applied sciences

MDPI

Article

Improving Abstractive Dialogue Summarization Using Keyword Extraction

Chongjae Yoo [1] **and Hwanhee Lee** [2,*]

[1] LG Electronics, Seoul 06772, Republic of Korea; chongjae.yoo@lge.com
[2] Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, Republic of Korea
* Correspondence: hwanheelee@cau.ac.kr

Abstract: Abstractive dialogue summarization aims to generate a short passage that contains important content for a particular dialogue spoken by multiple speakers. In abstractive dialogue summarization systems, capturing the subject in the dialogue is challenging owing to the properties of colloquial texts. Moreover, the system often generates uninformative summaries. In this paper, we propose a novel keyword-aware dialogue summarization system (KADS) that easily captures the subject in the dialogue to alleviate the problem mentioned above through the efficient usage of keywords. Specifically, we first extract the keywords from the input dialogue using a pre-trained keyword extractor. Subsequently, KADS efficiently leverages the keywords information of the dialogue to the transformer-based dialogue system by using the pre-trained keyword extractor. Extensive experiments performed on three benchmark datasets show that the proposed method outperforms the baseline system. Additionally, we demonstrate that the proposed keyword-aware dialogue summarization system exhibits a high-performance gain in low-resource conditions where the number of training examples is highly limited.

Keywords: abstractive summarization; dialogue summarization; keyword extraction

Citation: Yoo, C.; Lee, H. Improving Abstractive Dialogue Summarization Using Keyword Extraction. *Appl. Sci.* **2023**, *13*, 9771. https://doi.org/10.3390/app13179771

Academic Editors: Jae-Hoon Kim and Kichun Lee

Received: 30 June 2023
Revised: 17 August 2023
Accepted: 23 August 2023
Published: 29 August 2023

1. Introduction

Due to the COVID-19 pandemic, virtual meetings have rapidly increased [1]. Considering the amount of online meeting data, it is often necessary to rapidly determine the key points in these meeting records [2]. In this paper, we focus on the abstractive dialogue summarization task, which aims to capture the most critical part of the given dialogue and generate a short paragraph that can help people quickly understand the main contents of the dialogue [3]. One of the simplest ways to tackle this task is to use the existing summarization systems trained on widely used datasets such as CNN/DM [4] or XSum [5]. However, different from generating a summary for well-structured documents such as news articles or academic papers, generating a summary for the given dialogue requires additional consideration to the properties of colloquial texts [6]. Among them, the most representative characteristic of the colloquial text is that it often consists of multiple utterances from multiple speakers. This characteristic usually makes it difficult for the readers to grasp the speaker's information or catch the topic of the conversation. Also, topic shifts can frequently occur in long dialogues with multiple speakers. For such reasons, it is difficult to directly apply the existing summarization systems trained on widely used document summarization datasets, and researchers have released several task-specific dialogue summarization datasets such as SAMSum [7] and DialogSum [8].

However, the number of datasets in these systems is usually much lower than in previous summarization datasets. Hence, even though the systems are trained through these task-specific datasets, the systems often do not capture the speaker's information or misunderstand the topic of the conversation and produce very simple forms of sentences [9]. For instance, in Table 1, the baseline system often generates an uninformative summary consisting of only three words that includes the fragmentary facts of the dialogue.

Table 1. An example of summaries for the dialogue. Red color indicates extracted keywords.

Dialogue
Person1: What makes you think you are able to do the job? Person2: My major is Automobile Designing and I have received my master's degree in science. I think I can do it well. Person1: What kind of work were you responsible for the past employment? Person2: I am a student engineer who mainly took charge of understanding the corrosion resistance of various materials.
Summary
Person1 is interviewing Person2 about Person2's ability and previous experience.
Summary Without Keyword (Baseline)
Person1 interviews Person2.
Summary With Keyword (KADS)
Person1 asks Person2's major, the past work, and the reason to do the job.

To solve such problems in the dialogue summarization task, we propose a **k**eyword-**a**ware **d**ialgoue **s**ummarization system (KADS) that efficiently utilizes the keyword information using a keyword extractor. By leveraging the keyword information to the summarizer, we adopt the advantage of extractive summarization systems to the abstractive summarization systems. KADS first extracts the keywords from a dialogue using a state-of-the-art pre-trained keyword extractor such as keyBERT [10]. Then, we construct the input text by prepending extracted keywords after a special token <*keyword*> and inserting a segment token </*s*> before the dialogue text. And then we fine-tune the pre-trained encoder–decoder models like BART [11] to generate the summary of the given dialogue with the help of the keywords we added. Experimental results on three widely used dialogue summarization benchmark datasets show that our proposed KADS shows significant improvement over the baseline systems in ROUGE [12] metric, with a gain of about 2.7% in ROUGE-L. Also, through the qualitative analysis, we find that extracted keywords efficiently assist the system in generating main words, as shown in the summary generated by our system in Table 1. In this example, we can infer that keywords *"degree"* and *"responsible"* assist in generating the words *"major"* and *"reason to the job"*. Furthermore, we explore the usage of various keyword extractors on our system to find the best keyword extractor for dialogue summarization. Finally, we validate the performance of our keyword-aware summarization system in low-resource conditions where the number of the dataset is scarce, which often occurs in the dialogue summarization task. We demonstrate that our method is even more effective with larger performance improvement than baseline systems in these low-resource conditions. The main contributions of this study can be summarized as follows:

- We propose a novel keyword-aware abstractive summarization system that efficiently leverages the key information in a dialogue.
- We demonstrate that our proposed keyword-aware method outperforms baseline methods in three benchmark datasets.
- We explore the usage of various keyword extractors for dialogue summarization tasks to find the best usage.
- We demonstrate the effectiveness of the proposed keyword-aware method in low-resource conditions.

2. Related Works

2.1. Dialogue Summarization

A good summary characterizes a dialogue as a substitute for the original text considering not every sentence contains meaningful information [13]. However, most dialogue summarization datasets are in English, with very little data on daily conversations. Moreover,

owing to the lack of training data in dialogue summarization, learning vital information from the dialogue context becomes challenging. Fu et al. [14] discussed the limited number of words in extractive summarization and the slight difference between the input and target summaries owing to the limitations of the unsupervised methodology [15]. Unlike the unsupervised approach that makes qualitative evaluation difficult, the supervised approach can be easily evaluated [16] even if there is no sufficient database for dialogue summarization. Hence, in our work, we focus on using the dialogue summarization datasets that include human labels, such as DialogSum [8], SAMSum [7], and TweetSumm [17]. Recently, research on improving the performance of dialogue summarization systems using these datasets has been widely conducted. For example, some researches [18] have improved the summarization performance by making them aware of the structure or introducing underlying knowledge similar to the approaches in document summarization system [19]. However, this method does not migrate to an existing model easily [20]. Furthermore, summarized text may occasionally not include valid keywords, even if a keyword is present [21]. Therefore, a method for migrating existing methods simply while confirming the qualitative evaluation improvements is required. Owing to these limitations, the performance of the generative summary has not improved significantly. Nevertheless, the proposed method can improve the performance of generative summaries by simply making changes to the input by using keywords without changing the model.

2.2. Keyword-Aware Summarization

Zhong et al. [22] have shown that using keywords is beneficial for extractive text summarization systems. Recently, Bharti et al. [23] utilize keywords in abstractive document-level summarization tasks [24]. From these works, we can infer that keywords can reduce redundant information in a text to generate summaries efficiently. By focusing on these points, Li et al. [21] proposed keyword-guided selective mechanisms to improve the source encoding representations for the summarization system. The decoder in this system can dynamically combine the information of the input sentence and the keywords to generate summaries. And Liu et al. [25] proposed a method of extracting a set of prominent sentences from an input document for the summary to generate an improved summary. Compared to the previous systems, these keyword-aware methods improved performance depending on the various keyword-aware techniques. Nonetheless, these keyword-aware systems were not only validated on the document summarization system, and they did not show any meaningful performance indicators in other domains, such as the dialogue summarization system. Unlike previous works, our work focuses on dialogue summarization compared to previous document summarization systems. Also, our work confirms that the keyword extractor shows a significant performance improvement in the summary without changing the model architecture.

2.3. Keyword Extractor

In our work, we explore the usage of various keyword extractors for the proposed keyword-aware method. We briefly explain each keyword extractor we used in the following section.

2.3.1. KeyBERT

KeyBert [10] is a keyword extractor based on the self-supervised contextual retrieval system that uses BERT [26] embeddings and simple cosine similarity to identify the subphrases in a document most similar to the document itself. It feeds the sentence S to BERT and obtains the contextual feature vector W as follows:

$$W = \{w_1, w_2, ..., w_n\} = BERT(S) \quad (1)$$

The vectors of words in a sentence are averaged to acquire its sentence embedding vector. Subsequently, the method picks the words close to the sentence embedding vector to ensure the keyword captures the sentence's meaning. Finally, the similarity of the embeddings to

the sentence embedding is obtained using the cosine similarity metric.

$$Sim_i = cos(w_i, W) \quad (2)$$

Here, Sim_i is the cosine similarity between the word embedding vector w_i of a word i and the sentence embedding vector. Once the candidate keywords are extracted, it obtains their keyphrases through the rule of adjacent keywords.

2.3.2. RaKUn

RaKUn [27] refers to a rank-based keyword extraction via unsupervised learning and meta vertex aggregation. RaKUn uses graph-theoretic measures to identify the keywords using meta vertices and specially designed redundancy filters. RaKUn showed the highest performance on Facebook's fasttext benchmark dataset [28].

2.3.3. RAKE

RAKE [29], which stands for Rapid Automatic Keyword Extraction, is a highly efficient keyword extraction method that operates on individual documents to enable the application to dynamic collection. It employs word frequency, word degree, and the ratio of degree to frequency to extract keywords. RAKE shows fast speed, as its name suggests, and has already been used in various fields.

2.3.4. YAKE

YAKE [30] is a lightweight unsupervised automatic keyword extraction method relying on statistical text features extracted from single documents to select the most important keywords of a text. The algorithm removes similar keywords and retains the more relevant one (one with a lower score). The similarity is computed with the Levenshtein similarity [31], Jaro–Winkler similarity [32], or the sequence matcher. Finally, the list of keywords is sorted based on their scores.

2.3.5. PKE

PKE [33] is an open-source Python-based keyphrase extraction toolkit that contains various keyword extraction models. This toolkit provides an end-to-end keyphrase extraction pipeline in which each component can be easily modified or extended to develop new approaches. This toolkit is widely used due to the simple usage of various keyword extraction methods through the python library.

3. Method

3.1. Problem Formulation

For a given dialogue D that consists of n turns D = {(u_1), (u_2), ...,(u_j)}, the task of dialogue summarization aims to generate a short summary S for D. In other words, dialogue summarization aims to train a system that maximizes conditional probability $P(S|D;\theta)$. In addition to the summary, we formalize the dialogue summarization problem with additional input keywords from the pre-trained keyword extractor E. To develop a dialogue summarization system, we train a seq2seq model based on pre-trained language models such as BART [11]. We extract keywords K from each utterance D and aggregate them to the input of the dialogue summarization system to build a keyword-aware dialogue summarization system. In short, the final goal of keyword-aware dialogue summarization is to maximize the conditional probability $P(S|D, K;\theta)$, where $K = E(D)$. The overall flow of our keyword-aware dialogue summarization system is depicted in Figure 1. Our system consists of a *keyword extractor* and *keyword-aware summarizer* based on pre-trained language models. We use a keyword extractor to change the input as in Algorithm 1 and propose improved summarization using a pre-trained language model.

Algorithm 1: Flow of keyword aggregation algorithm

Data: D: input dialogue text, K: extracted keywords
Result: S^*: Dialogue texts with keywords
Let S^* be an empty list $S^* = []$;
for *each* $d \in D$ **do**
  Let K be extracted keywords from d
  Let S be a new string $S =$ <keyword>;
  for *each* $k \in K$ **do**
    $S = S + k +$ </s>
  end
  $S = S +$ <dialogue> $+ d$;
  $S^*.add(S)$
end
return S^*;

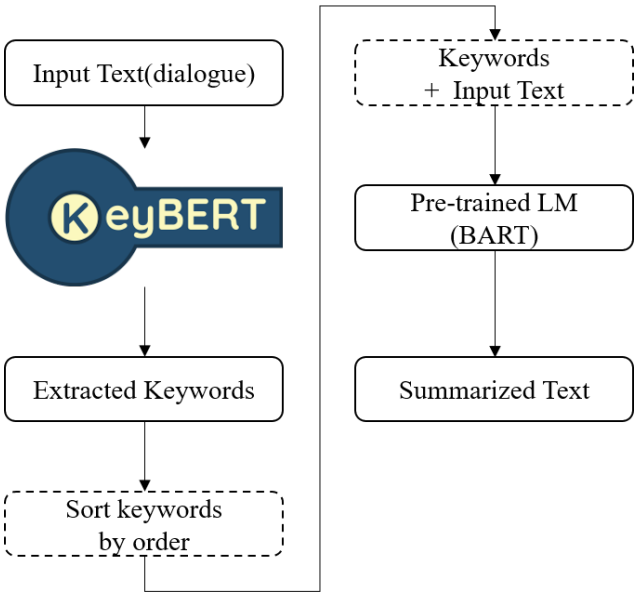

Figure 1. Overall flow of keyword-aware dialogue summarization.

3.2. Pre-Trained Language Models

Our proposed summarizer is built upon seq2seq-based pre-trained language models (LMs) BART and T5. BART is a transformer [34] based seq2seq model for various natural language processing tasks. BART combines bidirectional and autoregressive training techniques, which is effective for both natural language generation and understanding tasks. BART is pre-trained to reconstruct the original input text from the noisy or corrupted text. T5, which stands for "Text-to-Text Transfer Transformer", is also a seq2seq pre-trained LM built upon the transformer architecture. T5 is pre-trained on a large corpus, learning to generate masked parts in the input text. We fine-tune pre-trained LM like BART and T5 for dialogue summarization using the task-specific datasets for our work.

3.3. Keyword-Aware Summarizer

We propose a keyword-aware summarizer that efficiently utilizes the information from various keyword extractors as explained in Section 2.3. We depict the overall architecture of our proposed keyword-aware summarizer in Figure 2. After adding a keyword as a special token to the input dialog, embed it internally using the pre-trained LM and create a new summarized dialogue without any additional model changes through encoder-decoder, seq2seq. We first extract the keywords $K = \{k_1, k_2, ..., k_m\}$ from keyword extractor using a dialog D as follows. Recently, various keyword extractors such as KeyBERT [10] exhibit high

performance, but to develop a keyword-aware dialogue summarization (KADS) system, choosing a suitable keyword extractor is necessary. Thus, we explore the usage of various keyword extractors for the main component of our keyword-aware summarization system.

$$K = \{k_1, k_2, ..., k_m\} = E(D) \tag{3}$$

We find that the order of keywords affects the performance of the keyword-aware summarizer. Hence we adjust the order of extracted keywords by the order of occurrence in the dialog, which shows the best results from our experiments. Specifically, we sort keywords K in the order of occurrence in dialogue D as shown in Algorithm 2.

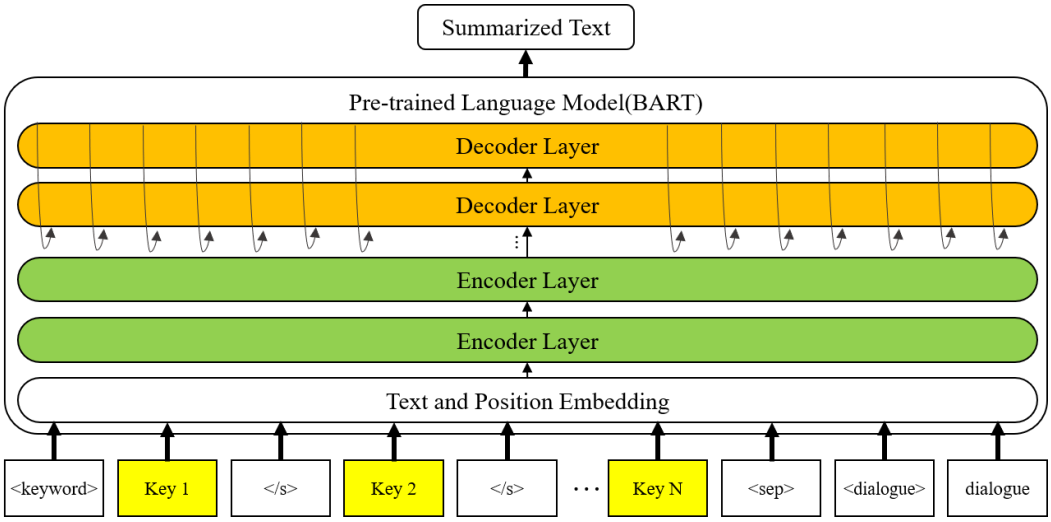

Figure 2. The overall architecture of our proposed keyword-aware dialogue summarization system.

And then, we aggregate the keywords K with the dialogue D to construct the input of BART to utilize keywords in summarization as in Algorithm 1. Specifically, we first add <*keyword*> and <*dialogue*> as special tokens to the input text. After extracting n keywords using a keyword extractor, we put n keywords in order as each keyword is separated into </*s*> to the input. And we append the <*sep*> token to the end of keywords, then add the <*dialogue*> token at the beginning of the original input, the dialogue text.

Algorithm 2: Flow of keyword order algorithm

Data: D input dialogue text
Result: K^* extracted keywords
Let K be a keyword list from keyword extractor
Let O be an empty dict O = {};
for *each* $k \in K$ **do**
  Get k's index in D and set key on O
end
Order by O's key and set to value in K^*
return K^*;

Finally, we fine-tune the pre-trained language model to generate a summary S for a given dialogue using keywords K as follows.

$$S = \{s_1, s_2, ..., s_n\} = BART([K, D]) \tag{4}$$

4. Experiments

4.1. Datasets

We used three public dialogue summarization benchmark datasets [35], DialogSum, SAMSum, and TweetSumm [17]. As shown in Table 2, the number of dialogues for Di-

alogSum and SAMSum are significantly larger than in TweetSumm. And we argue that DialogSum is the most appropriate dataset for our research for the following reasons. First, summarizing daily spoken dialogues from the perspective of downstream applications should help both businesses and personal requirements. For example, dialogue summaries help personal assistants keep track of complex procedures such as business negotiations. Also, from the perspective of the method, DialogSum has a larger scale of long dialogue data, which can facilitate the study of dialogue summarization using deep neural network-based methods. Furthermore, while most dialogue datasets often have insufficiently lengthy dialogue or unspoken daily conversations based on chat dialogues, DialogSum represents a real-life dialogue by mitigating these limitations. For these reasons, we choose DialogSum for our main experiment.

Table 2. Types of abstractive dialogue summarization datasets.

Datasets	Style	Scenario	Dialogues	# of Examples
DialogSum	spoken	daily life	13,460	1.8 M
SAMSum	written	online	16,369	1.5 M
TweetSumm	written	online	1100	1.8 M

4.2. Implementation Details

We chose four widely used pre-trained language models as the backbone of our keyword-aware summarizer and compared their performance. BART is an encoder–decoder transformer model that is pre-trained on a large corpus. And T5 [36] is also a pre-trained encoder–decoder system that treats all NLP tasks as text-to-text problems and allows the same model, objective, training procedure, and decoding process to be applied to various downstream tasks.

As shown in Table 3, the BART-large model showed the best performance ROUGE score among various models. And DialogSum contains 13,460 dialogues, which are divided into training (12,460), validation (500), and test (500) sets. We use a large version of BART for a conversation summary and fine-tune it into 5000 training steps/200 warm-up steps, and set the initial learning rate to 3×10^{-5}. We compute the average score by running ten times for each experiment.

Table 3. Performance comparison using several types of BART and T5 on DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline			
BART-base	44.8874	19.6440	37.0678
BART-large	46.1996	21.0814	38.8086
T5-base	41.5242	16.4631	33.6254
T5-large	42.1325	17.3326	34.4822
KADS			
BART-base	45.9874	20.9440	38.1678
BART-large	47.2237	22.1353	39.8665
T5-base	44.2605	18.8368	36.2043
T5-large	45.2232	18.9618	37.7235

4.3. Performance Comparison

We presented the dialogue summarization performance of each summarization system in Table 3. We used BART as a baseline system and also experimented with T5 [36]. We observed that our proposed KADS showed improvement over baseline systems in all cases. As shown in Table 3, the performance improved by approximately 2.7% compared to baseline and KADS on BART-large, 7.6% in T5-base, and 9.4% in T5-large, respectively. And the results show that the lower the performance of the baseline model, the greater the improvement through our proposed keyword-aware method. However, if only keywords

were extracted and applied, the performance improvement was negligible and improved significantly depending on how keywords were sorted. Also, we observed that performance varies depending on the type of keyword extractor.

4.4. Ablations

4.4.1. Keyword Extractor

We extract the keyword of the dialogue using a pre-trained keyword extractor and then use a special token to make an input for BART based summarizer. In this process, the accuracy of the pre-trained keyword extractor is critical for the performance of the keyword-aware summarization system. And for these keyword extractors, we first set default parameters of these extractors to train a keyword-aware summarization system and measure the performance of the summarizer. And we choose six widely used keyword extractors for comparison and represent the results in Table 4. We observed that the rapid automatic keyword extraction (RAKE) extractor performed best. However, the performance varied depending on the parameters of each keyword extractor. Hence, we proceeded with the experiment with the parameters showing the best performance for each keyword extractor.

Table 4. Performance comparison according to the keyword extractor type on DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L
KADS			
KeyBERT	47.2237	22.1353	39.8665
RAKUN	45.8668	20.9832	38.3474
RAKE	46.2899	21.0008	38.9808
YAKE	46.2077	20.9943	38.5658
PKE	45.8123	20.7648	38.3878

4.4.2. Keyword Selection Strategy

We explored the cause behind the similarity in results generated by different keyword extractors. We found that if it were to diversify the keywords/keyphrases, they would be less likely to represent the document collectively. Hence, to diversify our results, we conducted experiments on a delicate balance between the accuracy of keywords/keyphrases and their diversity. We used two algorithms to diversify our results:

- Max Sum Similarity [37];
- Maximal Marginal Relevance [38].

The maximum sum distance between pairs of data refers to the maximized distance between pairs of data. This method tries to maximize the candidate's similarity to the document while minimizing the similarity between candidates. Max Sum Similarity method selects the top 20 keywords/keyphrases and picks five that are the least similar to each other. We also investigated the maximal marginal relevance (MMR) method, which minimizes redundancy and the diversity of the effects on text summarization tasks. We use a keyword extraction algorithm called EmbedRank [39] which implements MMR for diversifying keywords/keyphrases.

As shown in Table 5, we observed that the performance of Max Sum similarity for keyBERT was higher than that based on the Max Sum simplicity of keyBERT.

Table 5. Performance comparison on different keyword selection strategy for DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L
KADS			
KeyBERT-MaxSum	47.2237	22.1353	39.8665
KeyBERT-MMR	46.6064	21.4615	38.9653

4.4.3. Keyword Order

Additionally, most keywords obtained by the keyword extractors were listed in order of the highest accuracy. However, in the case of dialogue, considering the meaning was revealed in a series of flowing interactions, the order in which keywords appeared was assumed to be more meaningful than the importance of the keywords.

Table 6 shows that the accuracy can be increased by rearranging the keywords in order of appearance in the dialogue than by the previous keyword accuracy. Also, ROUGE was mainly used for summarization. Ultimately, considering it is an index that evaluates the matching of strings, numerous questions are raised about the metric of translation/summary. Therefore, after obtaining contextual embedding using BERT, we checked whether the performance improved when BERTScore [40], which uses cosine similarity, was applied. Like conventional metrics, BERTScore calculates the similarity score between the reference and candidate sentences. However, instead of determining the exact match, it calculates the token similarity using contextual embedding. Table 6 shows that using KADS significantly improves the performance, even in BERTScore.

Table 6. Performance among the criteria for determining order of the keywords in the input text on DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
KADS				
Appearance	47.2237	22.1353	39.8665	0.9192
Accuracy	46.8676	21.9106	39.1413	0.9190

4.5. Analysis

4.5.1. Experiments on Other Dataset

The summary technique can be applied differently depending on the domain, and because of these characteristics, the best performance for each domain is different [41]. We also validate our keyword-aware summarization system to other dialog summarization datasets SAMSum and TweetSumm. SAMSum was in the form of an unrefined raw dialogue characterized by a mixture of terse dialogue and the frequent appearance of meaningless words. In addition, a considerable part of the customer consultation content in TweetSumm is already summarized. But, as shown in Table 7, our keyword-aware method improved performance. However, the changes were relatively minimal compared to DialogSum. Since BART was used for refining the document, the performance improvement was not significant in the data set that has already been summarized or in which many stopwords appear.

Table 7. Comparison of KADS performance on SAMSum and TweetSumm dataset.

Model	ROUGE-1	ROUGE-2	ROUGE-L
SAMSum			
Baseline	51.9170	27.6903	43.3052
KADS	52.0063	27.9083	43.4162
TweetSumm			
Baseline	42.2314	19.2241	35.5624
KADS	42.3342	19.3244	35.7624

4.5.2. Computation Cost

While applying keywords may improve performance, training time can be increased, and this may result in computational inefficiency. In general, when considering the performance and trade-offs, we refer to *memory*, *computation time*, and *storage*. However, as memory and storage margins increased, time costs became relatively significant, and thus we only compared the computation time with the performance [42]. Table 8 shows

that applying keyword extractors increased the training time, which varied significantly depending on the extractor used, making it essential to select an appropriate model.

Table 8. Comparison of training time in DialogSum. We measured the total training time and step per time.

Model	Total Time	Step per Second
Baseline	1574.0059	2.4750
KADS	2556.2950	3.0470

4.5.3. Keyword Verification on KADS

We discovered that inserting a keyword as a special token may increase the weight of a particular BART parameter [43]. For this reason, we study the effects of keyword input on the performance by adding various keywords by randomly extracting words from the dialogue to use them for keywords, as shown in Table 9. We calculate the average score by running ten times by random keyword selection. Obviously, we observed a decreased performance compared to KADS, as shown in Table 10. Even we confirmed that these random keyword methods performed worse than the baseline system that does not utilize keyword information.

Table 9. Comparison between using random keywords and KADS on DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline	46.1996	21.0814	38.8086
Random	44.3636	20.0146	37.1262
KADS	47.2237	22.1353	39.8665

Table 10. An example of summaries for comparing KADS and random keywords. Red color indicates extracted keywords.

Dialogue
Person1: What makes you think you are able to do the job? Person2: My major is Automobile Designing and I have received my master's degree in science. I think I can do it well. Person1: What kind of work were you responsible for the past employment? Person2: I am a student engineer who mainly took charge of understanding the corrosion resistance of various materials.
Reference Summary
Person1 is interviewing Person2 about Person2's ability and previous experience.
Baseline
Person1 interviews Person2.
KADS
Person1 asks Person2's major, the past work, and the reason to do the job.
Random keyword
Person2 says I am a student engineer who mainly took charge of understanding of the mechanical strength and corrosion resistance of various materials. I think I can do it well.

4.5.4. Low-Resource Conditions

Generally, the number of training datasets for dialogue summarization tasks is relatively small compared to document summarization datasets. Hence, it is often difficult to train a task-specific system for dialogue summarization tasks, and it is especially common for dialogue summarization task [44]. Therefore, we investigate whether our proposed keyword-aware method is efficient for low-resource conditions where the number training dataset is not enough. To validate the performance of the proposed system in the low-resource scenario, we train a system using various portions of the training dataset and present the results in Table 11. And we find that our proposed keyword-aware summarization system is especially effective compared to the baseline systems for this low-resource condition. Especially we find that the gap between the baseline and our proposed KADS generally increases as the number of training datasets decreases.

Table 11. Performance comparison with the systems trained with full dataset and trained with half of the datasets randomly sampled from DialogSum.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline			
100%	46.1996	21.0814	38.8086
75%	44.8136	20.6273	38.0324
50%	44.0717	20.1916	36.6705
25%	43.8293	20.0013	36.0213
10%	41.2341	18.7535	34.3425
KADS			
100%	47.2237 (+1.0241)	22.1353 (+1.0539)	39.8665 (+1.0579)
75%	46.2476 (+1.4340)	21.2876 (+0.6603)	39.2494 (+1.2170)
50%	45.2720 (+1.2003)	20.5406 (+0.3490)	38.1776 (+1.5071)
25%	45.9331 (+2.1038)	20.9613 (+0.9600)	37.7504 (+1.7291)
10%	43.2545 (+2.0204)	19.6724 (+0.9189)	36.0252 (+1.6827)

5. Conclusions

We proposed a dialogue summarization system, KADS, that efficiently utilized keyword information to improve the performance of dialogue summarization systems. We showed that the keyword extractor performance could significantly affect the results of dialogue summarization. Experimental results on widely used dialogue summarization datasets indicated that our proposed keyword-aware dialogue summarization showed improvement over baseline systems. We believe that the performance of KADS can be additionally improved if a superior keyword extractor is proposed in the future. Also, we showed that our proposed system is especially efficient in low-resource conditions.

Author Contributions: Conceptualization, C.Y. and H.L.; methodology, C.Y. and H.L.; software, C.Y.; validation, C.Y. and H.L.; formal analysis, C.Y. and H.L.; investigation, H.L.; resources, C.Y.; data curation, C.Y.; writing—original draft preparation, C.Y. and H.L.; writing—review and editing, H.L.; visualization, C.Y. and H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Chung-Ang University Research Grants in 2023. This research was supported in part by Institute for Information & Communications Technology Planning & Evaluation (IITP) through the Korea government (MSIT) under Grant No. 2021-0-01341 (Artificial Intelligence Graduate School Program (Chung-Ang University)).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KADS	Keyword-Aware Dialogue Summarization system
BERT	Bidirectional Encoder Representations from Transformer
LM	Language Model
RaKUn	Rank-based Keyword extraction via Unsupervised learning and meta vertex aggregation
RAKE	Rapid Automatic Keyword Extraction
PKE	Python-based Keyphrase Extraction
BART	Bidirectional Auto-Regressive Transformers
T5	Text-To-Text Transfer Transformer
ROUGE	Recall-Oriented Understudy for Gisting Evaluation

References

1. Pratama, H.; Azman, M.N.A.; Kassymova, G.K.; Duisenbayeva, S.S. The Trend in using online meeting applications for learning during the period of pandemic COVID-19: A literature review. *J. Innov. Educ. Cult. Res.* **2020**, *1*, 58–68. [CrossRef]
2. Zhong, M.; Liu, Y.; Xu, Y.; Zhu, C.; Zeng, M. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *arXiv* **2021**, arXiv:2109.02492.
3. Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; Dolan, B. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv* **2019**, arXiv:1911.00536.
4. Nallapati, R.; Zhou, B.; dos Santos, C.; Gulçehre, Ç.; Xiang, B. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 280–290. [CrossRef]
5. Narayan, S.; Cohen, S.B.; Lapata, M. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 1797–1807. [CrossRef]
6. Lee, S.; Yang, K.; Park, C.; Sedoc, J.; Lim, H. Who speaks like a style of Vitamin: Towards Syntax-Aware DialogueSummarization using Multi-task Learning. *IEEE Access* **2021**, *9*, 168889–168898. [CrossRef]
7. Gliwa, B.; Mochol, I.; Biesek, M.; Wawer, A. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv* **2019**, arXiv:1911.12237.
8. Chen, Y.; Liu, Y.; Chen, L.; Zhang, Y. DialogSumm: A Real-Life Scenario Dialogue Summarization Dataset. *arXiv* **2021**, arXiv:2105.06762
9. Lee, D.; Lim, J.; Whang, T.; Lee, C.; Cho, S.; Park, M.; Lim, H.S. Capturing Speaker Incorrectness: Speaker-Focused Post-Correction for Abstractive Dialogue Summarization. In Proceedings of the Third Workshop on New Frontiers in Summarization, Online, 10 November 2021; pp. 65–73.
10. Grootendorst, M. KeyBERT: Minimal Keyword Extraction with BERT. Available online: https://maartengr.github.io/KeyBERT/ (accessed on 28 August 2023).
11. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. 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, Online, 5–10 July 2020; pp. 7871–7880.
12. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
13. Moratanch, N.; Chitrakala, S. A survey on abstractive text summarization. In Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–7. [CrossRef]
14. Fu, X.; Zhang, Y.; Wang, T.; Liu, X.; Sun, C.; Yang, Z. RepSum: Unsupervised Dialogue Summarization based on Replacement Strategy. 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), Virtual Event, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 6042–6051. [CrossRef]
15. Liu, J.; Hughes, D.J.D.; Yang, Y. Unsupervised Extractive Text Summarization with Distance-Augmented Sentence Graphs. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 2313–2317.
16. Collins, E.; Augenstein, I.; Riedel, S. A Supervised Approach to Extractive Summarisation of Scientific Papers. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 195–205. [CrossRef]
17. He, R.; Zhao, L.; Liu, H. TWEETSUM: Event oriented Social Summarization Dataset. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 5731–5736. [CrossRef]

18. Feng, X.; Feng, X.; Qin, B. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. In *China National Conference on Chinese Computational Linguistics*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 127–142.
19. Qu, C.; Lu, L.; Wang, A.; Yang, W.; Chen, Y. Novel multi-domain attention for abstractive summarisation. *CAAI Trans. Intell. Technol.* **2022**. [CrossRef]
20. Zhao, L.; Yang, Z.; Xu, W.; Gao, S.; Guo, J. Improving Abstractive Dialogue Summarization with Conversational Structure and Factual Knowledge. Available online: https://openreview.net/forum?id=uFk038O5wZ (accessed on 28 August 2023).
21. Li, H.; Zhu, J.; Zhang, J.; Zong, C.; He, X. Keywords-guided abstractive sentence summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8196–8203.
22. Zhong, N.; Liu, J.; Yao, Y. *Web Intelligence*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2003.
23. Bharti, S.K.; Babu, K.S. Automatic keyword extraction for text summarization: A survey. *arXiv* **2017**, arXiv:1704.03242.
24. Li, C.; Xu, W.; Li, S.; Gao, S. Guiding generation for abstractive text summarization based on key information guide network. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 2, (Short Papers), pp. 55–60.
25. Liu, Y.; Jia, Q.; Zhu, K. Keyword-Aware Abstractive Summarization by Extracting Set-Level Intermediate Summaries. In Proceedings of the Web Conference 2021, Virtual Event, 12–23 April 2021; Association for Computing Machinery: New York, NY, USA, 2021; WWW '21, pp. 3042–3054. [CrossRef]
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; (Long and Short Papers); Association for Computational Linguistics; Volume 1, pp. 4171–4186. [CrossRef]
27. Skrlj, B.; Repar, A.; Pollak, S. RaKUn: Rank-based Keyword extraction via Unsupervised learning and Meta vertex aggregation. In *Statistical Language and Speech Processing: 7th International Conference, SLSP 2019, Ljubljana, Slovenia, 14–16 October 2019*; Proceedings 7; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 311–323.
28. Martinc, M.; Škrlj, B.; Pollak, S. TNT-KID: Transformer-based neural tagger for keyword identification. *Nat. Lang. Eng.* **2022**, *28*, 409–448. [CrossRef]
29. Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic keyword extraction from individual documents. *Text Min. Appl. Theory* **2010**, *1*, 1–20.
30. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword extraction from single documents using multiple local features. *Inf. Sci.* **2020**, *509*, 257–289. [CrossRef]
31. Yujian, L.; Bo, L. A normalized Levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1091–1095. [CrossRef] [PubMed]
32. Winkler, W.E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. 1990. Available online: https://eric.ed.gov/?id=ED325505 (accessed on 28 August 2023).
33. Boudin, F. pke: An open source python-based keyphrase extraction toolkit. In Proceedings of the COLING, Osaka, Japan, 11–16 December 2016.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30* . https://arxiv.org/abs/1706.03762.
35. Feng, X.; Feng, X.; Qin, B. A Survey on Dialogue Summarization: Recent Advances and New Frontiers. *arXiv* **2021**, arXiv:2107.03175.
36. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
37. Werner, T. A Linear Programming Approach to Max-Sum Problem: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1165–1179. [CrossRef] [PubMed]
38. Guo, S.; Sanner, S. Probabilistic latent maximal marginal relevance. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 833–834.
39. Bennani-Smires, K.; Musat, C.; Hossmann, A.; Baeriswyl, M.; Jaggi, M. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In Proceedings of the 22nd Conference on Computational Natural Language Learning, Brussels, Belgium, 31 October–1 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 221–229. [CrossRef]
40. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. *arXiv* **2019**, arXiv:1904.09675.
41. Kan, M.Y.; McKeown, K. Information Extraction and Summarization: Domain Independence through Focus Types. 1999. Available online: http://www.cs.columbia.edu/nlp/papers/1999/kan_mckeown_99.pdf (accessed on 28 August 2023).
42. Baker, S.G. The summary test tradeoff: A new measure of the value of an additional risk prediction marker. *Stat. Med.* **2017**, *36*, 4491. [CrossRef] [PubMed]

43. Carnegie, N.B.; Wu, J. Variable Selection and Parameter Tuning for BART Modeling in the Fragile Families Challenge. *Socius* **2019**, *5*, 2378023119825886. [CrossRef]
44. Zou, Y.; Zhu, B.; Hu, X.; Gui, T.; Zhang, Q. Low-Resource Dialogue Summarization with Domain-Agnostic Multi-Source Pretraining. *arXiv* **2021**, arXiv:2109.04080.