Approximation of Response Knowledge Retrieval in Knowledge-grounded Dialogue Generation

Wen Zheng
University of Nottingham
wen.zheng@nottingham.ac.uk

Natasa Milic-Frayling
University of Nottingham
natasa-milic@frayling.net

Ke Zhou
University of Nottingham & Nokia Bell Labs
Ke.Zhou@nottingham.ac.uk

Abstract

This paper is concerned with improving dialogue generation models through injection of knowledge, e.g., content relevant to the post that can increase the quality of responses. Past research extends the training of the generative models by incorporating statistical properties of posts, responses and related knowledge, without explicitly assessing the knowledge quality. In our work, we demonstrate the importance of knowledge relevance and adopt a two-phase approach. We first apply a novel method, **T**ransformer & **P**ost based **P**osterior **A**pproximation (**TPPA**) to select knowledge, and then use the Transformer with Expanded Decoder (TED) model to generate responses from both the post and the knowledge. TPPA method processes posts, post related knowledge, and response related knowledge at both word and sentence level. Our experiments with the TED generative model demonstrate the effectiveness of TPPA as it outperforms a set of strong baseline models. Our TPPA method is extendable and supports further optimization of knowledge retrieval and injection.

1 Introduction

In recent years, there have been concerted efforts to model dialogue interactions and generate an appropriate *response* to an initial user statement, referred to as a *post*. Research has led to generative models, e.g., Sequence-to-Sequence (Sutskever et al. (2014)) and Transformer (Vaswani et al., 2017), that produce reasonable responses using the original post solely during the generation process.

Recent studies (Weston et al., 2018; Ghazvininejad et al., 2018; Zheng and Zhou, 2019) explored more realistic dialogue models that include knowledge related to the posts, typically a collection of sentences that refer to the topics in the posts and responses. Consequently, the response generation process involves an information retrieval component that needs to be optimized for the selection and injection of relevant knowledge into the generative model.

Wiz Post: Yep. you've got to select for safety standards, of course, but when you're designing at a Mercedes level the folks buying those cars are going to expect a certain standard of comfort, too! **Wiz Response:** Especially, I think consumers expect great in Formula One, highest class auto racing.
TPPA (top 1): Formula One (also Formula 1 or F1 and officially the FIA Formula One World Championship) is the highest class of single seat auto racing that is sanctioned by the Federation Internationale de l'Automobile (FIA). **TPPA (top 2):** Stock car racing is a form of automobile racing found mainly and most prominently in the United States and Canada, with Australia, New Zealand and Brazil also having forms of stock car auto racing.
PRK (top 1): Mercedes is part of the McQueen family and is the longest serving McQueen on the series. **PRK (top 2):** He also won races in midget cars, and sprint cars.
RRK (top 1): Formula One (also Formula 1 or F1 and officially the FIA Formula One World Championship) is the highest class of single seat auto racing that is sanctioned by the Federation Internationale de l'Automobile (FIA). **RRK (top 2):** The FIA Formula One World Championship has been one of the premier forms of racing around the world since its inaugural season in 1950.

Table 1: Example of a post and a response from the Wizard of Wikipedia (Wiz) data set (§5.1) with top 2 ranked outputs from TPPA, the post-retrieved knowledge PRK and the response-retrieved knowledge RRK. Blue indicate words present in the Wiz response and RRK but not in PRK.

Evaluation of such approaches has shown that the knowledge based on posts alone may lack focus, i.e., may exhibit topic drifts and thus introduce noise. Table 1 illustrates Post-Retrieved Knowledge (PRK) that has a good overlap with the post but introduces content that is not present in the response and thus deemed non-relevant. By contrast, the Response-Retrieved Knowledge (RRK) shares content with the response, thus illustrating that dialogue training needs to incorporate relevant knowledge related to the response.

In practice, however, the key challenge is to implement an effective selection of response related knowledge, considering that the responses to posts

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3581–3591

are not observed during dialogue generation. In this paper, we present the **T**ransformer & **P**ost based **P**osterior **A**pproximation (**TPPA**) method that achieves that by applying multi-stage processing of posts, post related knowledge and response related knowledge to capture word and sentence level characteristics (through word embeddings, Transformer and max-pooling), that can be useful for ranking and selecting knowledge of new posts during the test phase.

Table 1 illustrates the high overlap that TPPA outputs achieve with true responses (for post-response pair from the Wizard of Wikipedia (Wiz) data collection (Dinan et al., 2019). Furthermore, we empirically demonstrate the effectiveness of TPPA, by injecting TPPA selected knowledge into generative models, in particular the Transformer Extended Decoder (TED) that allows integrating knowledge from multiple sources (Zheng and Zhou, 2019). The combination of TED and TPPA outperforms a set of strong baseline systems, including systems that do not separate knowledge selection from modelling response generation: Post-KS (Lian et al., 2019) and SKT (Sequential Latent-knowledge Selection) (Kim et al., 2020).

Most important contributions of our work are:

1. Empirical evidence that generative models with injecting response-retrieved knowledge outperform those that use only post-retrieved knowledge (§3).

2. New method for knowledge selection (TPPA) that includes Transformer-based representations of posts and post related knowledge to select relevant knowledge processed with word embedding and MaxPooling (§4).

3. Experimental results that demonstrate the benefit of TPPA knowledge injection into the TED generative model (Zheng and Zhou, 2019), outperforming state-of-the-art models on two publicly available data sets (§5, §6).

In addition, the separation of the knowledge selection from the generative models offers maximum flexibility for integrating and exploring alternative retrieval models and knowledge representations. We make our codes publicly available at *https://github.com/tonywenuon/emnlp2020_tppa*.

2 Related Work

In this section we first discuss retrieval models and then knowledge injection into generative models.

Retrieval Models. Most traditional retrieval models, such as BM25 (Robertson et al., 2004), are unsupervised methods, relying on lexical matching between query terms and document text using different weighting and normalization schemes. In contrast, recent studies use neural ranking models, such as deep structured semantic models (DSSM) (Huang et al., 2013; Shen et al., 2014), weakly supervised neural ranking models (Dehghani et al., 2017) and jointly trained neural models (Yan et al., 2016; Mitra et al., 2017). They are built to respond to information needs represented by a query. We illustrate our approach by adopting BM25 for initial retrieval of relevant knowledge. We also use the post related results to create an extended representation of the post, similar to the pseudo-relevance-feedback in query-based search (Cao et al., 2008).

Generative Models & Knowledge Injection. Injection of knowledge into generative models has been pursued to improve the quality of responses, considering that during dialogue generation only a post and related knowledge are observed. Ghazvininejad et al. (2018) encode and merge knowledge with a post representation, creating a final vector representation that is input into the decoder. Tam (2020) extends this method with a copy-mechanism that enables the model to generate response words either from the post or from the generative model.

Zheng and Zhou (2019) use the Transformer Extended Decoder (TED) to incorporate words from multiple sources by assigning weights to knowledge sources based on relevance between the knowledge and the decoding words, and taking the weighted-sum vector to generate responses.

Closest to our work is the PostKS model proposed by Lian et al. (2019) that includes a knowledge manager which fits the prior word distribution (from posts) to the posterior word distribution (with both post and response observed). By applying the Gumbel-Softmax method, they select the best knowledge for the dialogue generation. Similarly, the sequential latent-knowledge selection (SKT) proposed by Kim et al. (2020) jointly trains the knowledge selection and the dialogue generation model. Both methods consider knowledge relevance to posts and responses during training but do not leverage post-retrieved knowledge during testing.

Our proposed Transformer & Post based Posterior Approximation (TPPA) model distinguishes

itself by explicitly incorporating response related knowledge into training and applying pseudo relevance feedback approach by training an auto-pointer vector to identify potentially the most relevant knowledge. Combined with the TED generative model, TPPA leads to responses that outperform state-of-the-art methods (§6).

3 Problem Statement and Motivation

3.1 Key Notations and Research Objectives

Dialogue generation models that incorporate knowledge aim to expand the input beyond the observable post and incorporate a responder's knowledge. It is assumed that the available knowledge K_p for a given post p includes content that is related to the response, although the quality of that knowledge is not certain. The key issue is, thus, to determine which of the knowledge statements $k \in K_p$ are relevant to the unobserved response r. During the training phase, where the post p, response r and K_p are all available, we use p and r as queries to rank all the statements in K_p and create the corresponding ranked lists: Response-retrieved Knowledge RRK and Post-retrieved Knowledge PRK, respectively. We use lower-case rrk_1 and prk_1 to indicate top 1 ranked item in RRK and PRK, respectively.

3.2 RRK Assessment on Wiz Training Data

In this section, we analyze RRK for the Wiz training data (§5.1) where both posts p and responses r are known as well as the corresponding knowledge set K_p. Assuming that we deploy a reasonable search algorithm, we expect that rrk_1 will have a high overlap with the response r that is used as a query. We also assume that generative models will be able to use rrk_1 to generate a good quality response considering its overlap with the true response. The objective of this section is to gain insights on what difference RRK can make compared to the use of PRK alone.

Word count. We compare the number of common words (after removing stop words) between the original response r and the four sequences: (1) the post p, (2) prk_1, i.e., the top 1 ranked item in PRK, (3) rrk_1, i.e., the top 1 ranked item in RRK, and (4) a random post chosen from the data set. The distributions of word overlaps are shown in Figure 1. The x-axis indicates the count of common words and y-axis shows the percentage of the posts p and responses r sample with the given word overlap.

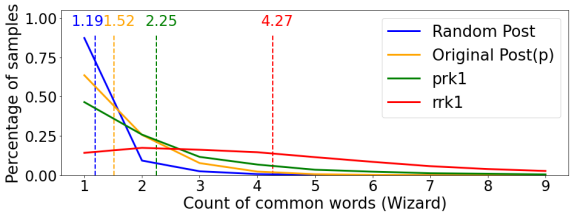

Figure 1: Common words count distribution between each source and the target response on the Wiz training set. The dashed lines are the average count of common words of each group (after removing stop words).

As expected, the word overlaps of p and prk_1 with r are similar, with the overlap of p and r being lower. For the randomly selected post p, the average term overlap with r is slightly lower but close to post p, suggesting that posts alone are not very informative for the response generation. The difference for prk_1 and rrk_1 is quite marked showing that rrk_1 has on average almost twice the overlap of the prk_1 (98% increase). Based on the Kolmogorov–Smirnov test, all the differences among the four groups in Figure 1 are statistically significant. For the Holl-E data set (that is another data set we used in §5.1), a similar trend is observed.

Response generation. We assess the effectiveness of RRK when injected into the generative model by conducting experiments with the standard Transformer (Vaswani et al., 2017) and the Transformer with Expanded Decoder (TED) (Zheng and Zhou, 2019). Transformer takes only a post while TED uses a post and multiple sources of knowledge to get the responses.

Table 2a shows the results for Transformer with (1) original post, (2) a randomly selected sentence, (3) prk_1, (4) rrk_1 and (5) a human selected knowledge, i.e., a sentence provided in Wiz. Table 2 with results metrics (BLEU, METEOR and Div-2, §5) show that replacing the original post by a randomly selected sentence reduces the performance significantly. Using prk_1 leads to lower performance, indicating a possible topic drift and noise. Using rrk_1 shows promising performance improvement; with higher retrieval performance, it may achieve the effectiveness of the human selected knowledge. Similarly, for the TED generative model, we incorporate the post content and evaluate the cumulative effect of adding knowledge from different sources. As expected, the best performance is achieved by the human selection of knowledge followed by the RRK (Table 2b).

In conclusion, it is worthwhile putting an ef-

(a) Transformer	BLEU-4	METEOR	Div-2
Original Post	1.76	6.6	7.3
Random Post	0.39	4.47	0.19
prk_1	1.23	6.36	5.62
rrk_1	**2.85**	**7.99**	**12.88**
Human selection	4.6	9.97	18.86
(b) TED	BLEU-4	METEOR	Div-2
Post+1 Random sentence	2.8	7.13	18.73
Post+prk_1	3.35	8.45	16.2
Post+rrk_1	**8.14**	**11.36**	**24.63**
Post+Human selection	10.06	13.13	25.7

Table 2: Injection of various sources into the Transformer and TED using Wiz data set. All the values are percentages reported by the performance metrics (%).

fort to create resources that represent a responder's knowledge and effective retrieval methods to retrieve knowledge relevant to the response content. Since the response is not available, we devise TPPA to leverage post p and post-retrieved knowledge PRK and train models to approximate RRK.

4 TPPA Method

In this section, we describe the architecture and the process of selecting knowledge using the TPPA method. Figure 2 depicts three TPPA components: (**1**) **Post Processing Unit** comprising a word embedding and a Transformer that incorporates the post p and a set of n of retrieved prk_i, where n is determined empirically (typically $n = 10$ out of 50 knowledge items in K_p, on average). The results are a Transformer representation v_p for the post and v_{PRK} for all of the $prks$. In the end, a single v_{prk} (representing the potentially most useful prk for identifying the rrk_1) is selected based on Auto-Pointer and Gumble Softmax algorithms.
(**2**) **Response Processing Unit** that, during training, considers each response r and corresponding K_p to get rrk_1 and a set of $negs$ (i.e., m negative samples which are non-relevant knowledge to the rrk_1) in order to train a word embedding that forms knowledge representation (we call it as v_k). The number of negative examples m is selected empirically, to avoid overfitting.
(**3**) **Knowledge Selection Unit**, a search component that uses v_p and v_{prk} as queries to score the knowledge representation v_k. The score is a weighted sum of similarity metrics using a hyperparameter α that can be chosen to emphasize the similarity with p or prk.

TPPA operation consists of **Phase 1**: Training phase that utilizes training data (p, r, K_p) to train all the three components of the system based on known responses r; and **Phase 2**: Test phase during which individual post-knowledge samples (p, K_p) are processed in order to arrive at a selection of knowledge $(k \in K_p)$ to be injected into the generative models.

4.1 TPPA Training phase

4.1.1 Post and PRK Processing

The post p and a set of $prk_i, i = 1, ..., n$ (i is the i-th ranked post-related knowledge) are processed with the same Transformer encoder to obtain word representations and then passed through the max-pooling to obtain the sequence semantic vector.

$$e(p) = \textit{Transformer}_{\Theta}\,(e(w_i))\; 1 \leq i \leq L \quad (1)$$

$$v_p = \textit{maxpool}\,(e(p)) \quad (2)$$

where Θ is the trainable parameter set inside the Transformer. *p* is the input post, w_i is the *i-th* word of the *p* post sequence. *L* is the maximum post length. $e(w_i) \in \mathbb{R}^d$ is the post word embedding for w_i, and d is the embedding dimension. $e(p)$ represents the semantic representation of all the words in the post while v_p is the post representation (sentence-level). For the prk_i, they follow exactly the same process following Equation 1 and 2.

We consider multiple knowledge items prk_i in order to construct an effective query for knowledge selection that complements the post and increases the chances of selecting knowledge that is relevant to the response. We train an auto-pointer to assign scores to each prk_i. The auto-pointer module takes v_{PRK} as input and outputs a PRK scores vector (v_{ap}) that indicates the importance degree of the $prks$. This is followed by a Gumbel-Softmax (Jang et al., 2016) module to select the best prk for knowledge retrieval:

$$v_{ap} = (v_{PRK}W^T + b)W^T_{auto\_pointer} \quad (3)$$

$$v_{prk} = \textit{Gumbel-Softmax}(v_{ap}, v_{PRK}) \quad (4)$$

where $v_{PRK} \in \mathbb{R}^{n \times d}$ represents all prk_i representations obtained by Eq. 1 and 2 and v_{prk} is the representation of the finally chosen post-related knowledge. $W \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^d$ are trainable parameters; $W_{auto\_pointer} \in \mathbb{R}^{1 \times d}$ is the trainable auto-pointer for selecting useful *prk*.

4.1.2 Response Processing Unit

The knowledge representation v_k is obtained by going through raw knowledge word embedding[1]

[1]Alternative approaches, e.g., using Transformer based representations, were considered but led to sub-optimal results within the current TPPA set up.

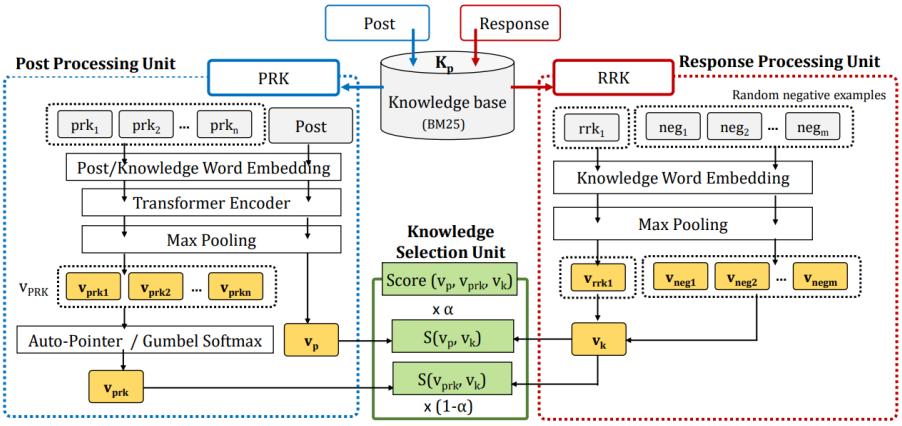

Figure 2: TPPA Architecture comprises (1) Post Processing Unit, (2) Response Processing Unit (right) and (3) Knowledge Selection Unit (middle).

and a max-pooling operation (seeing Figure 2 Response Processing Unit). The conduction of obtaining v_k is similar to Eq. 1 and 2 but replacing the Transformer to a raw knowledge word embedding lookup operation.

Since the objective is to augment vocabulary and avoid noise, during training, we constrain the positive knowledge to the highly relevant knowledge item, i.e. rrk_1 by using BM25. We also randomly select knowledge to be as negative samples (from the union of all K_p after the $rrk_1 s$ of the posts are removed). Both of the positive sample and negative samples will pass through the Response Processing Unit to gain their representations.

4.1.3 Knowledge Scoring and Selection

Following the post v_p and v_{prk} representation and knowledge representation v_k, we compute similarities $S(p,k)$ and $S(prk,k)$:

$$S(p,k) = \frac{cosine(v_p, v_k)}{\|v_p\| \cdot \|v_k\|}; S(prk,k) = \frac{cosine(v_{prk}, v_k)}{\|v_{prk}\| \cdot \|v_k\|} \quad (5)$$

where $S(\cdot)$ designates the similarity function; v_p, v_k and v_{prk} refer to the representations of the post, knowledge and the selected prk, respectively.

Depending on a type of dialogue, the response may incorporate the content of the post to a different degree. Thus, to support flexible scoring with regards to p and prk, we introduce a hyper-parameter, α to the final scoring function:

$$Score(p,prk,k) = \alpha \times S(p,k) + (1-\alpha) \times S(prk,k) \quad (6)$$

We tune α parameter on the training set and in the final $Score(p,prk,k)$, setting it to 0.7 to give more importance to the post.

After we get the scores of the positive and negative samples, for all the positive-negative sample pairs, we apply softmax to the similarity scores:

$$P(k_i|p,prk) = \frac{exp(\lambda Score(p,prk,k_i))}{\sum exp(\lambda Score(p,prk,k_i)} \quad (7)$$

calculating the probability of each k_i given the post p and $prks$. $k_i \in \{rrk_1; neg_1, neg_2, \ldots, neg_m\}$ are shown in the response processing unit in Figure 2, where $neg_1, \ldots, neg_m$ are m negative samples. λ is a smoothing factor of the softmax function and is a trainable parameter (Huang et al., 2013). We maximise the difference between the positive sample and the negative samples scores.

$$Loss = \sum \left(-log(P(rrk_1|p) + \sum_j log(neg_j|p)) \right) \quad (8)$$

where $P(rrk_1|p)$ is the positive score, $P(neg_j|p)$ stands for the *j-th* negative score, where $1 \leq j \leq m$. m is the number of negative samples. During training, all of the trainable parameters, including the post word embedding, Transformer architecture, auto-pointer and the knowledge word embedding, are updated by mini-batch gradient descent (the setup is in §5.2).

4.2 TPPA Test Phase

During the test phase, each new post p and corresponding K_p is processed using the Post Processing Unit and Response Processing Units, with parameter obtained during the training phase. Each knowledge k_i and its corresponding post are scored using the $Score(p,prk,k_i)$ (Eq. 6) and TPPA returns the final rank of the knowledge candidates.

5 Experiments

Our approach for knowledge injection separates the *knowledge selection* from the *response generation* models. We, thus, evaluate TPPA in terms of (1) precision in selecting relevant knowledge for a given post, judged by whether the rrk_1 can be ranked within top n position, and (2) effectiveness of the retrieved knowledge when injected into a response generation model.

5.1 Data

We experiment with two publicly available data sets: **Wizard of Wikipedia (Wiz)** (Dinan et al., 2019) comprises controlled human-to-human dialogue interactions where the participant can assume the role of a teacher or a student and take turns to discuss a topic. A teacher answers a student's post based on pre-retrieved knowledge that is related to the current topic and the dialogue context. The Wiz data set consists of 22,311 dialogues with 201,999 turns. Each post-response pair is assigned the related-knowledge, i.e., manually selected relevant sentences or paragraphs from Wikipedia.
Holl-E (Moghe et al., 2018) comprises dialogues between two Amazon Mturk workers[2] about a selected movie, supported by selected sources of background knowledge: movie plots, reviews, comments, and the fact tables related to the movie. A response to a post is either copied or suitably modified from the provided grounded knowledge, mixed from the four knowledge sources. Holl-E data contains 9,071 conversations, covering 921 movies.

5.2 Baselines, Setup and Metrics

Baselines In our experiments we compare TPPA knowledge selection on the retrieval performance with three baseline models: ***BM25*** (Robertson and Walker, 1994) is an unsupervised probabilistic retrieval algorithm, which is robust for short document (sentence) retrieval. ***DrQA*** (Chen et al., 2017) uses bigram hashing and TF-IDF matching with a multi-layer recurrent neural network model. ***CNN-DSSM*** (Shen et al., 2014) uses CNN for semantic matching of queries and documents.

In order to evaluate the effectiveness of the selected knowledge for *response generation*, we compare TPPA output with three models: ***WSeq*** (Tian et al., 2017) uses weighted sum and concatenation of the post and its contextual utterances, and obtain representations through an RNN. ***MemNet*** (Ghazvininejad et al., 2018) leverages a multi-task learning framework to jointly train 'post-to-response', 'knowledge-to-response' and 'knowledge-to-knowledge' tasks for response generation. ***TED*** (Zheng and Zhou, 2019) adopts Transformer as the backbone framework to inject knowledge by assigning weights to the knowledge from multiple sources.

Finally, we consider two methods that jointly train knowledge selection model and dialogue generation model, and use them in both sets of experiments: ***Post-KS*** (Lian et al., 2019) approximates posterior-distribution of knowledge, i.e., $p(k|p,r)$ using prior-distribution $p(k|p)$ and jointly train a knowledge selection model and a dialogue generation model. ***SKT*** (Kim et al., 2020) takes into a account context from multi-turn dialogues (current action and 2 prior turns) and considers knowledge selection as a sequential decision process.

Experimental Setup In our experiments, the dimension of word embedding is 300, and the multi-head number of Transformer is 4. The vocabulary is obtained by ranking the training data by word frequency, with the size of 50,000 top frequent terms selected. The minimum post length is set to 8 tokens. Each knowledge item is represented by a sentence. During model training, we use mini-batch size 64. Adam optimiser is used for optimisation. The initial learning rate is set to 0.001 and halved when reaching the plateau (decreasing patience is set to 2 epochs). All the experiments are run on a single TITAN V GPU. The TPPA model requires 2 hours to train on the Wiz data set.

Metrics Quality of the generated responses is evaluated using five standard metrics: *BLEU* (Papineni et al., 2002), *Meteor* (Banerjee and Lavie, 2005), and *Bert-Score (BS)* (Zhang et al., 2019) that are based on co-occurrence of n-grams between the system response and the ground-truth, calculating the token similarity using contextual embeddings. In this work, the BS version we used is *roberta-large_L17_idf_version=0.3.3(hug_trans=2.8.0)*[3]; *Diversity scores (Div-2)* (Li et al., 2015) calculates the proportion of distinct bi-grams out of all the distinct words.

For knowledge selection, we use *P@n* that calculates the precision at a given rank n, measuring whether the ground truth (rrk_1) exists within the top n retrieved knowledge.

[2] Amazon Mturk is a crowd-sourcing marketplace that can employ workers to annotate corpus, https://www.mturk.com/.

[3] https://github.com/Tiiiger/bert_score

Exp Model		Wizard of Wikipedia (%) P@1	P@5	P@10
BM25		4.9†	18.6†	31.1†
DrQA		4.1†	13.6*†	21.7*†
CNN-DSSM		8.2*†	31.3*†	48.8*†
Post-KS		6.2*†	-	-
SKT		9.01*	-	-
TPPA	1rrk-1neg-10prk	8.9*†	33.0*†	49.2*†
	1rrk-4neg-10prk	10.0*	36.5*†	54.5*
	1rrk-10neg-10prk	9.8*	36.4*†	54.2*†
	1rrk-20neg-10prk	10.1*	37.8*	55.0*
	1rrk-30neg-10prk	**10.1***	**38.0***	**55.1***
	1rrk-40neg-10prk	8.2*†	31.3*†	48.2*†
TPPA	1rrk-30neg-1prk	10.2*	38.4*	55.1*
	1rrk-30neg-10prk	10.1*	38.0*	55.1*
	1rrk-30neg-20prk	10.0*	37.3*†	55.1*
	1rrk-30neg-30prk	9.7*	35.2*†	52.4*†
Exp Model		Holl-E (%) P@1	P@5	P@10
BM25		10.5†	33.4†	48.5†
DrQA		13.3*†	29.4*†	35.4*†
CNN-DSSM		15.2*†	34.9*†	50.0†
Post-KS		5.5*†	-	-
SKT		11.6*†	-	-
TPPA	1rrk-1neg-10prk	13.6*†	37.0*†	51.3*†
	1rrk-4neg-10prk	15.5*†	38.3*†	52.7*†
	1rrk-10neg-10prk	**16.6***	**40.4***	**54.5***
	1rrk-20neg-10prk	14.8*†	36.9*†	51.1†
	1rrk-30neg-10prk	15.7*†	39.1*†	53.2†
	1rrk-40neg-10prk	16.2*	39.5*	53.2
TPPA	1rrk-10neg-1prk	16.3*	39.0*†	52.7*†
	1rrk-10neg-10prk	16.6*	40.4*	54.5*
	1rrk-10neg-20prk	16.6*	39.0*	52.9*†
	1rrk-10neg-30prk	15.4*†	38.6*	52.7*†

Table 3: Retrieval precision on the Wiz and Holl-E data sets. '*' means t-test $p < 0.05$ compared with the baseline $BM25$; '†' is the $p < 0.05$ compared with the best performing group. **Bold** indicates the best performance group when changing the number of negative samples. Underline indicates the best group among all methods.

6 Experimental Results

Knowledge Selection Evaluation. For the TPPA method, the quality of the selected knowledge is determined by the embedding parameters obtained during the training phase. They are, in turn, related to the knowledge resources used for training (Response Processing Unit) and the quality of the transformer representation of p and prk (Post Processing Unit), shown in Figure 2. The resources are constructed from individual knowledge sets K_p, where p is the post in the training set. For each training sample, it consists of a post p, a rrk_1 (i.e. the top 1 ranked response-retrieved knowledge), n $prks$ (i.e. the top n ranked post-retrieved knowledge) and m $negs$ (i.e. randomly chosen m sentences). Thus, $1rrk$-$1neg$-$10prk$ indicates that we selected the rrk_1, 1 random knowledge item and top 10 $prks$ for each p. In the test experiments, we monitor whether, for a new post p in the test set, different retrieval models rank its corresponding ground truth, i.e., rrk_1 for p within the top 1, 5, or 10 ranked items.

Results in Table 3 show that: **(1)** TPPA provides at least one model that outperforms all other models on the Wiz and Holl-E data sets, on all three metrics P@1, P@5, and P@10. **(2)** The composition of the knowledge base affects the TPPA knowledge selection: for the Wiz data set and fixed number of $10prk$, increasing the number of neg items improves the performance until reaching its plateau at $1rrk$-$30neg$-$10prk$; for the Holl-E data set, the best combination is $1rrk$-$10neg$-$10prk$. **(3)** For a fixed number of neg we vary the number of $prks$ items and find that: (i) for Wiz and n=30, the optimal prk number is 1; and (ii) for Holl-E and neg=10 the optimal prk number is 10.

Exp Model	Wizard of Wikipedia (%) BLEU-4	METEOR	Div-2	BS
MemNet	1.24	6.39	2.24	81.5
WSeq	2.13	7.17	13.29	82.86
Post-KS	1.35	5.96	22.32	81.3
SKT	3.14	7.29	27.8	83.4
TED	3.91	8.82	18.16	82.9
Exp Model	Holl-E (%) BLEU-4	METEOR	Div-2	BS
MemNet	5.59	7.63	0.18	84.6
WSeq	5.9	7.94	3.63	83.71
Post-KS	3.79	5.98	2.41	81.3
SKT	9.16	8.48	22.9	82.9
TED	12.66	10.37	17.95	84.1

Table 4: Performance of generative models MemNet, WSeq and TED with the best TPPA knowledge selection. Post-KS and SKT rely on their jointly trained models. BS refers to Bert-Score.

Based on these findings we use $1rrk$-$30neg$-$1prk$ for Wiz and $1rrk$-$10neg$-$10prk$ for Holl-E as sets for TPPA to select knowledge for use with MemNet, WSeq and TED models on response generation.

Response Generation Evaluation. We conduct the initial set of experiments to assess the robustness of the generative models (Table 4) and find that: (i) SKT and TED models outperform others, (ii) MemNet has unstable performance and constantly under-performs on Div-2. Furthermore, since SKT and Post-KS cannot inject multiple knowledge items, for further discussion, we choose experiments with WSeq and TED. We combine them with knowledge selection from (i) BM25, (ii) SKT (single knowledge item), (iii) CNN-DSSM (supervised search algorithm on post only), (iv) TPPA using both post and post-retrieved knowledge items, and (v) rrk_i (i means top i ranked response-retrieved knowledge, it is set to 1, 5 and 10 in our setting), to determine the upper bound when responses are known). The comparisons for the two data sets are shown in Table 5 and Table 6.

We observe that: **(1)** Injecting knowledge from

TED+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	3.35	8.45	16.2	82.7
SKT	**4.05***	**8.82***	18.8*	82.8*
CNN-DSSM	3.5	8.62	**20.08***	82.8
TPPA	3.91*	**8.82***	18.16	**82.9**
rrk_1	8.14*	11.36*	24.63*	84.3*
TED+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	3.17	7.81	**18.33**	82.99
CNN-DSSM	3.81	8.82	16.98	83.16
TPPA	**3.88***	**8.97***	17.22*	**83.23**
rrk_5	4.99*	10.49*	19.04*	83.7*
TED + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	3.01	7.98	15.7	83.2
CNN-DSSM	**3.59***	8.98*	**14.8***	83.38
TPPA	3.53*	**9.09***	14.66*	**83.4***
rrk_{10}	4.05*	9.56*	15.87*	83.6*
WSeq+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	1.94	6.98	12.96	82.76
SKT	2.0	7.02	**13.73**	82.8
CNN-DSSM	2.04	7.07	13.25	82.81
TPPA	**2.13**	**7.17***	13.29	**82.86**
rrk_1	2.23*	7.35*	13.23	83.0*
WSeq+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	2.05	7.18	17.59	82.85
CNN-DSSM	2.07	7.37	18.32	83.03*
TPPA	**2.15***	**7.57***	**18.55***	**83.1***
rrk_5	2.61*	8.0*	18.75*	83.3*
WSeq + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	2.31	7.44	19.48	83.0
CNN-DSSM	2.44	7.88*	**20.19**	83.3*
TPPA	**2.59**	**7.97**	19.72	**83.35**
rrk_{10}	3.01*	8.67*	21.07	83.66*

Table 5: Knowledge-injection results on the Wizard of Wikipedia data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5' 'Top 10' denotes injecting top 1 or 5 or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group.

TED+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	9.87	9.09	**26.21**	83.6
SKT	9.01	8.56	19.86*	83.4*
CNN-DSSM	11.56*	9.84*	23.51*	83.9
TPPA	**12.66***	**10.37***	17.95*	**84.1***
rrk_1	45.94*	30.61*	29.03*	89.6*
TED+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	11.4	10.22	**24.16**	83.9
CNN-DSSM	12.02	10.4	23.71	84.0
TPPA	**12.92***	**11.12***	17.87*	**84.2**
rrk_5	21.81*	17.15*	24.96*	85.9*
TED + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	5.5	**8.36**	2.45	83.5
CNN-DSSM	5.39	8.24	**2.6***	**83.6**
TPPA	**5.6**	8.24	2.53*	**83.6**
rrk_{10}	6.53*	9.88*	2.75*	84.0*
WSeq+Top 1	BLEU-4	METEOR	Div-2	BS
BM25	4.58	7.25	4.33	83.68
SKT	5.81*	7.77*	3.09	83.6*
CNN-DSSM	5.6*	7.62*	**4.48***	83.5*
TPPA	**5.9***	**7.94***	3.63*	**83.71**
rrk_1	6.5*	8.95*	4.6*	83.97*
WSeq+Top 5	BLEU-4	METEOR	Div-2	BS
BM25	5.15	7.51	8.65	83.43
CNN-DSSM	5.53*	7.69	**9.78***	83.17*
TPPA	**5.96***	**7.74***	7.82*	**83.59***
rrk_5	7.22*	9.55*	9.39*	83.85*
WSeq + Top 10	BLEU-4	METEOR	Div-2	BS
BM25	5.28	7.15	13.85	83.43
CNN-DSSM	5.88*	7.35*	**16.26***	83.3*
TPPA	**5.89***	**7.43***	12.43*	**83.7***
rrk_{10}	8.19*	10.41*	15.73*	84.3*

Table 6: Knowledge-injection results on the Holl-E data set. The values are percentages (%). '*' means the t-test $p < 0.05$ compared with the BM25 algorithm. 'Top 1', 'Top 5', 'Top 10' denotes injecting top 1 or 5, or 10 ranking knowledge. BS is Bert-Score. **Bold** indicates the best score apart from the rrk_i group.

SKT, CNN-DSSM and TPPA generally outperforms the post only selection using BM25 (Table 5 and 6) on both the Wiz and Holl-E data sets in terms of the BLEU-4, METEOR and Bert-Score. TED performance suffers from increased knowledge injection. Indeed, for TED + rrk_i, i.e., using 'perfect knowledge' the performance decreases with the increasing number of knowledge items. Zheng and Zhou (2019) claim that TED lacks a noise-filtering mechanism and thus underperforms with too much data. **(2)** Not surprisingly, knowledge selection methods with better retrieval performance achieve better response generation metrics. We consider Table 5 and 6 and the corresponding retrieval performance in Table 3. For the Wiz data set, the TPPA with $1rrk$-$30neg$-$1prk$ achieves the best retrieval performance and better results (Table 5) on both generative models (TED and WSeq) across different settings. This is confirmed on the Holl-E data set (Table 6) where TPPA outperforms other models, including Post-KS and SKT. This confirms our conjecture that improving retrieval for knowledge injection should improve the response generation.

Upper-bound Analysis. The upper bound for knowledge selection is the rrk_i group. We observe how all of the retrieval models perform in combination with TED and WSeq (Table 5 and 6). For the sake of concreteness we focus on the BLEU-4 metric. Table 5 and 6 show that low levels of knowledge-injection, e.g., a single knowledge item (Top 1), leads to large differences between TPPA and RRK in BLEU-4: 4.23% (8.14%-3.91%) for Wiz and 33.28% (45.94%-12.66%) for Holl-E data set. Despite that, TPPA manages to better approximate RRK than other models and improves response generation.

Analysis of Added Useful Words. In order to analyze the properties of the generated responses, we define two metrics to quantify: *useful word* and *useful word overlapping rate* (UWOR). If a word appears in the response but not in the post, it is *useful*. UWOR measures the coincidence ratio of two sequences and is defined as: $UWOR(p, r)$ = $overlap(p, r)$ / $distinct(r)$ for post p and response r. The $overlap(\cdot)$ is the number of distinct overlapping useful words between two sequences. $distinct(\cdot)$ is a distinct number of words. We remove the stop words of the two sequences before

Exp Name		Wizard of Wikipedia	Holl-E
UWOR(p, r)		14.6	7.52
UWOR$(k-p,r)$	BM25	4.11	9.42
	SKT	9.0	9.52
	CNN-DSSM	9.32	14.92
	TPPA	**10.25**	**15.98**
	rrk_1	34.52	67.84

Table 7: The useful word overlapping rate results of Wiz and Holl-E data sets. All values are shown as percentages (%).

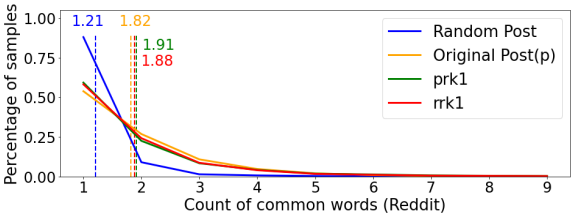

Figure 3: Common words count distribution between each source and the target response on the Reddit training set. The dashed lines are the average count of common words of each group (after removing stop words).

calculating UWOR.

We further test whether the retrieved knowledge brings additional useful words. We calculate *UWOR*$(k-p,r)$, where $k-p$ is a set of words in the knowledge ($k \in K_p$) but not in the associated post p, i.e., $\{w|w \in k \cap w \notin p\}$, w is the word of a sequence.

The results are shown in Table 7. For each experiment group in Table 7, we select the top 1 ranked sentence for calculation. *UWOR(p, r)* values for the Wiz and Holl-E data sets are just 14.6% and 7.52%, respectively. Considering the TPPA, for Wiz the number of additionally added useful words are comparable to what the post brings (10.25% vs. 14.6%); for the Holl-E, the retrieved knowledge brings more than double the useful words than the post (15.98% vs. 7.52%). This demonstrates the effectiveness of TPPA that can expand additional useful words from knowledge.

7 Conclusions and Discussions

Our investigations of the knowledge associated with post-response pairs lead us valuable insights into how well selected response-retrieved knowledge RRK can improve the performance of the generative models. Considering that response is not observable in the test phase, we developed a TPPA method that selects knowledge items by the careful embedding of the knowledge and optimized representation of the post and post-related knowledge PRK. We empirically demonstrate the superiority of TPPA, and being separated from the generative models. This provides flexibility to explore alternative components and models.

Despite its effectiveness, we now discuss one potential limitation of our TPPA model.We find that the quality of the knowledge base has a huge impact on the effectiveness of TPPA. The Wiz and Holl-E we experiment with are two data sets from which candidate knowledge items are of high quality and manually selected. As shown in Figure 1 for the Wiz dataset, rrk_1 group contains on average more than two common words than prk_1 group that would help to constitute the ground truth response. The same trend also holds for the Holl-E data set.

However, when looking at the Reddit data set[4], as shown in Figure 3, we find that rrk_1 group and prk_1 group almost contain the same number of common words, compared to the ground-truth response. This is not surprising given the nature of this dataset: Reddit is an online forum where each post is typically initiated with a URL to a web page (grounding) that defines the topic of the post, provided by the author. However, the repliers of the post might not read that information at all and respond according to their own knowledge. Empirically, we find TPPA can not benefit from the knowledge under this circumstance and perform worse than the baselines. This implies that when knowledge is potential of low quality, using PRK as the source of evidence for pseudo relevance feedback can result in potential topic drift.

In future work, we would like to (1) make TPPA more robust irrespective of the quality of provided knowledge; (2) develop an end-to-end model that directly model response generation with the help of response-related knowledge.

Acknowledgments

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.

[4]https://github.com/mgalley/DSTC7-End-to-End-Conversation-Modeling

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM.

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*.

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*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. *arXiv preprint arXiv:2002.07510*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. *arXiv preprint arXiv:1902.04911*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299. International World Wide Web Conferences Steering Committee.

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.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*, pages 232–241. Springer.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 101–110. ACM.

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.

Yik-Cheung Tam. 2020. Cluster-based beam search for pointer-generator chatbot grounded by knowledge. *Computer Speech & Language*, page 101094.

Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–236.

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.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64. ACM.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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.