Knowledge-Grounded Dialogue Generation with Term-level De-noising

Wen Zheng[1], **Natasa Milic-Frayling**[1], **Ke Zhou**[1,2]
[1]University of Nottingham
[2]Nokia Bell Labs
{wen.zheng,ke.zhou}@nottingham.ac.uk natasa-milic@frayling.net

Abstract

Dialogue generation has been improved through injecting knowledge into generative models. However, addition of knowledge through simple selection of sentences or paragraphs is likely to introduce noise and diminish the effectiveness of the generative models. In this paper, we present a novel **K**nowledge **T**erm **W**eighting **M**odel (KTWM) that incorporates term-level de-noising of the selected knowledge. KTWM includes a module for generating Simulated Response Vectors (SRVs) and uses SRVs attention distributions with the knowledge embeddings to determine knowledge term weights. Our experiments demonstrate that KTWM, combined with various knowledge selection algorithms, consistently achieves statistically significant improvements over methods without term weighting when applied to two publicly available datasets Wizard of Wikipedia (Wiz) and Holl-E. The results are particularly improved for the Wiz test data with unseen topics, demonstrating the robustness of the KTWM noise-reduction approach.

1 Introduction

Research in dialogue generation has rapidly evolved from sequence-to-sequence (Sutskever et al., 2014) and Transformer models (Vaswani et al., 2017) to approaches with pre-trained models such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and T5 (Raffel et al., 2020). More recently, it included techniques that use knowledge, in addition to the original posts, to improve the quality of the generated responses (Ghazvininejad et al. (2018), Moghe et al. (2018), Dinan et al. (2019), Galley et al. (2019), Lian et al. (2019), Zheng and Zhou (2019), Zhao et al. (2020a), Zhao et al. (2020b)).[1] This approach is referred to as *knowledge-grounded dialogue generation* and is the primary concern of this paper.

Post: I am a big fan of education. I think people don't realise how important it is.
Ground-truth response: Sure, education is important since it facilitates learning and the acquisition of skills.
Knowledge terms weighted by KTWM:
Education is the process of facilitating learning , or the acquisition of knowledge , skills , values , beliefs , and habits
Response generated by KTWM: I agree. Education is a great way to learn about facilitating learning.
0 0.2 0.4 0.6 0.8 1

Table 1: Example of a post, ground truth response, injected knowledge and generated response by the Knowledge Term Weighting Model (KTWM). The term highlights indicate the predicted probability of a term being useful.

In particular, we consider the key issue of effectively incorporating the selected knowledge into the generation process. For example, Weston et al. (2018) apply a *retrieve and refine* method to expand the post with the retrieved knowledge and then use it in the generation process. Lian et al. (2019) consider the post and response *posterior distributions* and the post *prior distribution* to train jointly the model for knowledge selection and response generation. Kim et al. (2020) view the knowledge selection as a sequential decision problem, first selecting the best ranked knowledge using a sequential latent variable model, and then generating a response based on selected knowledge.

To the best of our knowledge, all prior approaches focus on the selection and injection of knowledge at the sentence or paragraph level. However, that makes it hard to control for potential

[1]Previous works used a variety of terms to refer to a post such as 'question', 'utterance','source' and 'query'. Similarly for response they used 'answer', 'response', and 'target'. In this paper we call the first role the 'post' and the second role the 'response' and we aim to generate the response for the given post.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2972–2983

noise, i.e., for inclusion of non-relevant words, and previous studies (Galley et al. (2019), Zheng et al. (2020)) have shown that adding noise can decrease the response generation quality. Therefore, it is important to investigate whether and how we can adjust the contributions of terms in the selected knowledge. Prior research has not considered that issue systematically.

Our paper fills this gap by introducing a novel *Knowledge Term Weighting Model* (KTWM) for dialogue generation, which effectively estimates term weights of the injected knowledge and incorporates such weights into the response generation. The response generation thus benefits from such nuanced term-level knowledge weighting, promoting important knowledge terms rather than treating equally all the terms in the selected sentences. In Table 1 we show an example of the KTWM term weighting and its generated response: the terms 'education', 'is', 'facilitating' and 'learning' are given higher weights correctly as they do appear in the ground-truth response, while the words 'values' and 'beliefs' are correctly assigned lower scores.

We conducted an extensive range of experiments with KTWM on two publicly available datasets: Wiz (with seen and unseen test topics) (Dinan et al., 2019) and Holl-E (Moghe et al., 2018). KTWM performs consistently well with different selections of knowledge, specifically with Post-KS (Lian et al., 2019), SKT (Sequential Latent-Knowledge Selection) (Kim et al., 2020) and TED (Transformer with Expanded Decoder) (Zheng and Zhou, 2019). Our work achieves both a superior performance in knowledge-grounded dialogue generation and new insights into the impact of the knowledge term weighting on that performance. The code of our method is publicly available at *https://github.com/tonywenuon/acl2021_ktwm* and enables reproducibility of our results.

2 Related Work

The knowledge-grounded dialogue generation can be tackled by decomposing it into two subproblems: (1) selecting knowledge from a large pool of candidates (*knowledge selection*), and (2) generating a response from the selected knowledge and context (*knowledge-grounded response generation*).

Knowledge-grounded Response Generation Ever since the knowledge-based dialogue generation task was released by DSTC-7 (Galley et al., 2019), research interest in the topic has been steadily growing. Ghazvininejad et al. (2018) proposed a multi-task learning approach to produce responses. The posts and knowledge are used in the encoders and share the same decoder parameters. Luan et al. (2017) expanded the scope and introduced personality information into the model. They assumed that the trainable parameters can potentially capture persona from the non-conversational data (Tweets). Yavuz et al. (2019) adopted pointer-generator networks within a hierarchical framework that enabled them to include external knowledge in addition to the context. Ye et al. (2020) proposed a latent variable based generative model, which contains a joint attention mechanism conditioned on both context and external knowledge. Li et al. (2019) applied a deliberation network to create a two-stage generative model that combines both context and knowledge and, in the second generation stage, makes use of the outputs from the first stage. Zheng and Zhou (2019) proposed Transformer with Expanded Decoder (TED) architecture that assigns different weights to different knowledge sources and incorporates them into the generation process.

While the above approaches and models focus on incorporating knowledge and context to generate responses, they do that at the sentence or paragraph level. Our work deals with the quality of the incorporated knowledge at the term level, weighing all the individual knowledge terms when generating the responses.

Knowledge Selection Considering the mechanisms for response generation, Weston et al. (2018) proposed to retrieve candidate content from a knowledge set and use it to expand the post. The result is a truncated sequence that represents a refined post. (Lian et al., 2019) select the knowledge by approximating the prior distribution (i.e., $p(knowledge|post)$) with the posterior-distribution (i.e., $p(knowledge|post, response)$) and then inject it into the decoder. Kim et al. (2020) trained a knowledge selection module and a response generation module jointly, but treated the knowledge selection as a sequential decision problem, using input and knowledge from the previous turns to select the knowledge in the subsequent turns. Zheng et al. (2020) separated the knowledge selection process from the generation process so that all the downstream generation tasks can use

the selected knowledge. They mapped posts to the best knowledge representations in both the training and the testing phase, and used the learned models to rank new post-knowledge pairs.

In this context, KTWM can be viewed as an optimization step following the knowledge selection. It is focused on learning knowledge term weights to distinguish between relevant and non-relevant terms and weighing higher those that are useful for the response generation.

3 Method

In this section we introduce the basic concepts and describe in detail our method KTWM for term-weighting of the injected knowledge. We assume that for a collection of posts P and responses R, we have a collection $\{K_{pr}\}$ of knowledge sets with sentences relevant to the specific post-response pair (p, r). For a given pair (p, r) we consider a knowledge injection process that involves three stages: (1) knowledge selection, (2) knowledge term-weighting, and (3) decoding with the weighted knowledge terms. Our primary focus is on (2), i.e., the effectiveness of the term-weighting for the knowledge incorporated in the KTWM. Thus we provide a detailed description of the term-weighting model (Figure 1) and the use of the KTWM decoder (Figure 2).

3.1 Knowledge Selection and Representation

We represent each post p, response r, and a knowledge sentence k as a vector of terms. The set K_{pr} typically contains multiple knowledge sentences and we use BM25 retrieval method to rank the sentences by their relevance to the post (in the test phase) or response (in the training phase). For knowledge injection we take the top ranked sentence. When the knowledge injection requires a specific number of terms to be used, we include additional sentences from the ranked list to meet that requirement (used in §4.4.3).

When a knowledge sentence k is retrieved based on a response r as a query, we define a ground truth vector GT_{know} for the knowledge k with the weight of 1 assigned to the knowledge terms that are present in r and the weight of 0 assigned to those that are not, i.e., $GT_{know} = (e_1, e_2, \ldots, e_l)$, where $e_i \in \{0, 1\}, i = 1, \ldots, l$.

Encoders. We adopt Transformer (Vaswani et al., 2017) as the backbone framework for the training and testing of KTWM. Transformer encoder consists of a self-attention layer and a transition layer involving the layer normalisation and residual network. Formally, the attention is defined as

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_m}}\right)V, \quad (1)$$

where Q, K, and V are embedding matrices and d_m is the embedding dimension of the model. First we compute the dot similarity of the Q and K and then apply the weighted summation with V. The representation of Q is updated with the information from K and V. If Q, K, V originate from the same source, e.g., an input post, the attention is referred to as self-attention. Otherwise, if they originate from different sources, e.g., Q relates to the decoding token and K and V are from a post, the attention turns to be a mutual-attention operation.

Figure 1 shows transformer encoders (*encoders* for short) used for the post, knowledge, and response representations and processing. We use w to designate an original term and $\widehat{w}$ to designate the term's representation. In Figure 1, n, m, and l are three pre-defined hyper-parameters which refer to the length of the post (p), response (r), and knowledge (k), respectively (e.g. w_{pi} means the i-th term of the post). Any sequence that is longer or shorter than the given length will be truncated or padded to the given length. By applying the encoder

$$V_{post} = \textit{Encoder}(w_i)(i \in [1, n]) \quad (2)$$

we obtain the post terms representations V_{post}, comprising $\widehat{w}_{p1}, \widehat{w}_{p2}, \ldots, \widehat{w}_{pn}$ (in Figure 1), from the original terms $w_{p1}, w_{p2}, \ldots, w_{pn}$. Similarly to V_{post} in Eq. (2), we obtain V_{know} and V_{resp} as term representations of the corresponding knowledge and the response, respectively.

3.2 Knowledge Term Weighting

The fundamental premise of our approach is that knowledge terms related to or present in the response should be more effective in improving dialogue generation. Thus, it can be beneficial to use methods such as attention distribution of response and knowledge embeddings to determine the weights of individual knowledge terms. However, in the real setting and during the test phase, we can only use terms and knowledge related to the post. Furthermore, the post embeddings can significantly differ from the response ones. Thus, assigning weights to the knowledge terms based on

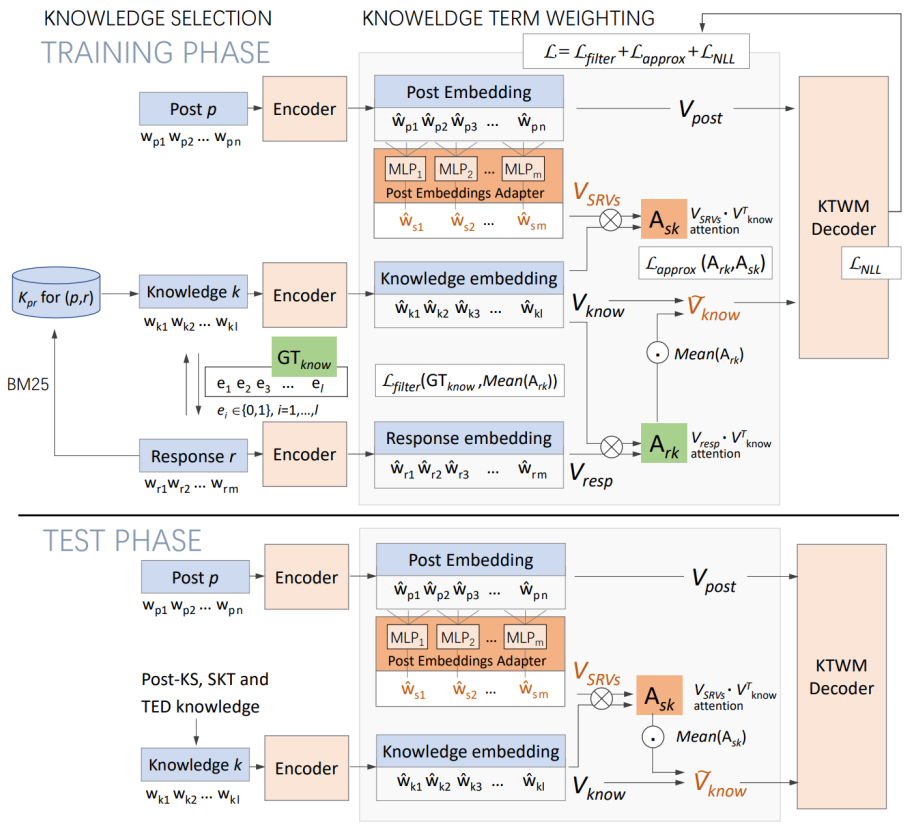

Figure 1: Architecture of the Knowledge Term Weighting Model (KTWM) showing the operations in the training and test phase. $\otimes$ designates matrix multiplication; $\odot$ designates element-wise multiplication.

their similarity to post embeddings is unlikely to be sufficient (Xing et al., 2018).

For that reason, we aim to learn how to transform the post embeddings to be effective in knowledge term weighting. We achieve that by training a *Post Embeddings Adapter* that can, for a new post, generate *Simulated Response Vectors* (SRVs) and use them in place of the response vectors to score post related knowledge terms.

To that effect, we introduce a set of Multi-Layer Perceptrons (MLPs):

$$MLP = \sum_{i=1}^{n} \widehat{w}_{pi} W_i + b \tag{3}$$

$$\widehat{w}_{sj} = MLP_j(\widehat{w}_{p1}, \widehat{w}_{p2}, \ldots, \widehat{w}_{pn})(j \in [1, m]) \tag{4}$$

where W_i and b are trainable parameters for each term p_i of the post p; $\widehat{w}_{sj}$ is the representation of the j-th term of the simulated response vector (SRV). The number of MLPs is the same as the number of terms in a given response.

During the training phase, MLPs learn the transformation of the post embeddings into SRVs that captures the ground truth response representation for a given post p. SRVs are then used to assign appropriate weights to the knowledge terms when response information is not available.

SRVs Approximation and Training. The training phase begins with V_{post}, V_{know}, V_{resp} and randomly initiated parameters of MLPs to produce the initial set of V_{SRVs} for a given post. Each iteration then involves comparison of (a) the response embeddings V_{resp} and knowledge embeddings V_{know}, and (b) SRVs with the knowledge embeddings V_{know}. More precisely, we compute the term-wise attention distributions A_{rk} and A_{sk}:

$$A_{rk} = sigmoid(V_{resp} V_{know}^T) \tag{5}$$

$$A_{sk} = sigmoid(V_{SRVs} V_{know}^T) \tag{6}$$

where $A_{rk} \in \mathbb{R}^{m \times l}$ and $A_{sk} \in \mathbb{R}^{m \times l}$; m and l are hyper-parameters that are the maximum length of the response and knowledge sentence.

A_{rk} reflects the relationship between the response terms and the knowledge terms: for each response term, A_{rk} includes attention scores with all knowledge terms. Similarly, A_{sk} includes attention scores between SRVs and the knowledge representations. The knowledge terms with larger

response-knowledge attention scores are expected to produce output closer to the true response. In the training phase, that is guided by the *filtering loss* for A_{rk}:

$$\mathcal{L}_{filter} = BCE(GT_{know}, Mean(A_{rk})) \quad (7)$$

where GT_{know} is the *knowledge ground truth* vector which indicates whether the knowledge terms appear in the corresponding response or not and *BCE* is the Binary Cross Entropy loss function. $Mean(\cdot)$ computes the mean values for knowledge terms (in the matrix columns) across response terms ($Mean(\cdot) \in \mathbb{R}^l$).

At the same time we aim to train MLPs to create SRVs similar to the response representations V_{resp}. In each iteration we compute and compare A_{sk} to A_{rk} and apply the *approximation loss* function:

$$\mathcal{L}_{approx} = MSE(Mean(A_{rk}), Mean(A_{sk})) \quad (8)$$

where $MSE(\cdot)$ is the Mean Squared Error function. $Mean(\cdot)$ of A_{rk} and A_{sk} produces l-length knowledge term vectors whose values are used to characterise the importance of each knowledge term. We use these weights to update the knowledge vector:

$$\widetilde{V}_{know} = Mean(A_k) \odot V_{know} \quad (9)$$

where $\odot$ denotes element-wise multiplication and A_k corresponds to A_{rk} in the training phase and to A_{sk} in the test phase. V_{post} and the weighted knowledge vector $\widetilde{V}_{know}$ become input for the KTWM decoder.

3.3 KTWM Decoder

In order to incorporate multiple sources of input, we adopt a decoder design that is similar to the TED model by Zheng and Zhou (2019). Figure 2 shows the architecture of our KTWM decoder. The blue frames are the standard Transformer decoder set-up

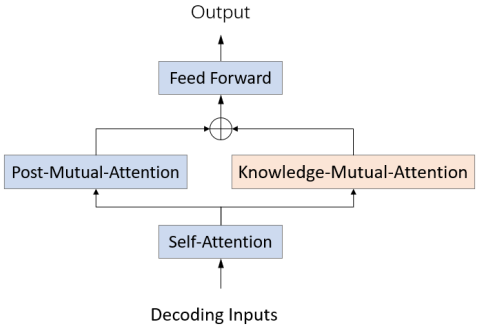

Figure 2: Knowledge Term Weighting Model Decoder.

with a self-attention layer and a mutual-attention layer (for the post), followed by a feed-forward layer.

KTWM includes an additional knowledge-mutual-attention layer which applies the same process to the knowledge, i.e., replicates the post-mutual-attention layer for the knowledge. However, while TED focuses on assigning different weights to different sources, KTWM is already provided with scored knowledge terms. We use V_{PMA} to denote the post-mutual attention, V_{KMA} for knowledge-mutual attention and V_{dec} for the decoding tokens representation matrix. With the attention defined by Eq. (1), we can express:

$$V_{PMA} = Attention(V_{dec}, V_{post}, V_{post}) \quad (10)$$

$$V_{KMA} = Attention(V_{dec}, \widetilde{V}_{know}, \widetilde{V}_{know}). \quad (11)$$

The *final mutual attention* V_{MA} in the decoder is then calculated from V_{PMA} and V_{KMA}:

$$V_{MA} = V_{PMA} \oplus V_{KMA} \quad (12)$$

where $\oplus$ means element-wise summation. The feed forward layer is a standard Transformer transition layer (Vaswani et al. (2017)).

Finally, we adopt Negative Log Likelihood (NLL) to train the model:

$$\mathcal{L}_{NLL} = -\sum_{t=1}^{m} logP(y_t|y_{<t}, p, k). \quad (13)$$

Given a post (p), knowledge (k), and the previously predicted terms ($y_{<t}$), $\mathcal{L}_{NLL}$ maximises the probability of the currently predicted term. During the training phase, $P(y_t|y_{<t}, p, k)$ is replaced with $P(r_t|r_{<t}, p, k)$, i.e., we use the ground truth response as the input instead of the model output from the previous steps (Goyal et al., 2016).

We assume that all three loss functions are equally important and create the final loss function as a sum:

$$\mathcal{L} = \mathcal{L}_{filter} + \mathcal{L}_{approx} + \mathcal{L}_{NLL} \quad (14)$$

KTWM thus provides a flexible learning framework, enabling injection of knowledge based on different selection criteria. We compare KTWM effectiveness when used with Post-KS, SKT and TED model by incorporating the knowledge that each of these methods selects.

4 Experiments

We conduct empirical evaluation of KTWM compared to state-of-the-art baselines.

4.1 Datasets

In our experiments we use two publicly available datasets: Wizard of Wikipedia (Dinan et al., 2019) and Holl-E (Moghe et al., 2018). Both are purposefully created by humans editors to support dialogue generation research.

Wizard of Wikipedia (Wiz). Dinan et al. (2019) employed Amazon Mechanical Turk (MTurk) workers to generate the datasets. The workers can assume two different roles: a wizard (a teacher) and an apprentice (a student). An apprentice asks a question according to a given topic and a wizard answers the question based on the provided question-related information (retrieved from Wikipedia). The response can quote the retrieved knowledge or can be generated entirely by the wizard without considering the knowledge. Thus, for each question-response pair there is related knowledge that can be used for knowledge-grounded dialogue generation research.

The Wiz dataset consists of 22,311 dialogues with 201,999 dialogue turns divided into a training dataset and two test datasets referred to as *seen test set* and *unseen test set*. The seen test set includes topics that have already been seen in the training set. In the unseen dataset, there are topics that may not have been included in the training dataset.

Holl-E Moghe et al. (2018) also made use of MTurk workers to create an annotated dataset that focuses on movies as two workers talk with each other about a chosen movie. When answering another worker's question, one is provided with four sources: movie plots, reviews, comments, and fact tables related to the movies. These sources can be considered as background knowledge. The final response is produced by copying from the sources or by modifying the sources. The Holl-E dataset provides training set and test test and contains 9,071 conversations, covering 921 movies.

4.2 Metrics, Setup and Baselines

Metrics. For performance evaluation, we adopted standard lexical-based metrics: BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007) and embedding-based metric: BOW Embedding (Liu et al., 2016). BLEU 1-4 metrics measure co-occurrence of n-gram terms in two given sequences, e.g., the generated responses and the ground truth responses. METEOR is an adaptation of BLEU that considers the presence of synonyms and common word stems. BOW Embedding measures the similarity of two sentences from the semantic perspective. Specifically, it computes the *average metric*, *greedy metric* and *extrema metric* based on word embeddings of compared sentences. The average metric considers cosine distance between pairs of sentence-level representations (e.g., the predicted response and ground truth response) by averaging the representations of their constituent words and calculates the average across all pairs. The greedy metric considers the maximum cosine scores along rows and columns in the similarity matrix. The extrema metric of two sentences first creates a sentence vector with the highest word-embedding values (along the dimension) and then computes the similarity score. BLEU, METEOR[2] and BOW Embedding[3] are calculated using NLG evaluation sources.

Experiment Setup. For the sake of comparison, we fixed a set of parameters across all the experiments. The number of dimensions in embeddings is set to 100. The vocabulary size is 30,000. The vocabulary is obtained by ranking terms by word frequency in the training set. The minimum sequence length is set to 8 and the maximum length is 30. We train using mini-batches of size 64. We use Adam optimiser (Kingma and Ba, 2015) for optimisation. The initial learning rate is set to 0.001 and halved when the loss score does not decrease for two epochs. In the training phase we use response-retrieved knowledge, i.e., the sentences retrieved by BM25 algorithm using responses as queries (see Figure 1). The top 1 ranked knowledge sentence is injected into KTWM. In the test phase, we retrieve knowledge using BM25 algorithm and posts as queries. All the experiments are conducted on a single TITAN V GPU. For Wiz dataset, an experiment requires about 6 hours to complete, while for Holl-E about 2.5 hours.

Baselines. We compare KTWM with three strong baselines:

Post-KS (Lian et al., 2019) uses an elaborate knowledge selection module and injects the selected knowledge into a generative model by approximating prior-distribution (i.e., $p(k|p)$) with posterior-distribution (i.e., $p(k|p, r)$).

[2] https://github.com/Maluuba/nlg-eval

[3] https://github.com/neural-dialogue-metrics/EmbeddingBased.

SKT (Kim et al., 2020) considers knowledge selection as a sequential problem. It jointly trains a knowledge selection and a generative model by taking into account inputs and knowledge from previous turns.

TED (Zheng and Zhou, 2019) uses a knowledge-grounded generative model that assigns different weights to different sources when generating responses. It applies knowledge ranking using BM25, which is the same as in our setting.

4.3 Experiment Design

Our experiments focus on term weighting of the selected knowledge rather than the knowledge selection itself. Since the baseline models (Post-KS[4], SKT[5] and TED[6]) incorporate knowledge selections, we conduct a comparative evaluation of KTWM by incorporating knowledge specific to each baseline method. Furthermore, since all three baselines inject knowledge at the sentence level, by selecting the top ranked sentence, we do the same with KTWM.

4.4 Experiment Results

4.4.1 Performance of Generating Response

We summarize KTWM experiments with the Wiz and the Holl-E datasets in Table 2. Results for the Wiz seen and unseen test sets are in Table 2, sections (a) and (b), respectively. Results for the Holl-E dataset are in Table 2, section (c). Since METEOR extends BLEU metrics by considering word stems and synonyms, we take it as the main metric for discussing the experiment results. We observe that:

(1) For all of three datasets, KTWM outperforms each baseline method across all lexical and embeddings based metrics with a statistically significant difference.

(2) KTWM with Post-KS knowledge achieves the largest relative improvement considering the METEOR score: increase of 45.3%, 54.5% and 40.0% for the three test sets, respectively.

(3) For the Holl-E dataset, KTWM with TED knowledge outperforms other two baseline models. TED knowledge comprises top sentences retrieved using BM25 algorithm.

(4) On the Wiz datasets, KTWM achieves a remarkable performance in terms of BLEU-1 and METEOR scores. A consistent and strong performance in the Wiz unseen test data indicates the robustness and generalization of KTWM.

[4] https://github.com/bzantium/Posterior-Knowledge-Selection
[5] https://github.com/bckim92/sequential-knowledge-transformer
[6] https://github.com/tonywenuon/Transformer_ED

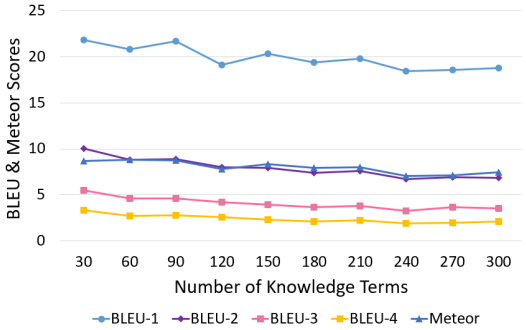

Figure 3: Effects of the increased number of knowledge terms on the KTWM performance (Wiz seen test set).

4.4.2 Results of Knowledge Term Weighting

The loss function (Eq. (7)) controls KTWM ability to distinguish between relevant and non-relevant knowledge terms, similar to a binary classifier. We set a threshold of 0.5 for a knowledge term's predicted score and consider the overlap between the predicted and the truth useful knowledge terms. This leads to precision/recall evaluation of the positive and the negative class prediction. Table 3 shows results from the Wiz seen test set. They are representative of the results for the other two datasets.

We observe that the precision of predicting useful terms is 50% and noisy terms is over 91% (with a high F-1 score, 94%). Thus KTWM term weighting is effective in detecting noisy terms while only half of the predicted useful terms overlap with the ground truth terms. Since noisy terms are assigned lower term weights, KTWM is effective improving the dialogue generation performance. Appendix A shows illustrations of the KTWM noise reduction.

4.4.3 Analysis of Input Sequence Length

We analyze the effects of knowledge de-noising by considering the *useful terms proportion* (UTP) as we increase the number of injected knowledge terms: $UTP = \frac{Num\ of\ distinct\ useful\ terms}{Num\ of all\ injected\ terms}$. We use UTP_K for UTP when the number of injected knowledge terms is K (e.g., UTP_{30} for 30 knowledge terms). Our analysis shows that UTP_{30} is 12.23% and UTP gradually decreases with additionally injected knowledge leading to UTP_{300} of only 3.35%. Figure 3 shows a gradual decline of the KTWM performance with the increased length of injected knowledge, as the proportion of noisy

Generation Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	Average	Greedy	Extrema
(a) Wiz seen test data								
Post-KS	17.56	6.35	2.68	1.35	5.96	0.611	0.364	0.334
KTWM w Post-KS knowledge	21.98*	**10.03***	**5.56***	**3.44***	**8.66***	**0.684***	0.394*	**0.376***
SKT	16.45	7.97	4.75	3.14	7.29	0.639	0.385	0.366
KTWM w SKT knowledge	**22.00***	10.0*	5.47*	3.35	8.59*	0.681*	**0.398***	0.370
TED	20.26	9.43	5.32	3.35	8.45	0.658	0.385	0.366
KTWM w TED knowledge	21.86	10.02	5.51	3.35	**8.66***	0.682*	0.394*	0.374*
(b) Wiz unseen test data								
Post-KS	17.25	5.58	2.03	0.81	5.5	0.598	0.352	0.305
KTWM w Post-KS knowledge	**21.66***	**8.98***	**4.41***	**2.41***	**8.5***	**0.681***	**0.388***	**0.361***
SKT	14.09	5.72	2.89	1.72	5.8	0.591	0.36	0.304
KTWM w SKT knowledge	20.46*	8.07*	3.85*	2.03*	7.77*	0.664*	0.38*	0.337*
TED	19.28	7.83	3.83	2.09	7.02	0.634	0.363	0.327
KTWM w TED knowledge	20.46*	8.32*	4.03*	2.17*	7.92*	0.668*	0.379*	0.342*
(c) Holl-E dataset								
Post-KS	14.07	7.07	4.96	3.81	5.98	0.639	0.382	0.333
KTWM w Post-KS knowledge	19.91*	11.0*	8.02*	6.42*	8.37*	0.675*	0.387*	0.350*
SKT	21.54	13.81	10.94	9.17	8.48	0.637	0.391	0.333
KTWM w SKT knowledge	**23.05***	13.96*	10.66	8.71	9.73*	0.673*	0.389	0.362*
TED	21.62	13.71	10.83	9.17	9.13	0.685	**0.414**	**0.366**
KTWM w TED knowledge	22.42*	**14.01***	**10.98**	**9.28**	**10.2***	**0.688***	0.402*	**0.366**

Table 2: KTWM performance on the Wiz seen and unseen test data and Holl-E dataset with different knowledge sources. Comparison with Post-KS, SKT and TED models. '*' indicates statistical significance ($p < 0.05$). **Bold** indicates the best performance for a given metric. 'w' denotes 'with', i.e., injecting the knowledge source that is used in a specific baseline model.

Name	Prec	Rec	F-1
Useful Term Prediction	0.50	0.32	0.39
Noisy Term Prediction	0.92	0.96	0.94

Table 3: Precision, Recall, and F-1 scores for the useful and noisy term predictions on the Wiz seen test set.

Name	BLEU-1	BLEU-4	METEOR	Average
KTWM	21.86	3.35	8.66	0.682
- w/o $\mathcal{L}_{Filter}$	20.69	3.67	8.77	0.661
- w/o $\mathcal{L}_{Approx}$	7.49	1.59	5.42	0.598

Table 4: Ablation study of the multi-component loss function on the Wiz seen test set. w/o means 'without'.

terms increases.

We also investigate the effects of the loss functions $\mathcal{L}_{filter}$ and $\mathcal{L}_{approx}$ on the KTWM performance by running experiments with and without them. In Table 4 we show the results on the Wiz seen test set using BM25 to select knowledge. We note that, after removing $\mathcal{L}_{filter}$ loss function, BLEU-1 and Average scores decrease, while BLEU-4 and METEOR scores increase. Since $\mathcal{L}_{filter}$ aims to ensure that relevant response terms are promoted, it is not surprising that the metrics focused on unigrams are most affected. However, this impact on KTWM is less notable than the removal of the $\mathcal{L}_{approx}$. Without $\mathcal{L}_{approx}$, the KTWM loses the ability to align simulated response vectors SRVs with the response embeddings to capture the attention distribution between the knowledge and the response embeddings that is needed to score knowledge terms. This increases the noise ratio and reduces the KTWM performance scores across all metrics.

5 Conclusions

Current knowledge-grounded dialogue models select and inject knowledge either through traditional (unsupervised) retrieval technique, such as BM25, or by incorporating knowledge selection within the dialogue generation model. Most of them incorporate knowledge as sentences or paragraphs. Past research provided evidence (Galley et al. (2019), Zheng et al. (2020)) that inserting useful terms can increase the response generation performance but it is necessary to control for negative effects of noisy terms.

In our work, we introduce a novel Knowledge Term Weighting Model (KTWM) that performs knowledge term-level weighting and de-noising of injected knowledge. We demonstrate that KTWM effectively estimates weights of knowledge terms and yields better response generation performance than state-of-the-art baseline models when evaluated on two broadly used datasets. Besides the superior response generation outcomes, our research

provides important insights into the importance of the knowledge term weighting. As part of our future work we intend to (1) extend the KTWM models to incorporate multiple sources of evidence, such as balancing between selected knowledge and dialogue contexts (i.e., previous dialogue turns) and (2) take into account inter-dependencies among terms when weighting the selected knowledge.

Acknowledgements This work is partly supported by Engineering and Physical Sciences Research Council (EPSRC Grant No. EP/S515528/1, 2102871). The Titan V used for this research was donated by the NVIDIA Corporation. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations (ICLR)*.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded response generation task at dstc7. In *AAAI Dialog System Technology Challenges Workshop*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: a new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4608–4616.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *International Conference on Learning Representations (ICLR)*.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. In *IJCAI International Joint Conference on Artificial Intelligence*, page 5081.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. 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*, pages 2122–2132.

Yi Luan, Chris Brockett, William B Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 605–614.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92.

Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. 2019. Deepcopy: Grounded response generation with hierarchical pointer networks. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 122–132.

Hao-Tong Ye, Kai-Lin Lo, Shang-Yu Su, and Yun-Nung Chen. 2020. Knowledge-grounded response generation with deep attentional latent-variable model. *Computer Speech & Language*, 63:101069.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations (ICLR)*.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390.

Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2020. Approximation of response knowledge retrieval in knowledge-grounded dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3581–3591.

Wen Zheng and Ke Zhou. 2019. Enhancing conversational dialogue models with grounded knowledge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 709–718.

Appendix

A Examples of Knowledge Term Weights and KTWM Generated Responses

In Table 5 and 6 we present examples of post/response pairs and selected knowledge with terms weighted by KTWM.

As explained in §4.4.2, we use a threshold of 0.5 on term scores to classify terms into useful and noisy ones and study the effect of this selection on the overall performance of KTWM. In the examples, we visually show the weights of each terms. Terms are highlighted in different shades of blue colour according to the weight (note the colour legend at the bottom of the tables). All the examples are extracted from the Wiz seen test set. They are sorted by the number of words that exceed the threshold.

In Table 5 we see that the key words are tagged with dark blue, indicating that KTWM has assigned high weights to them. From the KTWM generated responses, we can see that if the words appear in the post and ground-truth response simultaneously, the KTWM works effectively, i.e., can correctly incorporate injected knowledge into the generated response.

On the other hand, the negative examples in Table 6 show that the term scoring can be ineffective if there is no good overlap with the ground truth response. We observe in these examples that most of the words with relatively high scores do not exist in both post and response. At the same time, if the injected knowledge does not contain useful terms, the produced responses might be irrelevant. In Table 6, most of terms have light blue colour, indicating that KTWM detected a relatively low importance of these terms correctly.

The examples in these two tables also confirm statistical results shown and discussed in §4.4.3. KTWM term weights still induce noise, especially when the injected knowledge does not contain useful terms (i.e. terms that present in the ground truth response), resulting in a worse response generation performance.

We note that the both sets of examples include highlighted punctuation (e.g., ‘,’) and language structural terms (e.g., ‘the’, ‘is’) which obtain high KTWM weights. We assume that such terms are widely distributed in post and response sets and therefore detected as important.

Post: I love avengers!! Have you watched the Infinity War movie? I think it was really good and I can’t wait till the next movie to find out what happens. **Response:** The Avengers Infinity War was so good. What a cliff hanger. Blows my mind that the Avengers made their comic debut way back in 1963. **Knowledge Terms Weighted by KTWM:** In ” Avengers : Infinity War ” , the Avengers join forces with the guardians of the galaxy to confront Thanos , who is trying to collect the infinity stones **Response generated by KTWM:** I love the Avengers. the Avengers was the Avengers, Infinity War. He is such a great show.
Post: I have never actually been to a horse race in Kentucky, where is the Derby held at anyways? **Response:** Louisville Kentucky, it ends the Kentucky Derby festival usually. **Knowledge Terms Weighted by KTWM:** The Kentucky Derby presented by Woodford reserve is a horse race that is held annually in Louisville , Kentucky , United States , on the first Saturday in may . **Response generated by KTWM:** I’m not sure but I know that it is located in Kentucky race in the United States and Canada.
Post: New York city is quite impressive. **Response:** Yeah they say that New York and LA are the two largest markets in our country. **Knowledge Terms Weighted by KTWM:** The city of New York , often called New York City or simply New York , is the most populous city in the United States **Response generated by KTWM:** Yes, it is. The city of New York is the most populous city in the United States.
Post: Hi there! I drive a Honda Civic. Do you have a car? **Response:** I have a VW golf, but was looking to change to a Honda Civic. I heard that the Civic has got larger throughout the years. **Knowledge Terms Weighted by KTWM:** The Honda Civic type R is the highest performance version of the Honda Civic made by Honda Motor company of Japan **Response generated by KTWM:** I have a Honda Civic. They are the second most commonly used in the US in Japan.
Post: Well that’s interesting, but seems like it could get confusing! I bet humans have been hiking ever since we could walk though! **Response:** Yeah hiking has been around for ever, the term bush walking is endemic to Australia. **Knowledge Terms Weighted by KTWM:** Hiking is the preferred term , in canada and the united states , for a long , vigorous walk , usually on trails (footpaths) , in the countryside **Response generated by KTWM:** Well, hiking is a long term hiking is used in the United States, and the countryside on trails.
0 0.2 0.4 0.6 0.8 1

Table 5: Positive cases that the injected knowledge is properly weighted by the proposed KTWM model. The background colour stands for the probability of being a useful term.

Post: Yes, indeed. What cultural meanings come to mind when you think of dance?
Response: Some of its many functions that come to mind are social, ceremonial, and competitive in meaning.
Knowledge Terms Weighted by KTWM:
One open question regarding the nature of the mind is the mind body problem , which investigates the relation of the mind to the physical brain and nervous system
Response generated by KTWM: I'm not sure but I do know that the spiritual core is associated with mind and mind.

Post: I have written a few historical fiction novels that i am currently trying to get published.
Response: That's really cool, historical fiction is such a fun genre to read.
Knowledge Terms Weighted by KTWM:
Historical romance (also historical novel) is a broad category of fiction in which the plot takes place in a setting located in the past
Response generated by KTWM: That's great. I love romance novels. It's a broad category of historical fiction, and fantasy.

Post: I would have to say Snooki still, she is such a ball of fire.
Response: Since you like Snooki so much, did you happen to see the Spinoff of Jersey shore called "Snooki and Jwoww"?
Knowledge Terms Weighted by KTWM:
She attended Columbia high school , in east greenbush , New York , and New York Institute of Technology
Response generated by KTWM: I don't know much about Vietnamese style, but i know it has been around since 1700

Post: Are they popular in other areas of the world like Europe or Asia ?
Response: Similarly! In the European Union vehicles in this size are known as large good vehicles.
Knowledge Terms Weighted by KTWM:
Western civilisation traces its roots back to western Europe and the western Mediterranean
Response generated by KTWM: I'm not sure, but I do know that they are native to the Mediterranean Mediterranean cuisine.

Post: Wow that's interesting. The food world has really diversified.
Response: Yeah or you can cook without heat. Like in south America they make Ceviche, which is fish that is cooked in lemon or lime juice and the acid cooks the fish.
Knowledge Terms Weighted by KTWM:
Bow Wow Wow are an English 1980s new wave band , created by Malcolm Mclaren
Response generated by KTWM: Yes, it is! It's the world's most popular foods in the world.

Table 6: Negative cases that the injected knowledge is not properly weighted by the proposed KTWM model. The background colour stands for the probability of being a useful term.