Manipulating Emoticons and Data Augmentation for Abstractive Dialogue Summarization

Matteo Celia
Politecnico di Torino
S316607
s316607@studenti.polito.it

Paolo Favella
Politecnico di Torino
S320144
s320144@studenti.polito.it

Xiangbo Gong
Politecnico di Torino
S295754
s295754@studenti.polito.it

Gioele Costa
Politecnico di Torino
S318947
s318947@studenti.polito.it

***Abstract*—This project aims to extend previous studies that addressed the challenge of abstractive summarization of dialogues by harnessing the distinctive attributes of conversations that involve shared commonsense knowledge among participants. Starting from the SICK architecture, a framework that utilizes commonsense inferences as supplementary context we propose minor modifications to analyse their effect on the model. Unlike prior approaches that rely solely on input dialogue, SICK incorporates an external knowledge model to produce a diverse array of commonsense inferences, subsequently employing a similarity-based selection method to identify the most probable one. Thus, we conduct an analysis of the model's behavior, exploring variations such as word removal or replacement, as well as the manipulation of emoticons within the input, to gain insights into their effect on the performance, obtaining also relevant results.**

Our GitHub repository is available at: https://github.com/GongXiangbo/Extended_SICK_Summarization.git

I. Introduction

Abstractive dialogue summarization involves generating concise summaries while retaining the conversation's context. Unlike summarizing conventional documents like news articles or scientific papers, dialogue-to-document summarization faces challenges due to the disparity between input and output formats, making it harder to learn mapping patterns.

In this scenario one of the most challenging problem to solve is to reveal unspoken intentions to better understand an utterance. The dialogues treated by the model include chat-like dialogues. This specific kind of interactions maybe be non-trivial to properly formalize. Indeed these kind of dialogues often include terms and abbreviations that might be familiar to the two parties that are communicating by having some shared context which can be difficult to grasp from a model.

Moreover, emoticons might be useful for getting closer to grasp these underlying meanings. Even though the wide spread use of emoticons in the training dataset is not documented, we try to analyse their contribution to the understanding of the dialogues.

Indeed we first evaluate the impact of removing completely the emoticons present in the dialogues, and then employ a very simple way to substitute emoticons with some textual representation in order to evaluate their impact on the performance of the model.

It is worth noting that models do not consider the actual input as a whole and only look at certain parts of the input (this is the reason why SICK++ proposed an auxiliary task called commonsense supervision, to enforce the model to use commonsense knowledge), hence the way in which we tried to translate emoticons into their textual explanation might not be very effective.

Different data augmentation techniques will also be employed to try to improve the generalization capabilities of the model.

II. Problem Statement

In the realm of abstractive dialogue summarization, integrating commonsense knowledge into existing state-of-the-art language models presents a significant challenge. Current approaches often face issues due to the counterintuitive expansion of source contents and the lack of robust inferences when simply adding additional inputs to pre-trained language models. This undermines the core goal of condensing dialogues into succinct summaries.

Theoretical Formalization of the Problem:

- *Expected Input:* The expected input for our task consists of dialogues, wherein each dialogue comprises a series of conversational exchanges between two or more participants. Additionally, commonsense knowledge related to the dialogue context is provided as supplementary information.
- *Addressed Task:* Our task primarily focuses on abstractive dialogue summarization, aiming to distill the essence of the dialogues while preserving their contextual nuances. Moreover, we strive to effectively exploit information hidden into the emoticons with the hope to enhance the quality and relevance of the generated summaries and analyse their contribution to convey unspoken intents in dialogues.
- *Expected Output:* The expected output is a concise and informative summary that encapsulates the key points and underlying meanings conveyed within the input dialogues. These summaries should reflect an understanding of the dialogues' intent, incorporating relevant commonsense knowledge to enrich the summarization process and ensure coherence and relevance in the output summaries.

III. Methodology

In the following section we present the framework on which our analysis is based, describing its most important modules and methodologies. In order to do that it is necessary to provide an overview of the SICK (Summarizing with Injected Commonsense Knowledge) architecture (we will not treat SICK++). Then we are going to present some extensions that we provided to explore this framework.

A. SICK

The SICK (Summarizing with Injected Commonsense Knowledge) framework was proposed in [1] to address abstractive summarization of dialogues. The novelty of SICK is that it uses an external knowledge model to generate a rich set of commonsense inferences and selects the most probable one using a similarity-based selection method. To do so a commonsense knowledge model is used, COMET [2] and its extension, PARACOMET [3] which adopts an internal memory module to consider previous dialogue history when generating an output. The model generates commonsense inference based on the input and then filters them based on the similarity score between utterance and commonsense pair computed with SBERT [4]. The commonsense inference with the highest score for each utterance is then selected. Then, the dialogue and its corresponding set of commonsense inferences are cross concatenated using special tokens $< \mathbf{I} >$, $< /\mathbf{I} >$ in the following way:

$$X = D \oplus C = \ldots \| u_i \| < \mathbf{I} > c_i < /\mathbf{I} > \ldots$$

Where u_i represents the i-th utterance and c_i corresponds to the commonsense information related with that utterance.

SICK is built upon a transformer-based encoder-decoder architecture (BART language model). The encoder fuses the information from the two different modalities (i.e., dialogue and commonsense inference). By the stack of decoders, the encoder output is used for cross-attention with the summary.

B. Model replacing

As said before, SICK exploits BART, a transformer-based state-of-the-art denoising autoencoder for pretraining sequence-to-sequence models. In particular it is based on the Huggingface implementation of a BART language model using the weight checkpoint of BART-xsum. We tried employing BART-CNN instead which is a BART model pre-trained on English language, and fine-tuned on CNN dataset, to see how dependent the model is from the dataset used for fine-tuning.

C. Data augmentation

In order to try to improve the generalization capabilities and robustness of the model we applied different data augmentation strategies:

- Random Deletion: Randomly remove each word in the sentence with probability p [5].
- Random Replacement: Replace random words that are not stop words with one of their synonyms with probability p [5].

D. Emoticon Manipulation

It is clear that inside dialogues there exists information that can only be understood when the hidden meanings are revealed. Emoticons are usually used to either strengthen the meaning of what is said or to express some unspoken intent. Even though the presence of emoticons in the dialogues the model is trained on might not be that wide spread, could still be useful to exploit information expressed through them.

We first try to evaluate the effect on the performance when deleting emoticons. Then we use a very simple approach which map each emoticon to its textual explanation thus modifying the input to the seq2seq model. This process is applied during the concatenation of utterance and common inferences so after the generation of the commonsense inferences. That's because we wanted to exploit the already calculated commonsense inferences based on the original dialogues that are provided by the authors of the SICK paper.

IV. Experiments

A. Data description

In the experiment, we used the SamSum and DialogSum data sets, and we only used SamSum dataset in the extension part of data enhancement.

The SAMSum dataset is a dataset built for the task of conversation summarization [6]. It contains more than 16,000 dialogues from various scenes of daily life, such as home, work, leisure, etc. Each conversation comes with a hand-written summary that succinctly summarizes the key take-aways from the conversation.

The DialogSum dataset is also a dataset specially designed for the conversation summarization task [7]. It contains approximately 13460 multi-turn conversations covering a wide range of topics such as personal life, work, education, travel, etc. Similar to the SamSum dataset, each conversation in DialogSum is accompanied by a hand-written summary designed to accurately capture the core content and key points of the conversation.

B. Experimental design

Different from the original article, in all of our experiments, we used an training initial learning rate of 2e-5, and the batch size was 4, with a total of 32 epochs.

1) Hardware and Software: We used the Tesla T4 GPU provided by Google colab, the number of GPUs is 1.We used the latest version of PyTorch in our experiments. Other major libraries include the 4.13.0 version of transformers, the latest version of nltk, the latest version of datasets, and etc.

2) Validation method: We used the training set, validation set and test set of original data to conduct experiments. The model is trained on the training set, then the hyperparameters are adjusted on the validation set, and finally tested on the test set.

3) Performance metrics:

- ROUGE [8] scores: Including ROUGE-1, ROUGE-2, and ROUGE-L, which compare word-level unigrams and bigrams, respectively, and longest common sequence overlap with golden summaries. Additionally, the ROUGE-LSUM score was added to this list, which takes into account the longest common subsequence overlap between sentence-level summaries and gold summaries, providing an assessment of overall structural similarity.
- BERTScore [9]: It is the recent popular metric for text generation, which computes the contextual similarity score between generated and reference summaries.

For simplicity, we use R-1, R-2, R-L, R-LSUM, and B-S to denote ROGUE-1, ROUGE-2, ROUGE-L, ROUGE-LSUM, and BERTScore.

C. Results and Analysis

Table I and Table II show the experiment results on SAMSum and DialogSum datasets. For simplicity, we use RE, TE, RD, and RR to denote Remove Emoticons, Translate Emoticons, Random Deletion, and Random Replacement.

1) BART-large models: When the BART-large model is pretrained on xsum, it is more accustomed to the task of generating summaries, especially when high-level generalization and creative inference are required, which makes it perform better when processing the SAMSum and DialogSum datasets. SAMSum and DialogSum are both dialogue summarization tasks, which require the model to extract or generate a summary of key information from the dialogue, which is more similar to the training goal of the xsum dataset and requires a higher level of generalization and reasoning capabilities. In contrast, when a model is pre-trained on a CNN dataset, it is better at extracting key sentences as summaries, a skill that may not be as directly relevant or effective as the needs of conversation summarization. Dialogue summarization often requires dynamic information processing in the dialogue flow and a high-level summary of participants' utterances, which may explain why the BART-large model pre-trained on XSum performs better on the SAMSum and DialogSum datasets than on the CNN dataset pre-trained model. This observation highlights the importance of pre-training dataset selection for downstream task performance, and how model performance can be optimized by selecting pre-training datasets that better match the target task structure and requirements.

2) Remove Emoticons and Translate Emoticons: The performance of training using the Remove Emoticons on data set will drop slightly compared to the baseline, but the training performance using the Translate Emoticons on data set will improve some metrics compared to the baseline. This phenomenon illustrates several key points:

- Importance of Emoticons: Emoticons play an important role in text, especially in conveying emotion and context. When emoticons are removed, the model loses some clues to understanding the user's emotion or the context of the text, which may be the reason for the slight decrease in training performance.
- Advantages of textual emoticons: Translating emoticons into text can help the model better understand and process the emotional information in the text. This approach improves training performance by preserving emotional information while converting it into a text format that is easier for the model to process.

3) Random Deletion and Random Replacement: For Random Deletion of words, when p increases, that is, when more words are removed, the performance of the model on the training data set will decrease. This is because removing too many words will cause the text to lose more information, making it difficult for the model to capture sufficient language features and contextual information, affecting the learning effect. As for Random Replacement of words, when p is very small, that is, when only a few words are replaced, the model performance will be slightly improved. This may be because replacing a small number of words introduces a certain amount of noise, which helps the model learn a more robust feature representation, similar to a data augmentation effect. However, as p increases, that is, when more words are replaced, the model performance begins to decline, because a large number of replacements will destroy the semantic consistency and contextual information of the text, making it difficult for the model to learn effective features.

TABLE I
RESULT ON SAMSUM

Model	R-1	R-2	R-L	R-LSUM	B-S
Baseline (BART-large-xsum)	53.38	28.67	**44.27**	**49.11**	71.62
BART-large-cnn	40.09	19.96	30.82	36.98	66.03
BART-large-xsum + RE	53.28	28.64	44.05	48.92	71.62
BART-large-xsum + TE	**53.41**	**28.74**	44.18	49.10	**71.79**
BART-large-xsum + RD (p=0.01)	53.23	28.44	43.90	49.08	71.50
BART-large-xsum + RD (p=0.1)	51.93	26.79	42.55	47.38	70.78
BART-large-xsum + RD (p=0.2)	50.64	25.18	41.27	46.23	70.02
BART-large-xsum + RD (p=0.3)	50.15	24.76	40.67	45.70	69.63
BART-large-xsum + RD (p=0.4)	49.65	24.10	39.98	45.27	69.36
BART-large-xsum + RR (p=0.01)	53.39	28.51	44.12	49.00	71.72
BART-large-xsum + RR (p=0.1)	53.16	28.26	43.97	48.31	71.35
BART-large-xsum + RR (p=0.2)	52.80	27.45	43.52	48.52	71.35
BART-large-xsum + RR (p=0.3)	52.72	27.37	43.61	48.44	71.31
BART-large-xsum + RR (p=0.4)	52.63	27.79	43.56	48.38	71.26

TABLE II
RESULT ON DIALOGSUM

Model	R-1	R-2	R-L	R-LSUM	B-S
Baseline (BART-large-xsum)	**45.85**	**21.11**	**37.63**	**40.92**	**71.01**
BART-large-cnn	36.24	15.59	28.00	31.50	65.24

V. CONCLUSIONS

The experiments highlight the importance of pre-training dataset selection for downstream task performance, and how model performance can be optimized by selecting pre-training datasets that better match the target task structure and requirements.

When processing text data containing rich emotional expressions, rational use of this emotional information (such as by translating emoticons rather than simply removing them) can significantly improve the understanding and performance of NLP models. This also suggests that when designing NLP systems, it is important to consider how to handle non-traditional text information.

Random Deletion of words and Random Replacement of words these two situations illustrate that when training an NLP model, moderate data perturbation (such as the replacement of a small number of words) can be used as a regularization method to improve the generalization ability of the model, but excessive perturbation (whether removal or replacement) can harm model performance because it causes too much useful information to be lost or too much noise to be introduced. Therefore, reasonable selection of the value of p and balancing the relationship between information retention and noise introduction are very critical for optimizing model performance.

References

[1] Kim S, Joo S J, Chae H, et al. Mind the Gap! Injecting Commonsense Knowledge for Abstractive Dialogue Summarization[J]. arXiv preprint arXiv:2209.00930, 2022.

[2] Rei R, Stewart C, Farinha A C, et al. COMET: A neural framework for MT evaluation[J]. arXiv preprint arXiv:2009.09025, 2020.

[3] Gabriel S, Bhagavatula C, Shwartz V, et al. Paragraph-level commonsense transformers with recurrent memory[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(14): 12857-12865.

[4] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.

[5] Wei J, Zou K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks[J]. arXiv preprint arXiv:1901.11196, 2019.

[6] Gliwa B, Mochol I, Biesek M, et al. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization[J]. arXiv preprint arXiv:1911.12237, 2019.

[7] Chen Y, Liu Y, Chen L, et al. DialogSum: A real-life scenario dialogue summarization dataset[J]. arXiv preprint arXiv:2105.06762, 2021.

[8] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.

[9] Zhang T, Kishore V, Wu F, et al. Bertscore: Evaluating text generation with bert[J]. arXiv preprint arXiv:1904.09675, 2019.