

Article

Strong Influence of Responses in Training Dialogue Response Generator

So-Eon Kim, Yeon-Soo Lim and Seong-Bae Park *

Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, Korea; sekim0211@khu.ac.kr (S.-E.K.); dladustn95@khu.ac.kr (Y.-S.L.)

* Correspondence: sbpark71@khu.ac.kr

Abstract: The sequence-to-sequence model is a widely used model for dialogue response generators, but it tends to generate safe responses for most input queries. Since safe responses are unattractive and boring, a number of efforts have been made to make the generator produce diverse responses, but generating diverse responses is yet an open problem. As a solution to this problem, this paper proposes a novel response generator, Response Generator with Response Weight (RGRW). The proposed response generator is a transformer-based sequence-to-sequence model of which the encoder is a pre-trained Bidirectional Encoder Representations from Transformers (BERT) and the decoder is a variant of Generative Pre-Training of a language model-2 (GPT-2). Since the attention on the response is not reflected enough at the transformer-based sequence-to-sequence model, the proposed generator enhances the influence of a response by the *response weight*, which determines the importance of each token in a query with respect to the response. Then, the decoder of the generator processes the response weight as well as a query encoding to generate a diverse response. The effectiveness of RGRW is proven by showing that it generates more diverse and informative responses than the baseline response generator by focusing more on the tokens that are important for generating the response. Additionally, the proposed model overwhelms the Commonsense Knowledge-Aware Dialogue generation model (ConKADI), which is a state-of-the-art model.



Citation: Kim, S.-E.; Lim, Y.-S.; Park, S.-B. Strong Influence of Responses in Training Dialogue Response Generator. *Appl. Sci.* **2021**, *11*, 7415. <https://doi.org/10.3390/app11167415>

Keywords: natural language processing; chat-bot; open-domain dialogue; response generator; keyword; response weight

Academic Editor: Giancarlo Mauri

Received: 14 July 2021

Accepted: 11 August 2021

Published: 12 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advent of sequence-to-sequence models [1,2], response generation in open domain dialogues has made great progress. Nevertheless, most current response generators often make very general and unattractive responses such as “I don’t know” or “What are you talking about?” [3,4], since such responses are appropriate to any query. The same response to any query harms the reliability of dialogue-based systems, and thus it is regarded as one of the most critical problems in response generation.

Some previous studies noticed that the traditional loss functions, such as maximum likelihood, assign a high probability to *safe* responses. One solution to avoid safe responses is to inject external knowledge into response generator. Ghazvininejad et al. adopted unstructured text knowledge [5] and Wu et al. chose structured graph knowledge [6] as external knowledge. However, this approach takes a great amount of memory and searching time during inference because appropriate external knowledge to a given query should be extracted from a knowledge base.

Another approach to creating diverse responses is to define a special loss function. The benefit of this approach is that it does not require any external knowledge. Li et al. showed that the bidirectional influence between a query and its response leads to the generation of more diverse and interesting responses [7]. Thus, they proposed the maximum mutual information as a loss function to model the bidirectional influence. On the other hand, Wu et al. found out that general responses overwhelm specific ones in most dialogue

corpora and thus they are preferred by response generators [8]. As a solution to this problem, they proposed the max-marginal ranking loss to highlight the impact of less common but more relevant tokens in a query. However, since these loss functions focus only on the relation among tokens, they fail in capturing the entire context of a response.

This paper proposes a novel response generator—Response Generator with Response Weight (RGRW)—which reflects the importance of each token in a query with respect to the whole response. The proposed generator is an encoder–decoder model of which the encoder is pre-trained Bidirectional Encoder Representations from Transformers (BERT) [9] and the decoder is a pre-trained Generative Pre-Training of a language model-2 (GPT-2) [10]. In addition to them, it includes a *response weight*, which captures the importance of every query token with respect to a response [11]. Even though the BERT-encoder is trained to reflect a response, it fails in capturing whole key tokens of a query with respect to the response because it focuses on encompassing the information within a query. Thus, the response weight delivers the relevancy of each query token toward a response for the decoder, that is, it identifies the key query tokens for generating a response. Then, the decoder reflects the response weight in the response generation by paying more attention to the query tokens relevant to the generating response.

The effectiveness of the proposed response weight is proved with the short-text keyword detection task [12]. For the experiment, the MAUI twitter data set, which consists of data pairs with short sentences and keywords of sentences, is used. Even if the response weight is not a direct keyword extractor, its performance on the MAUI twitter data set is competitive against those of direct keyword extractors. Especially, it outperforms the keyword extractor trained with non-MAUI twitter data. The performance of the response generation by RGRW is verified with the Reddit data set, an open-domain singleton dialogue data set [13]. According to the experimental results, RGRW shows the highest score among the baselines in Distinct- n and word-level Entropy, which implies that RGRW generates diverse and informative responses by focusing more on the tokens that are important for generating the response. Additionally, RGRW outperforms the state-of-the-art models, such as the Commonsense Knowledge-Aware Dialogue generation model (ConKADI).

The rest of this paper is organized as follows. Section 2 explains the previous studies to generate diverse responses. Section 3 introduces the need for the response weight and how to train it. Section 4 describes the proposed model, RGRW. Section 5 reports the experimental results on the MAUI Twitter data set and Reddit data set. Finally, Section 6 draws conclusions.

2. Related Work

A number of efforts have been made to minimize the generation of general responses since the sequence-to-sequence model was introduced to response generation. Xu et al. tried to produce diverse responses by adopting a generative adversarial network (GAN) in which a discriminative classifier distinguishes machine-generated responses from human-made ones [14]. Zhou et al. defined response generation as a one-to-many mapping at the discourse level [15]. Thus, they applied a conditional variational autoencoder (CVAE) in which a latent variable captures discourse-level variations. On the other hand, Li et al. distilled dialogue data to control the response specificity [16]. They trained a sequence-to-sequence model with dialogue data to respond a query. Then, they removed the training examples from the data that are close to common responses, and then re-trained the model with the remaining data. Since their method produces multiple sequence-to-sequence models for different levels of specificity, they also trained a reinforcement learning system for choosing a model with the best specificity.

One of the main reasons for the preference for safe responses is the lack of background knowledge in a response generator. One representative approach to solve this problem is to provide extra information to a response generator [17,18]. For instance, an unstructured text was used as external knowledge for a fully data-driven neural dialogue system [5], and a knowledge graph was adopted to provide common knowledge as external

information [6,13]. Another way to provide background knowledge is to use a dialogue corpus with additional information [19]. Zhang et al. and Rashkin et al. used a corpus with personal and empathetic information, respectively, to train their conversational agents [20,21]. However, it is very expensive in memory usage and requires a much longer inference time to use additional information. On the other hand, Jiang et al. noticed that the cross-entropy loss prefers high-frequent tokens to low-frequent ones [22]. Thus, they proposed the frequency-aware cross-entropy loss to balance the number of identical tokens appearing in a response.

Some previous studies attempted to leverage the response quality by exploiting keywords in a query [23]. Xing et al. obtained topic-related keywords from a pre-trained LDA model, and increased the probability of topic-related keywords through a joint attention mechanism [24]. However, since they focused only on a query to obtain keywords, the keywords do not deliver any information residing on a response. Tang et al. extracted a keyword from a query to control the intended content of a response [25]. Their method predicts a keyword from an entire dialogue history. As a result, the direct meaning of a current query is not reflected sufficiently. In addition, since their method extracts a single keyword from a query, it often misses the whole context of the query.

3. Learning Response Weight

The response weight aims at providing a decoder with the relatedness between a response and each token in a query. Table 1 shows that it enhances the quality of a response to identify key tokens in a query under the response context and reflect them into response generation. In this table, *Q* and *R* indicate a query and its response, respectively. The bold words in the queries are the key tokens that are related highly with a response, and the underlined words are the tokens highlighted by the attention of a transformer encoder–decoder. In the first example, the encoder–decoder attention focuses only on ‘orange or gold’, but the response contains the word ‘choice’ due to the word ‘pick’ in the query. All other examples also show a similar phenomenon. The expressions of ‘try’ and ‘still waiting’ appear at the responses because ‘shot’ and ‘haven’t received’ in the queries are response-related. Therefore, it helps to generate diverse responses to identify such key tokens in a query.

Table 1. The examples which show that the response context affects response generation in the Reddit data set.

Q: <u>Orange</u> or gold, pick one R: Orange. Easy <i>choice</i>
Q: I’d love if you gave it a shot! R: All right. I’ll give it a <i>try</i> . Do you have it somewhere I can download it?
Q: I ordered march 11th and I haven’t received anything yet. R: Ordered same day, <i>still waiting</i> on mine as well.

Figure 1 shows the overall architecture for computing the response weight. The response weight is obtained while representing a query q into a query vector \mathbf{q}^a with a transformer encoder so that \mathbf{q}^a can reflect a potential response of q . The transformer encoder is trained with a bitext classification task whose goal is to predict whether a query q entails a response r . In Figure 1, the bitext classification is solved by the *bitext classifier* which is implemented as a bilinear function, that is, the bitext classifier f_{bt} determines the entailment between q and r by the following:

$$f_{bt}(\mathbf{q}^a, \mathbf{r}) = \mathbf{q}^a W_{bt} \mathbf{r}, \quad (1)$$

where \mathbf{q}^a and \mathbf{r} are vector representations of q and r , respectively, and W_{bt} is a trainable weight matrix. This classifier is trained to maximize the log-likelihood by optimizing W_{bt} and the parameters of the transformer encoder with the Adam optimizer [26].

Note that \mathbf{q}^a is encoded by the transformer encoder. Thus, the transformer encoder is trained to express \mathbf{q}^a , similar to \mathbf{r} when q and r are a conversation pair from a real corpus. On the contrary, it encodes \mathbf{q}^a differently from \mathbf{r} , when q and r are from negatively-sampled data.

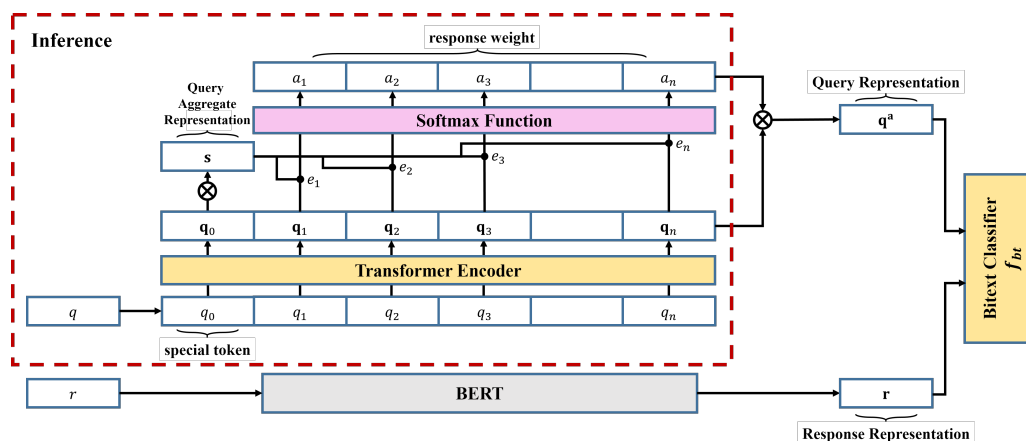


Figure 1. Overall structure for learning the response weight from a pair of a query and a response.

The representation of q into \mathbf{q}^a is done as follows. Denote $q = \langle q_1, q_2, \dots, q_n \rangle$ as a query composed of n tokens. A special token q_0 is added at the beginning of q , and it plays a similar role to the [cls] token of BERT. Thus, the output of the transformer encoder becomes $\mathbf{q} = \langle \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2 \dots, \mathbf{q}_n \rangle$, where \mathbf{q}_0 aggregates the sequence representation of the query. To obtain a richer representation, \mathbf{q}_0 is linearly transformed to \mathbf{s} as follows:

$$\mathbf{s} = W_s \mathbf{q}_0 + b_s, \tag{2}$$

where W_s and b_s are trainable parameters. Since \mathbf{s} is a summary of the query q , the importance of each query token q_i with respect to \mathbf{s} is computed by $e_i = \mathbf{s} \cdot \mathbf{q}_i$. Then, the final weight \mathbf{a} is obtained by applying the softmax to $\mathbf{e} = \langle e_1, e_2, \dots, e_n \rangle$, that is, the weight is the following:

$$\mathbf{a} = \langle a_1, a_2, \dots, a_n \rangle, \tag{3}$$

where

$$a_i = \frac{\exp e_i}{\sum_{j=1}^n \exp e_j}. \tag{4}$$

The final query representation \mathbf{q}^a is computed by a weighted sum of \mathbf{a} and \mathbf{q}_i s, that is, the following:

$$\mathbf{q}^a = \sum_{i=1}^n a_i \cdot \mathbf{q}_i. \tag{5}$$

The response r is encoded as a vector \mathbf{r} by the BERT [9] finetuned with only the responses of dialogue dataset. Then, the transformer encoder is trained to reflect the classification result of the bitext classifier.

At the inference time, the response r is not available. Thus, only the transformer encoder boxed with red dotted lines in Figure 1 is used to compute the response weight \mathbf{a} in Equation (3) from a query q . Since the transformer encoder is trained to reflect the response into encoding q , the vector \mathbf{a} is called the *response weight*.

4. Response Generation with Response Weight

The proposed RGRW has a sequence-to-sequence architecture composed of a BERT encoder and a GPT-2 decoder as shown in Figure 2. The key feature of the generator is that the decoder is given a response weight \mathbf{a} as well as the vector representation $\hat{\mathbf{q}}$ of a query \hat{q} . As a result, the generator can make a response that follows both query and response contexts.

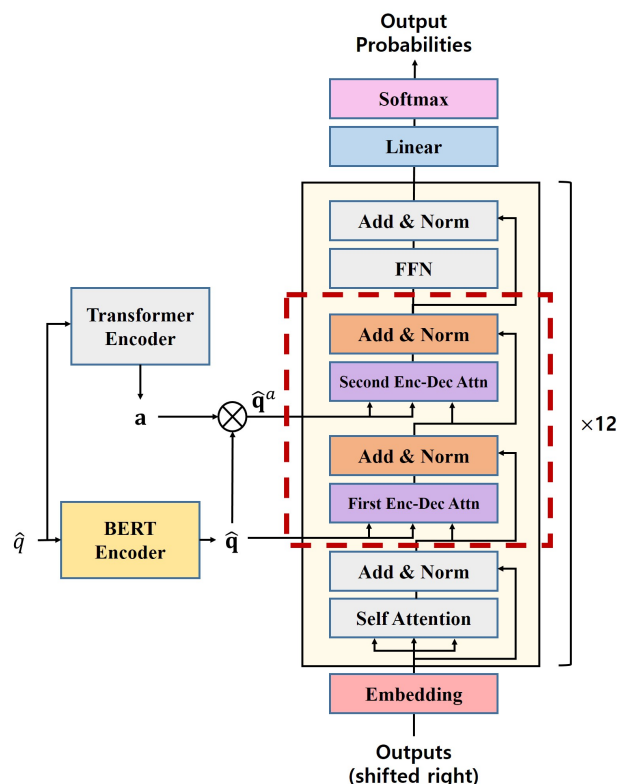


Figure 2. The structure of the proposed response generator, RGRW.

The encoder is a pre-trained BERT, and it takes a query \hat{q} as an input and then outputs its vector representation $\hat{\mathbf{q}}$. The decoder has a transformer structure, but the main difference between the decoder and the standard transformer is that it takes two kinds of inputs: $\hat{\mathbf{q}}$ and \mathbf{a} . When a decoder has multiple encoding inputs, their concatenation is often used as a single input [27,28]. However, the concatenation of $\hat{\mathbf{q}}$ and \mathbf{a} makes it difficult to grasp the key context of \hat{q} since the concatenation becomes just a lengthened representation of two similar encodings. Therefore, the proposed decoder has two individual attention layers that process $\hat{\mathbf{q}}$ and \mathbf{a} sequentially.

The first encoder–decoder attention layer uses $\hat{\mathbf{q}}$ for both the key and value and the output of the self-attention layer for query. On the other hand, the second encoder–decoder attention layer uses a scalar multiple of $\hat{\mathbf{q}}$ by an element of \mathbf{a} for both the key and value, that is, when the length of a query \hat{q} is m , the key and value of the second encoder–decoder attention layer is the following:

$$\hat{\mathbf{q}}^{\mathbf{a}} = \langle \hat{\mathbf{q}}_1^{\mathbf{a}}, \hat{\mathbf{q}}_2^{\mathbf{a}}, \dots, \hat{\mathbf{q}}_m^{\mathbf{a}} \rangle, \tag{6}$$

where $\hat{\mathbf{q}}_i^{\mathbf{a}} = a_i \cdot \hat{\mathbf{q}}_i$. The query of this layer is the output of the first encoder–decoder attention layer. Therefore, the decoder grasps the overall context of \hat{q} in the first encoder–decoder attention layer, and catches the response-related tokens of the query in the second encoder–decoder attention layer.

RGRW has twelve decoder blocks as shown in Figure 2. Because the proposed structure is partially the same as that of GPT-2 [10], the parameters of the pre-trained GPT-2 are

borrowed to avoid excessive resource consumption. That is, only the two encoder–decoder attention layers boxed with red dotted lines in the figure are optimized during the training time, while other layers are all fixed. The AdamW optimizer [29] is used for optimizing the response generator with the cross entropy loss.

5. Experiments

5.1. Experimental Settings

In order to train the proposed response weight and bitext classifier, the pairs of a query and a response are needed. Dailydialog [30] is used for this purpose, as this data set contains open-domain multi-turn dialogue pairs. Since RGRW does not target multi-turn dialogues, the dialogue pairs in Dailydialog are converted into single-turn dialogue pairs by regarding the odd-numbered utterances and the even-numbered utterances as queries and responses, respectively.

The performance of the response weight is verified through *keyword detection*. For this task, the MAUI Twitter data set [31] is used, since the tweets are relatively short and the keywords are labeled at the tweets of this set by crowd-sourcing. The average tweet length is 78.55, and the average number of keywords per tweet is 3.92. Even if this data set has its own training and validation sets, only the test set is used in the experiments since the proposed response weight and bitext classifier are trained with Dailydialog.

The Reddit data set [13] is adopted for the experiments of the response generator. The data set also consists of open-domain dialogues, but the dialogues are single-turn pairs. The queries and the responses that are shorter than four words or longer than 20 words are excluded from the data set following the study of Wu et al. [6]. Table 2 summarizes the statistics on the data sets after pre-processing.

Table 2. Simple statistics on the data sets for response weight and response generation.

	# Train	# Valid	# Test
Dailydialog	45,337	3851	-
MAUI Twitter	-	-	500
Reddit	1,352,961	40,000	40,000

The hyper-parameters for the transformer encoder of the response weight are equivalent to those of the transformer base model. The dimension of embedding vector is 512, and that of the inner-layers of feed-forward networks is 2048. The number of heads in multi-head attention is eight, and the number of transformer encoder layers is six. The batch size of training and validation sets is 32, and the learning rate is 0.0001. In addition, label smoothing [32] of $\epsilon_{ls} = 0.1$ is used.

The encoder of the response generator is the pre-trained BERT base model, and the transformer decoder setting for response generation is similar to GPT-2. The embedding vector dimension is 768, and the inner-layers of feed-forward networks have 3072 dimensions. The number of heads in the multi-head attention is 12, and the number of transformer encoder layers is six. The batch size of training and validation sets is 32, and the learning rate is 6.25×10^{-5} .

5.2. Response Weight

Since the response weight aims at finding the importance of each query token, its performance is verified with a *keyword detection task*. The proposed response weight is compared with five variations of MAUI [31,33], a strong keyword detection system trained with various features. MAUI-df is a default MAUI model trained with a decision tree and tf-idf, while MAUI-wv and MAUI-br use the structured skip-n-gram [34] and the Brown cluster feature [35] as well as the features of MAUI-df. MAUI-brwv uses all features stated above, and MAUI-out has the same structure as MAUI-brwv but is trained with the news articles used in the work of Marujo et al. [36]. Note that MAUI-out is the only variant that is not trained with the MAUI Twitter data.

Table 3 summarizes the evaluation results. RW in this table is the proposed response weight. Since the proposed response weight is not a direct keyword detector, the words in a query are chosen as keywords when their a_i in Equation (4) is greater than 0.3. All performances are measured for four extracted keywords. While MAUI-brwv shows the highest performance on F1-score, the proposed model ranks third. Note that RW is trained with Dailydialog, not with MAUI twitter data. Thus, it is difficult to compare the performance of RW directly with those of MAUI variants. Nevertheless, it outperforms MAUI-df, MAUI-wv, and MAUI-out. In particular, the fact that RW achieves higher performance than MAUI-out, another model trained with non-MAUI twitter data, proves the effectiveness of RW to identify keywords from a query.

Table 3. Precision, recall and F1-score on MAUI Twitter data set. Scores in bold stand for the leadership among the models

Model	P	R	F1
MAUI-df	53.97	53.15	53.56
MAUI-wv	55.80	54.45	55.12
MAUI-br	71.95	75.01	73.45
MAUI-brwv	72.05	75.16	73.57
MAUI-out	55.54	48.74	51.92
RW	55.75	55.57	55.66

Figure 3 depicts the change of the response weight a during its training when the query is “Orange or gold, pick one” and the response is “Orange. Easy choice.” RW-1 and RW-10 in this figure are the response weights after one epoch and ten epochs, respectively, and Enc-Dec is the average attention of all heads in the encoder–decoder attention layer of a standard transformer. When the training of the proposed response weight is at one epoch, the meaningless or general tokens in the query such as ‘or’ and ‘,’ have high weight. This is because neither the query nor the response is reflected enough to the weight. On the other hand, ‘orange’ and ‘pick’ have high attention after 10 epochs, since they are deeply related to ‘orange’ and ‘choice’ in the response.

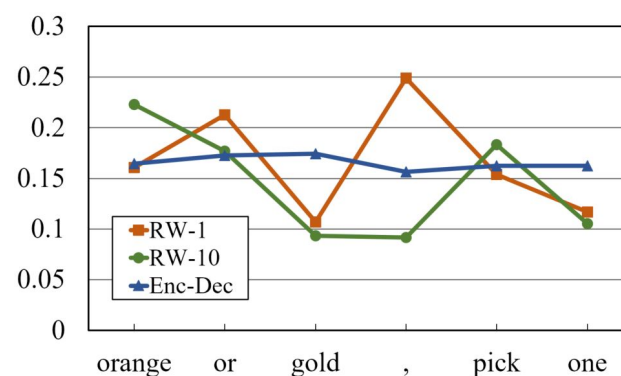


Figure 3. The distributions of the attentions for an example query and response. The distributions are obtained from the proposed response weight and the standard transformer.

The most important finding in this figure is that RW-10 shows a different weight distribution from Enc-Dec. Enc-Dec pays high attention to ‘gold’, ‘pick’, and ‘one’. Though the transformer reflects responses during its training, Enc-Dec is affected much more by the self-attention of the query. Thus, the word like ‘one’ that is not strongly related with the response is also spotlighted. In addition, the attention difference in Enc-Dec is insignificant. To sum up all these results, better responses can be generated by enhancing the influence of the response through the response weight.

5.3. Response Generator

The performance of RGRW is compared with those of five baselines, which are TS2S, COPY, GenDS, CCM and ConKADI. TS2S is a transformer with six blocks of an encoder and a decoder. COPY is an LSTM-based sequence-to-sequence model with the copy mechanism [37] and GenDS generates responses from the candidate facts retrieved from a knowledge base [38]. CCM [13] and ConKADI [6] are commonsense knowledge-aware response generators. The responses in the training set are used as posterior knowledge when training ConKADI, but they are not used in CCM.

The performance of the models is measured with eight automatic metrics. Emb_{avg} [39] is the similarity between the average embedding vector of a ground-truth response and that of the generated response, while Emb_{ex} measures the similarity between embedding vectors using vector extrema. Bleu-2 and Bleu-3 are the ratio of bi-gram and tri-gram overlaps, respectively [40], and Dist-1 and Dist-2 are the ratio of distinct uni-grams and bi-grams in all generated responses [7]. Entropy is the average word-level entropy [41]. The R_q is the relative score of a model when the arithmetic mean of metric scores of all other comparison models is set to 1.00 [6].

The results on these metrics are shown at Table 4. RGRW achieves the highest scores for all metrics, except Bleu and Emb_{avg} . For both Bleu-2 and Bleu-3, RGRW shows the second lowest performance. The generated responses are often acceptable, even if they are different from the ground-truth responses. Thus, the sole measurement with Bleu has a limit to evaluate the quality of responses.

Table 4. The empirical comparison of RGRW against its baselines. Scores in bold stand for the leadership among the response generators.

Metric	Embedding		Overlap		Diversity		Inform	R
	Emb_{avg}	Emb_{ex}	Bleu-2	Bleu-3	Dist-1	Dist-2	Entropy	R_q
TS2S	0.764	0.845	1.50	0.44	1.47	14.17	8.47	0.87
COPY	0.868	0.841	5.43	2.26	1.73	8.33	7.87	1.09
GenDS	0.876	0.851	4.68	1.79	0.74	3.97	7.73	0.89
CCM	0.871	0.841	5.18	2.01	1.05	5.29	7.73	0.96
ConKADI	0.867	0.852	3.53	1.27	2.77	18.78	8.10	1.19
RGRW	0.794	0.870	1.70	0.49	3.24	25.04	11.42	1.26

The embedding metric is one of the metrics that solve the limitation of Bleu. It compares the contexts of the generated responses and the ground-truth responses. GenDS shows the highest performance in Emb_{avg} , while RGRW achieves the best performance in Emb_{ex} . In general, the embedding vectors of general words are located close to the origin and the vectors of contextually important words are far from the origin. Thus, Emb_{ex} focuses more on contextually important words than general words [39]. In other words, the responses generated by RGRW are more similar to the key topics of ground-truth responses than those by other baselines.

Dist-1, Dist-2, and entropy measure how diverse and informative the generated responses are. RGRW shows the best performance for these metrics. One thing to note is that the higher the Bleu score that a model shows, the lower the diversity and informative scores that it achieves. This is because focusing on lexical coincidence with the ground-truth response affects the generation of diverse responses negatively. Finally, RGRW outperforms all baselines in R_q . In particular, it shows 0.07 higher R_q than ConKADI, the best baseline.

Summarizing all the results, RGRW generates a response that is not only lexically similar, but also contextually similar to a ground-truth response. In addition, RGRW generates more diverse and informative responses by adopting the response weight. All these results prove that the response generation through response weight is effective in producing diverse and context-preserving responses.

The effectiveness of the structure with two attention layers is shown by an ablation study with the encoder–decoder attention (EDA) and the response weight (RW). Table 5 shows the results of the ablation study on the Reddit data set. Without EDA, RGRW is influenced only by RW. As a result, the Dist-2 and entropy of ‘- EDA’ are rather higher than those of RGRW, but its Bleu-2 and R_a become lower than those of RGRW. This is because the information in a query is not delivered enough to the decoder. Without RW, RGRW is equivalent to a sequence-to-sequence model with a BERT encoder and a GPT-2 decoder. Since the decoder of ‘- RW’ does not receive the response weight, Dist-2 and the entropy of the model are much lower than those of RGRW. One thing to note in this table is that RGRW shows higher R_a than both ‘- EDA’ and ‘- RW’, which implies that the response weight does not contribute only to the generation of diverse responses, but also helps generate the responses similar to the ground-truth ones.

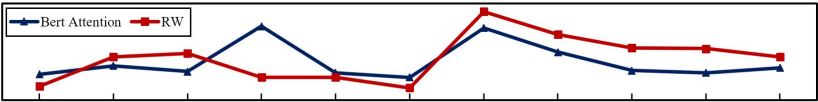
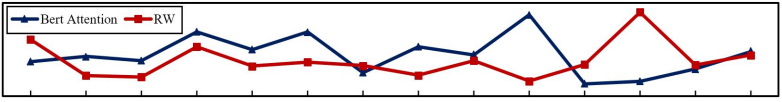
Table 5. Ablation study on two attention layers. Scores in bold are the highest score for each metric.

Model	Bleu-2	Dist-2	Entropy	R_a
RGRW	1.70	25.04	11.42	1.26
- EDA	1.59	25.60	11.45	1.23
- RW	1.52	20.45	8.96	1.08

5.4. Case Study

Table 6 shows two examples that compare the responses generated with the response weight and those without the attention. If the response weight is removed from RGRW, RGRW becomes equivalent to a sequence-to-sequence model with a BERT-encoder and a GPT-2 decoder. Thus, the responses in ‘Without RW’ in this table are the outputs of the GPT-2 decoder, while those in ‘With RW’ are the outputs of RGRW. In each example, the attention by the BERT-encoder (blue line) and the proposed response weight (red line) are depicted as graphs. In these graphs, the attention weights are expressed per a word unit for easy interpretation, where the attention weight of a word is obtained as an average of the attention weights for the subwords of the word. Note that the values for the response weight are $a_i s$ in Equation (4).

Table 6. The examples that show the effectiveness of the response weight in RGRW.

Query & Attentions	
Golden Response	All right. I'll give it a try. Do you have it somewhere I can download it?
W/o RW	It would be nice to have them.
W/RW	I'll give it a shot and see what happen!
Query & Attentions	
Golden Response	Too bad. They taste awful.
W/o RW	I think it's a good idea.
W/RW	Oh yeah! I've always wanted to try them! They taste great!

According to the table, 'With RW' generates responses more similar to the ground-truth ones and less general than the 'Without RW' ones. In the first example, 'love' and 'gave' have high BERT attention weights. As a result, the GPT-2 decoder generates the word 'have' in the response of 'Without RW', and the response is semantically unrelated with the ground-truth one. On the other hand, all words in the phrase of 'gave it a shot' have high a_i values. Since RGRW pays high response weight to 'gave it a shot', the response in 'With RW' follows the ground-truth one more semantically.

In the second example, 'them' and 'salt' are stressed on in the BERT attention, which leads to the generation of a very general answer in the response of 'Without RW'. On the other hand, RGRW focuses on the words 'cucumber', 'them', and 'munch' with high response weight. In particular, the focus on 'munch' results in the generation of 'try' and 'taste' in the response of 'With RW'. In addition, the generated answer is much less general and semantically similar to the ground-truth response, though the sentiment polarity of the answer is opposite to that of the ground-truth one.

6. Conclusions and Future Work

This paper proposes RGRW, a novel response generator in which the effect of a potential response is reflected strongly through *response weight*. By adopting the response weight, RGRW is able to reduce generating safe responses and make a response in accordance with a query. In order to obtain an optimal response weight, the bitext classifier is trained to distinguish whether a pair of a query and a response is real or not. Training of the bitext classifier leads to adaptation of the response weight and the transformer encoder to a response. As a result, the response weight is able to reflect a potential response into a query attention.

The proposed generator, RGRW consists of a BERT-encoder and a GPT-2-like decoder, where the decoder has two additional encoder–decoder attention layers to GPT-2. The first attention layer processes the overall context of a query given by the BERT-encoder, and the second attention layer catches the response-related tokens of the query using the response weight. To avoid excessive resource consumption, the parameters of the layers equivalent to GPT-2 are borrowed from the original GPT-2 and fixed, while those of the two attention layers are optimized newly.

The proposed response weight was verified through the short-text keyword detection on the MAUI Twitter data. Even if the response weight is trained on Dailydialog data, it shows competitive keyword extraction performance on the MAUI Twitter data. In particular, it outperforms other keyword extractors trained with non-MAUI data. It was also shown empirically for RGRW to generate more diverse and informative responses than the current state-of-the-art methods on the Reddit data set. RGRW achieves the best Emb_{ex} according to the experimental results, which implies that the responses of RGRW are semantically similar to ground-truth ones. In addition, through an ablation study, the effectiveness of the proposed structure of RGRW with two attention layers was proved. Unlike ConKADI, RGRW does not use external knowledge, so it does not require knowledge retrieving time during inference. Therefore, the inference speed is relatively faster than ConKADI. In addition, it takes less training time than a general transformer sequence-to-sequence model because it does not train all parameters of the model.

The RGRW approach has certain limitations. RGRW freezes some layers for efficient transfer learning when fine-tuning. Recently, a transfer learning method in which only low-level filters are transferred and frozen was proposed in the image generation task [42]. This study is based on research showing that low-level filters capture generality well. On the other hand, RGRW performs the transfer and freezing of all blocks equally without understanding the characteristics of each decoder block. Therefore, as part of future work, we will study how to differentiate the transfer learning by understanding the role of each decoder block.

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

Funding: This work was supported by the National Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2016R1D1A1B04935 67816) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform).

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

References

1. Sutskever, I.; Vinyals, O.; Le, Q. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 3104–3112.
2. Vinyals, O.; Le, Q. A Neural Conversational Model. *arXiv* **2015**, arXiv:1506.05869.
3. Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.Y.; Gao, J.; Dolan, B. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015; pp. 196–201.
4. Serban, I.V.; Sordoni, A.; Lowe, R.; Charlin, L.; Pineau, J.; Courville, A.; Bengio, Y. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3295–3301.
5. Ghazvininejad, M.; Brockett, C.; Chang, M.; Dolan, B.; Gao, J.; tau Yih, W.; Galley, M. A Knowledge-Grounded Neural Conversation Model. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
6. Wu, S.; Li, Y.; Zhang, D.; Zhou, Y.; Wu, Z. Diverse and Informative Dialogue Generation with Context-Specific Commonsense Knowledge Awareness. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; pp. 5811–5820.
7. Li, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 110–119.
8. Wu, B.; Jiang, N.; Gao, Z.; Li, M.; Wang, Z.; Li, S.; Feng, Q.; Rong, W.; Wang, B. Why Do Neural Response Generation Models Prefer Universal Replies? *arXiv* **2018**, arXiv:1808.09187.
9. 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; pp. 4171–4186.
10. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models Are Unsupervised Multitask Learners*; Technical Report; OpenAI: San Francisco, CA, USA, 2019.
11. Onan, A. Topic-enriched Word Embeddings for Sarcasm Identification. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 24–27 April 2019; pp. 293–304.
12. Onan, A.; Korukoğlu, S.; Bulut, H. Ensemble of Keyword Extraction Methods and Classifiers in Text Classification. *Expert Syst. Appl.* **2016**, *57*, 232–247. [[CrossRef](#)]
13. Zhou, H.; Young, T.; Huang, M.; Zhao, H.; Xu, J.; Zhu, X. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4623–4629.
14. Xu, Z.; Liu, B.; Wang, B.; Sun, C.; Wang, X.; Wang, Z.; Qi, C. Neural Response Generation via GAN with an Approximate Embedding Layer. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 617–626.
15. Zhao, T.; Zhao, R.; Eskenazi, M. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 654–664.
16. Li, J.; Monroe, W.; Jurafsky, D. Data Distillation for Controlling Specificity in Dialogue Generation. *arXiv* **2017**, arXiv:1702.06703.
17. Young, T.; Cambria, E.; Chaturvedi, I.; Zhou, H.; Biswas, S.; Huang, M. Augmenting End-to-End Dialogue Systems With Commonsense Knowledge. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 4970–4977.
18. Liu, Z.; Niu, Z.Y.; Wu, H.; Wang, H. Knowledge Aware Conversation Generation with Explainable Reasoning over Augmented Graphs. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 1782–1792.

19. Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; Weston, J. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv* **2018**, arXiv:1811.01241.
20. Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; Weston, J. Personalizing Dialogue Agents: I have a dog, do you have pets too? In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 2204–2213.
21. Rashkin, H.; Smith, E.; Li, M.; Boureau, Y.L. Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5370–5381.
22. Jiang, S.; Ren, P.; Monz, C.; Rijke, M. Improving Neural Response Diversity with Frequency-Aware Cross-Entropy Loss. In Proceedings of the Web Conference 2019, San Francisco, CA, USA, 15 May 2019; pp. 2879–2885.
23. Onan, A.; Toçoğlu, M.A. A Term Weighted Neural Language Model and Stacked Bidirectional LSTM based Framework for Sarcasm Identification. *IEEE Access* **2021**, *9*, 7701–7722. [[CrossRef](#)]
24. Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; Ma, W.Y. Topic Aware Neural Response Generation. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3351–3357.
25. Tang, J.; Zhao, T.; Xiong, C.; Liang, X.; Xing, E.; Hu, Z. Target-Guided Open-Domain Conversation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August; pp. 5624–5634.
26. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
27. Pal, S.; Herbig, N.; Krüger, A.; van Genabith, J. A Transformer-Based Multi-Source Automatic Post-Editing System. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Brussels, Belgium, 31 October–1 November 2018; pp. 827–835.
28. Zhang, M.; Wang, X.; Fang, F.; Li, H.; Yamagishi, J. Joint Training Framework for Text-to-Speech and Voice Conversion Using Multi-Source Tacotron and WaveNet. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 1298–1302.
29. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
30. Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; Niu, S. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; pp. 986–995.
31. Marujo, L.; Ling, W.; Trancoso, I.; Dyer, C.; Black, A.; Gershman, A.; Martins de Matos, D.; Neto, J.; Carbonell, J. Automatic Keyword Extraction on Twitter. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 637–643.
32. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
33. Medelyan, O.; Perrone, V.; Witten, I. Subject Metadata Support Powered by Maui. In Proceedings of the 10th Annual Joint Conference on Digital Libraries, Gold Coast, Australia, 21–25 June 2010; pp. 407–408.
34. Ling, W.; Dyer, C.; Black, A.; Trancoso, I. Two/Too Simple Adaptations of Word2Vec for Syntax Problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1299–1304.
35. Owoputi, O.; O'Connor, B.; Dyer, C.; Gimpel, K.; Schneider, N.; Smith, N. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 380–390.
36. Marujo, L.; Gershman, A.; Carbonell, J.; Frederking, R.; Neto, J.P. Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. In Proceedings of the Eighth International Conference on Language Resources and Evaluation, Istanbul, Turkey, 21–27 May 2012; pp. 399–403.
37. Gu, J.; Lu, Z.; Li, H.; Li, V. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 1631–1640.
38. Zhu, W.; Mo, K.; Zhang, Y.; Zhu, Z.; Peng, X.; Yang, Q. Flexible End-to-End Dialogue System for Knowledge Grounded Conversation. *arXiv* **2017**, arXiv:1709.04264.
39. Liu, C.W.; Lowe, R.; Serban, I.V.; Noseworthy, M.; Charlin, L.; Pineau, J. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2122–2132.
40. Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; Jurafsky, D. Adversarial Learning for Neural Dialogue Generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2157–2169.

41. Mou, L.; Song, Y.; Yan, R.; Li, G.; Zhang, L.; Jin, Z. Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 3349–3358.
42. Zhao, M.; Cong, Y.; Carin, L. On Leveraging Pretrained GANs for Generation with Limited Data. In Proceedings of the Thirty-Seventh International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020.