread 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 6895–6909.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 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.

Yicheng Zou, Jun Lin, Lujun Zhao, Yangyang Kang, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021a. 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.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021b. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic

modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14665–14673.

Yicheng Zou, Bolin Zhu, Xingwu Hu, Tao Gui, and Qi Zhang. 2021c. Low-resource dialogue summarization with domain-agnostic multi-source pretraining. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 80–91.

A Details of the OLDS dataset

A.1 Example of Automatic Omission Labeling

Figure 6 shows an example of the complete process of automatic omission labeling, which consists of three steps: oracle extraction, omission identification, and redundancy removal.

For oracle extraction, we select utterances greedily from the dialogue to maximize the Rouge score with respect to the summary. We obtain this subset of utterances as oracle labels, representing their membership in the summary. In this example, we generate oracle labels for the reference as *Gold Oracles*, i.e., an utterance set of {0, 2, 5, 6, 9, 12, 13, 14, 16, 19}, and oracle labels for the candidate as *Candidate Oracles*, i.e., {0, 7, 12, 14, 16, 19}.

In the process of omission identification, we traverse the utterances in Gold Oracles and extract W_G^u, which is a set of words containing the overlapping words between u and the reference. For instance, in the 14th utterance, "soon, Hector, Ashley" are the keywords appearing in the reference. Similarly, we extract W_C^u that contains the overlapping words between u and the candidate summary, where $u \in$ Gold Oracles. Then, by comparing W_G^u and W_C^u, we could obtain the omission words $W^u = \{w | w \in W_G^u, w \notin W_C^u\}$. For any utterance u where $W^u \neq \emptyset$, we label it as an omission utterance. In the example of Figure 6, the 14th utterance contains the keywords "soon, Ashley" which are omitted by the candidate, and it should be labeled as an omission.

Finally, we conduct redundancy removal to discard redundant omission utterances. In Figure 6, the 2nd, 5th, and 19th utterances have redundant omission words W^u, which are the same as those in other omission utterances. Hence, we remove these utterances and the final omission labels are {0, 9, 14, 16}.

A.2 Dialogue Domains

We build the OLDS dataset upon five existing dialogue summarization datasets that cover different domains, which are described as follows:

SAMSum It is the first high-quality online chat summarization corpus (Gliwa et al., 2019), which contains about 16k simulated conversations created by linguists with corresponding summaries.

DialogSum It is a summarization dataset (Chen et al., 2021) with 13.5k real-life scenario dialogues, which are face-to-face spoken dialogues that cover a wide range of daily-life topics.

EmailSum It is an email thread summarization dataset (Zhang et al., 2021) that consists of 2,549 email threads along with annotated summaries. The dataset has two types of summaries, short summary (<30 words) and long summary (<100 words). Here, we use the short version as references because they are more abstractive and challenging.

QMSum It is a query-based multi-domain meeting summarization benchmark (Zhong et al., 2021) that contains 1,808 query-summary pairs over 232 meetings. We concatenate queries with their corresponding text spans as the input dialogues [5].

TweetSumm It is a dataset focused on customer service conversations (Feigenblat et al., 2021), which contains 1,100 dialogues, each accompanied by 3 extractive and 3 abstractive summaries. We use the longest abstractive summary as the gold reference.

A.3 Candidate Generation

We use 6 abstractive models to generate candidates for the dialogues in OLDS, including $\text{BART}_{\text{large/base}}$, $\text{T5}_{\text{base/small}}$, vanilla Transformer, and $\text{Pegasus}_{\text{large}}$. $\text{Pegasus}_{\text{large}}$ is only used to generate candidates for dialogues in evaluation sets.

To obtain the candidate summaries in training sets, we train the summarization models by adopting a 10-fold cross-validation approach, and each model generates 10 candidates for each dialogue in the validation fold via different configurations of beam search and sampling. As a result, we can obtain 50 candidates (5 models $\times$ 10 inferences) for each dialogue in the training set. To ensure the diversity of the generated candidates, we further calculate the average Levenshtein distance (Levenshtein, 1965) for each candidate and pick out 10

[5] We removed 232 query-summary pairs which summarize the whole meeting transcripts because their input lengths are significantly different from other pairs. As a result, the final number of pairs used in our dataset is 1,576.

candidates with the largest scores. Specifically, we combine these candidates in pairs (a total of 50 $\times$ 50 = 2,500 pairs) and calculate the Levenshtein distance between them. Then, for each candidate, we average the distance results against the other 49 candidates to obtain the average Levenshtein distance. Finally, we rank these candidates based on the scores in descending order and pick out the top 10 candidates. As a result, we have 10 diverse candidates for each dialogue in the training sets.

For the evaluation set of OLDS, we train the aforementioned 6 models on the training set of each domain to produce candidate summaries. Each summarization model produces 2 candidates, which are decoded by beam search (beam size = 5) and sampling, respectively. Hence, we totally have 12 candidates for each dialogue in evaluation sets.

The training and inference process was conducted based on the official code of pre-trained language models [6]. All experiments were conducted on one node with 4 32GB V100 GPUs. The learning rate is set to 5e-5 for pre-trained models and is set to 1e-4 for Transformer. The pre-trained models are fine-tuned with 3 epochs, while the vanilla Transformer is trained with 20 epochs. For SAMSum, the maximum source length and target length is 512 and 90, and for DialogSum, EmailSum, QMSum, and TweetSumm, this setting is 512/150, 1,024/65, 2,048/200, and 1,024/120, respectively. The other hyper-parameters are set by default.

A.4 Details of Quality Assessment

Time Budget We recruited three annotators to conduct the quality assessment for OLDS. The total hits of judgment are 3000 (5 domains $\times$ 200 samples $\times$ 3 annotators). The annotating speed is 25 samples per hour and the workload is 120 hours (1000 / 25 * 3 = 120) in total.

Instructions Each annotator was presented with a sample containing the dialogue, reference summary, candidate summary, gold oracles, candidate oracles, and the labeled omission utterances along with their corresponding omitted words. We instruct the annotators to make a binary choice whether the set of labeled omission utterances is *Accept* or *Reject*. Annotators should compare the candidate with the reference and find out omissions. Then, they should locate omissions in the original dialogue and record the corresponding utterances. Finally, they should compare the automatically labeled utterances with the recorded ones and make a judgment. The set of labeled omission utterances should be marked as *Reject* as long as it misses any critical utterance, or includes any redundant or uninformative utterance. Otherwise, it should be marked as *Accept*. To ensure that each choice is justified, we additionally asked annotators to perform corrections and renew the corresponding omitted words if the choice is *Reject*. Thus, we could verify why the labeled omission is marked as *Reject*.

A.5 Data Format

To facilitate the community to explore the effect of possible elements on the omission problem, in the released version of OLDS, we additionally provide some auxiliary information. Specifically, apart from the basic information of dialogue, reference summary, candidate summary, and omission labels, we further provide the intermediate information during labeling, including Gold Oracles, Candidate Oracles, omission words, and the source model and decoding strategy for each candidate summary, e.g., *'bart_base, beam'*, which represents that the candidate is generated by BART$_{base}$ using beam search. A complete example is shown in Table 8.

B Omission Detection Models

B.1 Implementation Details

We use BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ as the backbone pre-trained encoder for the three frameworks. All the experiments were conducted on one node with a single A100 80GB GPU. For all three frameworks, the learning rate is set to 5e-5 and the training epoch is set to 5. The batch size was set to 128 for pair-wise classification and was set to 16 for sequence labeling and pointer network. We saved checkpoints after each epoch. The best performing checkpoint on the validation set was evaluated on the test set to report the final results.

Pair-wise Classification For the framework of pair-wise classification, we use the official code of classification with pre-trained language models [7]. The input format is <*s*> c </*s*> u_i </*s*>, where <*s*> and </*s*> are classification token and separation token, respectively. c and u_i represent the candidate and the i-th utterance in the dialogue.

[6] https://huggingface.co/docs/transformers/tasks/summarization

[7] https://huggingface.co/docs/transformers/tasks/sequence_classification

Sequence Labeling We use the same implementation as the extractive summarization model proposed by Liu and Lapata (2019). The only difference is that we append the candidate summary in front of the dialogue, denoted as <s> c </s> <s> u_1 </s> <s> u_2 </s> ... <s> u_N </s>. The <s> token before the candidate summary is not involved in the calculation. For SAMSum, we truncate each input into a maximum length of 512, while for DialogSum, EmailSum, QMSum, and TweetSumm, this setting is 512, 1,024, 2,048, and 1,024.

Pointer Network The autoregressive decoder of our pointer network is implemented by a Transformer decoder, which is proposed by Zou et al. (2021b) and was previously used for extractive summarization. Here, we also append the candidate summary in front of the dialogue, which has the same input format as in sequence labeling. The <s> token before the candidate summary is not involved in the calculation. We also set the same maximum length as in sequence labeling for input sequences in different domains.

B.2 Cross-Domain Results

In addition, we conduct the cross-domain evaluation to investigate domain gaps and the generalizability of detection models. From Table 7, we can conclude that there are obvious differences between these five domains. For example, the models trained on the other domains perform poorly when tested directly on QMSum. Among these five domains, the difference between SAMSum and DialogSum is relatively small due to their similar performances across domains. We also find that the model trained on SAMSum has a better capability of generalizing to other domains, even achieving the best result on the DialogSum and EmailSum domains. A possible explanation is that the SAMSum domain has more training samples, leading to better robustness.

Domains	SAM.	Dial.	Email.	QM.	Tweet.
SAM.	**63.97**	**59.39**	**66.80**	36.68	53.92
Dial.	49.65	58.51	66.58	43.50	53.77
Email.	37.78	30.69	59.04	20.98	24.13
QM.	41.39	47.00	61.15	**52.82**	28.20
Tweet.	44.92	48.46	57.07	14.74	**59.27**

Table 7: Cross-domain evaluation results. Each row represents the training set, and each column represents the test set. We use the sequence labeling framework equipped with $\text{RoBERTa}_{\text{base}}$ for these experiments and use the word-level omission recall (WR) for evaluation.

C Related Work

C.1 Dialogue Summarization

Dialogue summarization is a challenging and valuable task that has recently received much attention, where a variety of dialogue domains are investigated, such as mail threads (Rambow et al., 2004; Zhang et al., 2021), meetings (Chen and Yang, 2020; Zhong et al., 2021), customer service (Zou et al., 2021a,b; Feigenblat et al., 2021), medical conversations (Joshi et al., 2020; Song et al., 2020), and daily chats (Gliwa et al., 2019; Chen et al., 2021). Different from conventional documents, dialogues have several inherent characteristics that make the summarization task more challenging (Zou et al., 2021c; Feng et al., 2022), e.g., multi-party information, coreferences, topic drifting, etc. Recent works have explored the types of errors in generated dialogue summaries to develop robust models (Tang et al., 2022), and omission is assessed as the most dominant error type in candidate summaries, which is also supported by human evaluations in previous works (Chen and Yang, 2020; Liu and Chen, 2021; Liu et al., 2021). In this work, we comprehensively analyze the omission problem in dialogue summarization based on the curated benchmark and investigate the feasibility of omission detection for generated candidates.

C.2 Omission in Text Generation Tasks

Omission is a common error in machine translation (MT) (Russell, 1999; Sharma, 2015; Yang et al., 2019) and automatic speech recognition (ASR) tasks (Weng et al., 2020), which usually denotes the missing source information in the generated sequences. Although both summarization and MT/ASR belong to generation tasks, the definitions of omission error are different among these tasks. In MT/ASR tasks, the tokens between source and target sequences are usually well aligned, which means each token in the target sequence can locate its corresponding content in the source sequence. Due to such characteristics, previous works (Tu et al., 2016) in MT/ASR tasks usually adopted coverage mechanisms to eliminate the influence of omission error. Nevertheless, the source sequences in summarization tasks usually include abundant redundant and useless information, especially in dialogue scenarios, which makes omission a more serious problem in summarization-like tasks.

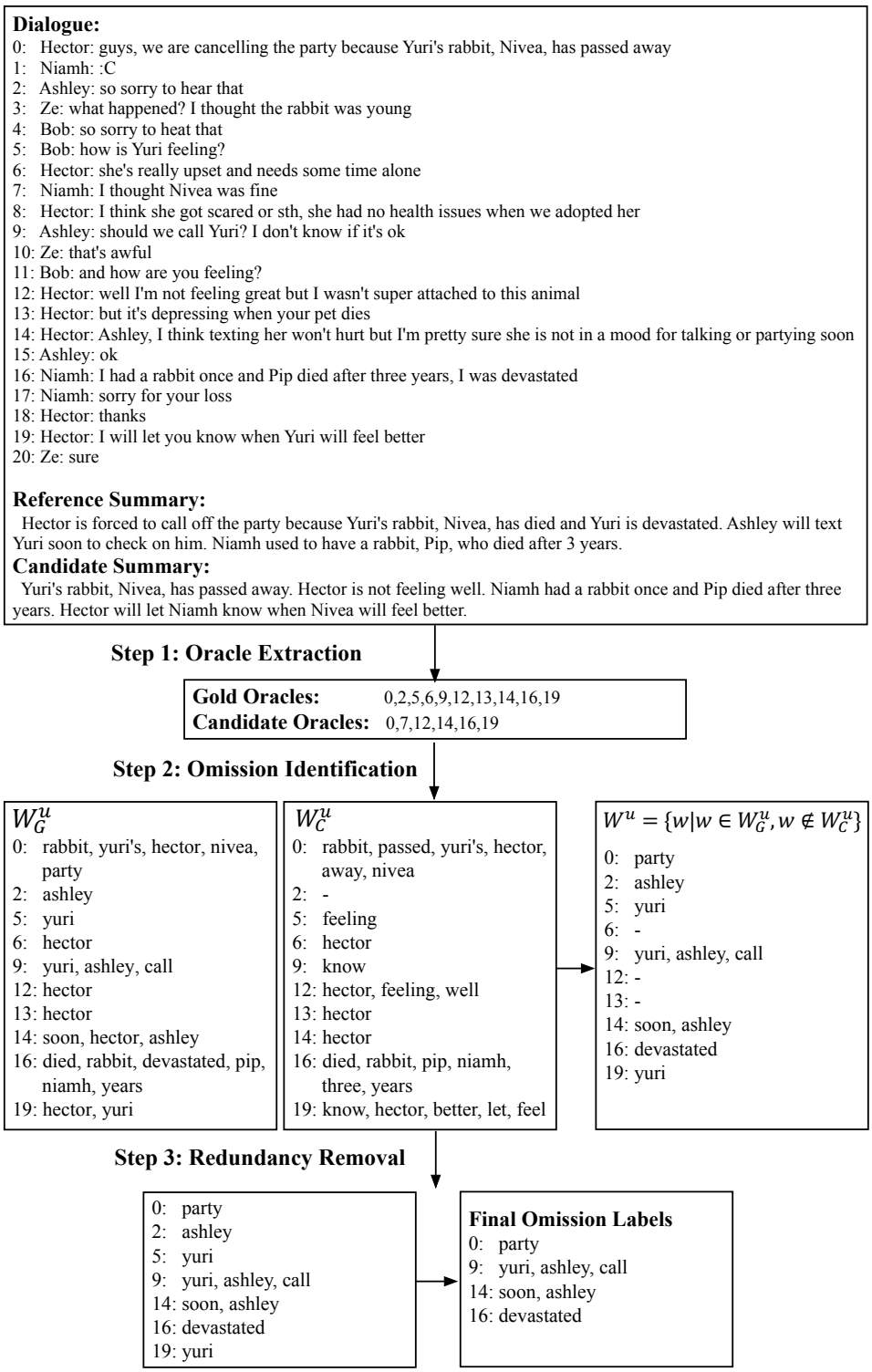

Figure 6: An example of the complete process of automatic omission labeling, which is sampled from the training set of SAMSum. W_G^u is a word set that contains all overlapping words between u and the reference summary. Similarly, W_C^u contains overlapping words between u and the candidate summary. W^u is the set of omission words.

Dialogue:

(0) @AzureSupport Hi guys we have signed up a trial for log analytics while we test setting up a custom log import. The issue I have already is trying to add a custom log import configuration, nothing is being added to the list, but no error messages? I have tried a hundred times.
(1) @242694 Could you please post here: -link- and send us the link so we can have an engineer on that team assist
(2) @AzureSupport Done.
(3) @242694 Could you please send us the link to the created post so we can alert the team? Thanks!
(4) @AzureSupport -link-
(5) @242694 Thank you! We have alerted the team and they should respond to your post shortly.
(6) @AzureSupport Thanks!
(7) @AzureSupport Hey guys no word yet, got a client waiting please.
(8) @242694 We're sorry about that. We've reached out again and will make sure that they reply to the forum post ASAP.
(9) @AzureSupport -emoji-
(10) @AzureSupport No word yet
(11) @242694 We're sorry about this. We'll reach out to the team to ask where they are on this.
(12) @242694 A forum engineer has replied to your post. Please have a look and reply to the thread if you need further assistance.

Reference summary:

The customer says that he has signed up a trail for log analytics but he is having an issue in adding a custom log import configuration. The agent asks to post the information regarding the isuue via provided link and to send them the link and says A forum engineer has replied to the post, then asks to reply to the thread for further assistance.

Candidate summaries:

[1] **Source: bart-large, Strategy: beam** Customer is complaining that he is trying to add a custom log import configuration, nothing is being added to the list, but no error messages. Agent updated that they have alerted the team and they should respond to their post shortly.
[2] **Source: bart-large, Strategy: sample** Customer is complaining that he is trying to add a custom log import configuration, nothing is being added to the list, but no error messages. Agent updated that they have alerted the team and they should respond to their post shortly.
[3] **Source: bart-base, Strategy: beam** The customer says that he has signed up a trial for log analytics while he test setting up a custom log import. The agent asks to send the link to the created post so that they can alert the team.
[4] **Source: bart-base, Strategy: sample** Customer is complaining that they have signed up a trial for log analytics while they test setting up a custom log import. Agent updated that they will reach out to the team to ask where they are on this issue.
[5] **Source: t5-base, Strategy: beam** Customer is complaining that they have signed up a trial for log analytics while they test setting up a custom log import. Agent updates that they have reached out again and will make sure that they reply to the forum post ASAP.
[6] **Source: t5-base, Strategy: sample** Customer is complaining that they have signed up a trial for log analytics while they test setting up a custom log import. Agent updates that they have reached out again and will make sure that they reply to the forum post ASAP.
[7] **Source: t5-small, Strategy: beam** AzureSupport Hi guys we have signed up a trial for log analytics while we test setting up a custom log import. The issue I have already is trying to add a custom log import configuration, nothing is being added to the list, but no error messages? I have tried a hundred times.
[8] **Source: t5-small, Strategy: sample** The issue I have already is trying to add a custom log import configuration, nothing is being added to the list, but no error messages? I have tried a hundred times. Could you please post here: -link- and send us the link to the created post so we can alert the team?
[9] **Source: transformer, Strategy: beam** The customer says that he is unable to find the product he can't work with his phone. The agent asks whether the customer is using and asks whether he will be able to send the issue and asks to assist further on the issue.
[10] **Source: transformer, Strategy: sample** Customer is complaining that he is unable to know about the delay of the product. Agent updates that they are unable to reach out for further assistance and requests to DM the issue.
[11] **Source: pegasus, Strategy: beam** The issue I have already is trying to add a custom log import configuration, nothing is being added to the list, but no error messages. The issue I have already is trying to add a custom log import configuration, nothing is being added to the list, but no error messages.
[12] **Source: pegasus, Strategy: sample** Customer is complaining that they have signed up a trial for log analytics while they are testing setting up a custom log import. Agent updates that they have alerted the team and they should respond to their post shortly and adds that they have reached out again and will make sure that they reply to the forum post ASAP.

Gold Oracles: (0) (1) (3) (8) (11) (12)

Candidate Oracles:

[1]: (0) (1) (5) (8)	[4]: (0) (1) (5) (8) (11)	[7]: (0)	[10]: (0) (3) (8) (11) (12)
[2]: (0) (1) (5) (8)	[5]: (0) (1) (5) (8) (11)	[8]: (0) (1) (3)	[11]: (0) (3) (7) (12)
[3]: (0) (1) (3) (8) (11) (12)	[6]: (0) (1) (5) (8) (11)	[9]: (0) (1) (3) (4) (8) (11) (12)	[12]: (0) (1) (5) (8) (11) (12)

Omission utterances (Labels):

[1]: (0) (1) (12)	[4]: (0) (1) (12)	[7]: (1) (12)	[10]: (0) (1) (12)
[2]: (0) (1) (12)	[5]: (0) (1) (12)	[8]: (0) (12)	[11]: (0) (1) (12)
[3]: (0) (12)	[6]: (0) (1) (12)	[9]: (0) (1) (12)	[12]: (0) (1) (12)

Omission Words:

[1]: (0) issue, analytics, signed (1) engineer, send, link (12) engineer, forum, replied, assistance, thread, reply
[2]: (0) issue, analytics, signed (1) engineer, send, link (12) engineer, forum, replied, assistance, thread, reply
[3]: (0) issue, configuration (12) engineer, forum, replied, assistance, thread, reply
[4]: (0) configuration (1) engineer, post, send, link (12) engineer, forum, replied, assistance, thread, post, reply
[5]: (0) issue, configuration (1) engineer, send, link (12) engineer, replied, assistance, thread
[6]: (0) issue, configuration (1) engineer, send, link (12) engineer, replied, assistance, thread
[7]: (1) engineer, post, send, link (12) engineer, forum, replied, assistance, thread, post, reply
[8]: (0) analytics, signed (12) engineer, forum, replied, assistance, thread, reply
[9]: (0) analytics, custom, import, signed, log, configuration (1) engineer, post, link (12) engineer, forum, replied, assistance, thread, post, reply
[10]: (0) analytics, custom, import, signed, log, configuration (1) engineer, post, send, link (12) engineer, forum, replied, thread, post, reply
[11]: (0) analytics, signed (1) engineer, post, send, link (12) engineer, forum, replied, assistance, thread, post, reply
[12]: (0) issue, configuration (1) engineer, send, link (12) engineer, replied, assistance, thread

Table 8: A complete example in the OLDS dataset, which is sampled from the test set of TweetSumm domain.