Summarizing Dialogues with Negative Cues

Junpeng Liu[†*]
Beijing University of Posts and Telecommunications, Beijing, China
jeepliu@bupt.edu.cn

Yanyan Zou[†]
JD.com, Beijing, China
zouyanyan6@jd.com

Yuxuan Xi[*]
Peking University, Beijing, China
xiyuxuan@pku.edu.cn

Shengjie Li and **Mian Ma** and **Zhuoye Ding**
JD.com, Beijing, China
{lishengjie1,mamian,dingzhuoye}@jd.com

Abstract

Abstractive dialogue summarization aims to convert a long dialogue content into its short form where the salient information is preserved, while the redundant pieces are ignored. Different from the well-structured text, such as news and scientific articles, dialogues often consist of utterances coming from two or more interlocutors, where the conversations are often informal, verbose, and repetitive, sprinkled with false-starts, backchanneling, reconfirmations, hesitations, speaker interruptions and the salient information is often scattered across the whole chat. The above properties of conversations make it difficult to directly concentrate on scattered outstanding utterances and thus present new challenges of summarizing dialogues. In this work, we propose to explicitly have the model perceive the redundant parts of an input dialogue history, so that the model is able to pay more attention to the salient pieces. To be specific, we design two strategies to construct examples without salient pieces as negative cues. Then, the sequence-to-sequence likelihood loss is cooperated with the unlikelihood objective to drive the model focus less on the unimportant information as well as pay more attention to the salient pieces. Extensive experiments on the benchmark dataset demonstrate that our simple method outperforms baselines with regard to both semantic matching and factual consistent based metrics. The human evaluation also proves the performance gains led by our approach.

1 Introduction

Online conversations have become an indispensable manner of communication in our daily work and life, where people tend to exchange their ideas, share information, consult via textual messages. Especially in the era of information explosion, it is much more challenging and time-consuming to go through all the conversation content and catch key ideas (Gao et al., 2020). Thus, it is paramount to present the most salient facts, instead of the whole lengthy dialogue history, which is beneficial to various scenarios and applications, such as online customer service (Liu et al., 2019a), meeting and email thread summary (Zhao et al., 2019). Therefore, this work focuses on the *abstractive dialogue summarization* task, aiming to automatically convert the long dialogue history into its shorter form retaining the most essential and informative content yet getting rid of the dispensable pieces, exemplified by a dialogue-summary instance in Figure 1.

Molly:	*Guys, do you think it's a very bad idea to go to Sweden for a week in January?*
Margaret:	We bought some cheep tickets half a year ago and now we're hesitating.
Peter:	Haha, no but it will be just dark and cold.
Margaret:	Rainy?
Kal:	Possibly. *But if you stay in Stockholm, there are always nice things to do. Museums, bars etc*
Kal:	Not so much nature though which is truly stunning around Stockholm.
Margaret:	Yes, but it's January, one would have to go to Argentina to enjoy nature.
Kal:	Exactly.
Peter:	*Visit the Vasa Museum, it's really fun.*
Molly:	We will:) Thanks :)
Peter:	Enjoy!
Summary:	Molly and Margaret are going to Sweden in January. Kal and Peter advise them to stay in Stockholm and visit Vasa Museum.

Figure 1: An example of dialogue with its summary. Green: nouns, *italic*: salient utterances.

One intuitive solution to summarizing dialogue content is to directly adopt existing summarization systems (Gehrmann et al., 2018; Zhang et al., 2020a; Zou et al., 2020) designed for well-structured text, such as news and scientific articles (Shang et al., 2018; Gliwa et al., 2019) or to employ hierarchical models to capture features from different turns of different speakers (Zhao et al., 2019;

[*]Work done during internship at JD.com.
[†]The first two authors made equal contributions. Corresponding to Yanyan Zou.

Proceedings of the 29th International Conference on Computational Linguistics, pages 6050–6056
October 12–17, 2022.

Zhu et al., 2020). Unfortunately, succinctly summarizing dialogue content presents new challenges due to intrinsic properties of conversations. Unlike the field of well-organized text merely from a single person, dialogues often consist of utterances coming from two or more interlocutors, where the conversations are often informal, verbose and repetitive, sprinkled with false-starts, backchannels, reconfirmations, hesitations, speaker interruptions (Sacks et al., 1978) and the key information is often scattered throughout the whole chat. The above properties of conversations make it difficult to concentrate on the scattered salient utterances.

Recent studies incorporate intrinsic information of dialogues to handle the challenges for summarizing dialogues, such as topic features (Liu et al., 2019b; Li et al., 2019; Chen and Yang, 2020; Liu et al., 2021a), dialogue acts (Goo and Chen, 2018), conversation stages (Chen and Yang, 2020) and coreference information (Liu et al., 2021b). The main idea of such existing summarization systems is to directly learn the salient information of the input dialogues with various architecture designed or extra knowledge added. Differently, this work proposes to train a dialogue summarization system by explicitly telling the model the unimportant/redundant pieces of an input dialogue, so that the model is able to focus less on the given negative hints and pay more attention to the salient information. To be specific, we design two strategies to construct negative examples, namely Noun Drop, and Salient Utterance Drop. Then, we design an unlikelihood objective to model the probability of producing the gold summary given a negative example. The model is then trained based on the summation of likelihood and unlikelihood objectives. Extensive experiments on the SAMSum dataset demonstrate that our proposed method outperforms baselines on both semantic matching and factual consistent based metrics. The human evaluation also proves the performance improvements of our simple yet effective method.

2 Method

2.1 Sequence-to-Sequence Learning

We consider the abstractive dialogue summarization task as a sequence-to-sequence learning problem. We use the Transformer (Vaswani et al., 2017) as our backbone architecture, where the model takes as input the dialogue utterances and generates a corresponding summary in an end-to-end fashion. To be specific, for a dialogue $D = (u_1, u_2, ..., u_{|D|})$, consisting of $|D|$ utterances, coupled with its corresponding summary $Y = (y_1, y_2, ..., y_{|Y|})$ in the length of $|Y|$, the goal is to learn the optimal model parameters θ and to estimate the conditional probability:

$$P_\theta(Y|D) = \prod_{i=1}^{|Y|} p_\theta(y_i|y_{1:i-1}, D) \quad (1)$$

where $y_{1:i-1}$ denotes the first $i-1$ tokens of the output sequence (i.e., $y_{1:i-1} = (y_1, y_2, ..., y_{i-1})$). Given the whole training set $(\mathcal{D}, \mathcal{Y})$, this model can be trained to maximize the log-likelihood by minimizing:

$$\mathcal{L}_{MLE}(\theta; \mathcal{D}, \mathcal{Y}) = -\sum_{(D,Y)\in(\mathcal{D},\mathcal{Y})} \log P_\theta(Y|D)$$

2.2 Unlikelihood Objective

We first introduce two strategies for constructing negative examples:

- **Noun Drop**: We simply remove all the nouns (e.g., named entities) appearing in dialogue D since most fact details (i.e., salient information) are presented in nouns, highlighted by green color in Figure 1.
- **Salient Utterance Drop**: An utterance is defined as a *salient* one when the ROUGE-2 (Lin, 2004) recall score between it and the gold summary is larger than zero. This strategy removes all the salient utterances on which the gold summary is grounded and the remaining utterances are concatenated in order to form a new dialogue content. The utterances marked in italic in Figure 1 are salient ones and removed from the dialogue to construct a negative example.

For each dialogue $D \in \mathcal{D}$, each strategy results in a single negative example, denoted as D', yielding a new set $\mathcal{D}'$. The unlikelihood objective is then calculated as:

$$\mathcal{L}_{UNL}(\theta; \mathcal{D}', \mathcal{Y}) = -\sum_{(D',Y)\in(\mathcal{D}',\mathcal{Y})} \log\left(1 - P_\theta(Y|D')\right)$$

Different from the unlikelihood training (Welleck et al., 2019) whose key idea behind is to decrease the model's generation probability of certain negative candidates conditioned on the original input

text, our unlikelihood objective aims to decrease the probability of producing the target summary given the negative input D'. The final loss for the sequence-to-sequence learning is then defined as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{MLE} + \mathcal{L}_{UNL} \\ &= -\sum_{(D,D'Y)\in(\mathcal{D},\mathcal{D}',\mathcal{Y})} [\log P_\theta(Y|D) \\ &+ \log\left(1 - P_\theta(Y|D')\right)]\end{aligned}$$

The goal is to minimize the loss $\mathcal{L}$, i.e., maximizing the probability of generating the summary Y given the original dialogue D, while minimizing the probability of producing Y given D', which is similar to the idea of contrastive learning. In this scenario, the negative examples D' can be considered as explicit negative cues to drive the model focus more on the salient information.

3 Experiments

3.1 Datasets

We evaluate our model on the widely-used dialogue summarization datasets, SAMSum. Such a dataset comprises of natural message-like conversations expressed in English written by two or more linguists, each of which is annotated with summary created by language experts (Gliwa et al., 2019). The training set consists of 14,732 dialogue-summary pairs, while the validation and test set contain 818 and 819 instances individually. We list the detailed data statistics of each split (i.e., training, validation, test) with regard to average tokens, utterances and speakers in Table 1.

3.2 Implementation Details

We adopted the sequence-to-sequence Transformer model as our backbone architecture, which is implemented using Fairseq toolkit[1] (Ott et al., 2019). To be specific, our model is initialized with a pre-trained sequence-to-sequence, i.e., BART (Lewis et al., 2020), . Thus they share the same architectures, a 12-layer encoder-decoder Transformer. Each layer has 16 attention heads, and the hidden size and feed-forward filter size is 1024 and 4096, respectively, resulting in 400M trainable parameters. The dropout rates for all layers are set to 0.1. The optimizer is Adam (Kingma and Ba, 2015) with β_1 = 0.9 and β_2 = 0.999. The peak learning rates for all experiments are set to $4e-5$ with 200 warmup steps. We also adopted the same learning rate schedule strategies as in Vaswani et al. (2017). The maximum number of tokens in each batch is 800. The model is trained for 4 or 5 epochs for different perturbation methods. Each epoch takes around 0.7 hours on single Tesla P40 GPU. To obtain all nouns in a dialogue, we applied the spaCy toolkit[2] to obtain the part-of-speech and named entities tags. When constructing negative examples where salient utterances are dropped, we simply adopt the ROUGE scores. All hyperparameters are set based on the performance of the validation set.

3.3 Baseline

- Lead3 is a commonly adopted method in the extractive document summarization task, which simply takes the first three leading sentences of an input text as its summary.
- PTGen (See et al., 2017) modifies a sequence-to-sequence generation model with the copy and coverage mechanisms to copy words originated from the input text.
- FastAbs-RL (Chen and Bansal, 2018) first selects pivot sentences and then generates abstract summary with reinforcement learning.
- DynamicConv + GPT-2/News (Wu et al., 2019) proposes a lightweight dynamic convolutions to replace the self-attention modules in the Transformer layers.
- BART (Lewis et al., 2020) is a pre-trained encoder-decoder Transformer model.
- MultiView BART (Chen and Yang, 2020) uses multi-view features to summarize dialogues.

3.4 Automatic Evaluation

To evaluate the effectiveness of the proposed model and compare it with other baselines, we adopted the full-length F1-based ROUGE scores (Lin, 2004) to measure the quality of summary output generated by different systems. Specifically, we used the files2rouge[3] package based on the official `ROUGE-1.5.5.pl` perl script to get the full-length ROUGE-1, ROUGE-2 and ROUGE-L F-measure scores. The recent popular automatic evaluation metric for text generation, BERTSCORE

[1]We empirically observed that different frameworks (e.g. Fairseq and Huggingface Transformer) may obtain different results even under the same hyperparameter settings.

[2]https://spacy.io/

[3]https://github.com/pltrdy/files2rouge Note that the ROUGE scores might vary with different ROUGE tookits.

Split	#Dial	#Speaker	#Turns	#Words (Dial)	#Words (Summary)
Train	14,732	2.40	11.17	83.90	20.35
Valid	818	2.39	10.83	83.26	20.14
Test	819	2.36	11.25	83.87	20.43

Table 1: Data statistics of the dialogue summarization dataset, SAMSum, including the total number of dialogues (#Dial), the average number of participants (#Speaker), the average number of turns (# Turns), the average number of words in the dialogue (# Words (Dial)) and in the summary (# Words (Summary)).

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTSCORE	QUESTEVAL
Lead3	31.4	8.7	29.4	-	-
PTGen	40.1	15.3	36.6	-	-
DynamicConv + GPT-2	41.8	16.4	37.6	-	
FastAbs-RL	42.0	18.1	39.2	-	-
DynamicConv + News	45.4	20.7	41.5	-	-
Multiview BART	**53.9**	28.4	44.4	53.0	40.3
BART	52.6	27.0	42.1	52.1	39.8
+ Noun Drop	53.4$*$	28.4$*$	**44.7**$*$	**53.5**	**41.6**
+ Salient Utterance Drop	53.2$*$	**28.7**$*$	44.6$*$	53.2	40.5

Table 2: Results on SAMSum test split. $*$ indicates the results are significantly different from BART baseline in terms of ROUGE scores ($p < 0.05$, according to the ROUGE script). The highest score is highlighted with **bold**, while the second highest is marked with underline.

(Zhang et al., 2020b), is also presented for comparisons. The above metrics mainly focus on the semantic similarity between the generated output and the ground truth, based on either string match or meaning similarity. Moreover, we also consider the QUESTEVAL (Scialom et al., 2021) to evaluate the summary's factual consistency. To be specific, given an input text (e.g., dialogue content in this paper) and a summary, QuestEval first extracts question answers (considering all the named entities and nouns) from either the input text or the generated summary, and then generates natural language questions from the input text or the summary correspondingly conditioned on the generated answers. A Question Answering (in short, QA) model is employed to consume the input text to answer the questions derived from the summary, resulting in a score, denoted as the PRECISION score. Such a score implies that a summary should contain only factual information consistent to the input text. Similarly, the QA model is also applied to address the questions generated from the input text, producing another score, namely the RECALL score, showing that the summary should contain the most important information from the source text. The final QuestEval score is the harmonic mean of the precision and recall, i.e., the F1-measure score. We adopted the version with learned weights for questions, which has proved high correlation with human judged consistency and relevance (Scialom et al., 2021).

As listed in Table 2, in terms of the semantic similarity-based metrics (i.e., ROUGE and BERTSCORE), the Noun Drop achieves highest ROUGE-L and BERTSCORE, while the Salient Utterance Drop obtains the highest ROUGE-2, demonstrating the effectiveness of negative cues with the unlikelihood objective. With regard to the factual consistency metric QUESTEVAL, the variant with Noun Drop obtained the highest score, which demonstrates its effectiveness to generate the factual consistent summaries since detailed fact are mainly presented in the form of named entities and nouns residing in the source input.

Overall, the variant with Noun Drop works the best for the three of five metrics. It is also worthy noting that MultiView BART requires extra topic segmentation algorithms to obtain the multi-view features, while our method only needs part-of-speech tags and ROUGE scores to construct negative examples which are easier to achieve.

We have also tried to combine the Noun Drop and Salient Utterance Drop. It is interesting that we did not obtained consistently improvement. One

Systems	1st	2nd	3rd	4th	MR
BART	0.04	0.12	0.34	0.51	3.34
MultiView BART	0.22	0.24	0.31	0.23	2.55
Ours	0.28	0.30	0.23	0.19	2.33
Gold	0.46	0.34	0.13	0.07	1.98

Table 3: Human evaluation on SAMSum: proportions of rankings. MR: mean rank (the lower the better).

possible reason is that the negative examples might lose too much information so that the negative signals become weaker.

3.5 Human Evaluation

We also elicit feedback from human efforts to evaluate the generated summaries from different summarization systems. We compared our best performing model (i.e. +Noun Drop) with the human references, as well as two baselines, BART (Lewis et al., 2020) and MultiView BART (Chen and Yang, 2020). We randomly select 100 dialogues from the test split of SAMSum dataset. To ensure fairness, for each dialogue, we list its candidate outputs in a random order, including human references (denoted as Gold), and outputs generated by three models. 10 participants are presented with a dialogue and its paired candidate summaries, where all participants are shown the same candidate order. For each selected dialogue, they are asked to rank the candidate output from the best to worst with regard to three criteria:

- *Fluency*: Is the summary fluent and grammatically correct?
- *Informativeness*: Does the summary contains the most informative pieces of the dialogue?
- *Succinctness*: Does the summary express in an abstractive way (e.g., without repetitions)?

Table 3 listed the proportions of different system rankings and mean rank (lower is better). The output of our proposed method is ranked as the most appropriate summary for 28% of all cases. Overall, we obtain lower mean rank than the other two systems but still lags behind the Gold one. The Fleiss' Kappa score (Fleiss, 1971) among participants is 0.527 that demonstrates fair inter-rater agreement.

4 Conclusion

Recent studies involved dialogue studies (e.g., topical information, coreference information, and dialogue acts) to make the model directly pay more attention to salient parts. However, the characteristics of dialogue content make it challenging to concentrate on scattered outstanding utterances. Rather, in this work, we propose a simple yet effective approach to explicitly tell a model the redundant pieces of a dialogue and thus focus more on the salient ones. We proposed two strategies to construct negative samples with redundant information and designed an unlikelihood objective to force the model learn less from redundant information, in other words, learning more from the salient pieces. Experiments on the benchmark dataset demonstrate the efficacy of the proposed model. In the future, we plan to investigate other strategies for constructing negative examples and replace the unlikelihood objective with the ranking loss.

5 Broader Impact Statement

Our simple yet effective abstractive dialogue summarization system could be used where there exists dialogue systems (two or multi-party dialogues). For example, it could be used for grasping the key points quickly or recapping on the salient information of online office meeting. In addition, the system can also be used for customer service, requiring employees to summarize the conversation records of customers' inquiries, complaints and suggestions.

The daily dialogue dataset used in this work is publicly available, and only for research purpose. There may exist biased views in them, and the content of them should be viewed with discretion.

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*.

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*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2020. From standard summarization to new tasks and beyond: Summarization with

manifold information. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

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*.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

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*.

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*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

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

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

Zhengyuan Liu, A. Ng, Sheldon Lee Shao Guang, AiTi Aw, and Nancy F. Chen. 2019b. Topic-aware pointer-generator networks for summarizing spoken conversations. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821.

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

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.

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*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, Min Yang, and Deng Cai. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, pages 3455–3461.

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*.

Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pre-training for abstractive document summarization by reinstating source text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.