Enhancing Conversational Dialogue Models with Grounded Knowledge

Wen Zheng
Wen.Zheng@nottingham.ac.uk
University of Nottingham
Nottingham, UK

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

ABSTRACT

Leveraging external knowledge to enhance conversational models has become a popular research area in recent years. Compared to vanilla generative models, the knowledge-grounded models may produce more informative and engaging responses. Although various approaches have been proposed in the past, how to effectively incorporate knowledge remains an open research question. It is unclear how much external knowledge should be retrieved and what is the optimal way to enhance the conversational model, trading off between relevant information and noise. Therefore, in this paper, we aim to bridge the gap by first extensively evaluating various types of state-of-the-art knowledge-grounded conversational models, including recurrent neural network based, memory networks based, and Transformer based models. We demonstrate empirically that those conversational models can only be enhanced with the right amount of external knowledge. To effectively leverage information originated from external knowledge, we propose a novel Transformer with Expanded Decoder (Transformer-ED or TED for short), which can automatically tune the weights for different sources of evidence when generating responses. Our experiments show that our proposed model outperforms state-of-the-art models in terms of both quality and diversity.

KEYWORDS

Sequence-to-Sequence; Copy-Mechanism; Memory Network; Multi-Task Learning; Transformer; Transformer-ED; TED; Generative Model; Knowledge-Grounded

ACM Reference Format:
Wen Zheng and Ke Zhou. 2019. Enhancing Conversational Dialogue Models with Grounded Knowledge. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, Paris, France, 10 pages. https://doi.org/10.1145/3357384.3357889

1 INTRODUCTION

Conversational dialogue systems have attracted attention from both academia and industry. Many products have been created, such as Google Assistant, Apple Siri, and Microsoft Cortana[1], most of which are task-oriented. For instance, Siri can be used to make calls and Cortana can operate your computer. Open-domain chat-bots, on the other hand, have also been developed, such as Microsoft XiaoIce[2].

Many of those conversational bots are developed through an end-to-end learning fashion [1]. However, the fully data-driven generative conversational models often struggle with bland and uninformative responses ([14, 27]). A human answers any question based on his/her own knowledge and it is impossible for one to answer a question correctly that he/she does not know. Analogue to this, the useful information for conversational responses is often stored in the external knowledge base. A piece of evidence for this argument can be found in Table 1, where the "question" refers to one utterance and the "target" denotes the response to that utterance[3]. The words tagged in red demonstrate that the background facts (external knowledge) contain certain information or vocabularies that do not appear in the question, which can be crucial for generating a reasonable response. We also observe that a model (e.g., Vanilla Transformer) without injecting facts is not able to generate a proper response, compared to a knowledge grounded solution (our proposed model, TED).

Recent studies [11, 22, 32, 38] have shown that effectively infusing useful background knowledge can produce better and more diverse responses. [32] builds two recurrent neural networks (RNN) based models to capture context information and shows that the hierarchical model performs better than the non-hierarchical one.

Table 1: The red tagged vocabulary in targets (responses) appear in facts, but not in the question (utterance) sequence. Incorporating the appropriate vocabularies from the facts, our proposed knowledge-grounded model (TED) is able to generate more superior responses than the model (Vanilla Transformer) that disregards facts.

Question: Do you know anything about the narcissus plant? **Target:** I do know that it is pretty much a spring perennial plant. **Fact:** A perennial plant or simply perennial is a plant that lives for more than two years. **Vanilla Transformer:** Controlled love n't that the is a much better lot and plant. **TED:** Deer do know that it is, much of weird perennial plant.
Question: Do you know when the Mustang was first made? **Target:** Yeah the original ford Mustang was manufactured in 1962! It was a two seater concept car. **Fact:** the ford Mustang is an American car manufactured by ford. **Vanilla Transformer:** Controlled, term formula Mustang was created by 1962. They was founded compact seater concept car. **TED:** Deer the modern civic Mustang was manufactured in 1962! It was a British seater concept car.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357889

[1] https://assistant.google.com/; www.apple.com/siri/; www.microsoft.com/cortana.
[2] https://www.msxiaobing.com/
[3] Those examples are originated from the real-world online conversations from Reddit (see Sec. 4.1 for details about the Reddit data).

Gaining benefits from the memory neural network (MemNN) and the multi-task learning method, [11] proposes a fully data-driven model to inject facts to questions, which achieves more diverse responses in the context of both Twitter and Foursquare. [5] employs Transformer and Universal Transformer model to apply multi-hop reasoning on the background knowledge. Through this method, it makes full use of knowledge from lowly ranked documents/facts of the retrieval process.

These three types of approaches (RNN based, MemNN based and Transformer based) form the state-of-the-art knowledge-grounded models. Although these models have been demonstrated to be useful for infusing knowledge in different contexts, there are no comprehensive evaluations that compare all of them. How to optimally select external knowledge is also unclear. In this paper, we aim to bridge the gap and answer the following research questions (RQs):

(1) **How do different types of state-of-the-art knowledge-grounded generative models perform?** We select typical approaches from all three types and conduct systematic experiments to assess their effectiveness in producing high quality and diverse responses.
(2) **How does the amount of knowledge affect the generative models' performance?** For different models, it is not clear how much knowledge (e.g., top 3, 10 or 20 sentences) should be retrieved to optimally enhance the conversational model, trading off between the relevant knowledge and noise.
(3) **Can we find an effective approach that optimally selects appropriate information from the utterance and the external knowledge?** By assuming that the utterance and the external knowledge contribute to the response in different ways, we propose a novel Transformer with Expanded Decoder (TED) model. Two additional functional modules, an attention-weighting layer and an attention-merging layer, are introduced based on the Transformer decoder architecture. Optimally tuning the weights to balance various sources of evidence, TED consistently outperforms previous models on two data sets, in terms of both quality and diversity.

To sum up, our contributions of the paper are three-fold:

- We propose a novel TED model[4] which introduces attention weighting and merging layers to effectively inject external knowledge to the generative model.
- We compare three types of knowledge-grounded generative models and empirically demonstrate that in addition to the proposed TED model, the hierarchical RNNs models perform better than the Multi-Task model.
- We show that enhancement from grounded knowledge can only be found on the right amount of knowledge (e.g., top 3 sentences for the TED model, and around top 12 sentences for the semantic representation hierarchical RNN model).

2 RELATED WORK

Overall, our work relates to generative models and knowledge-infusing methods in conversational dialogue.

[4]We make the codes publicly available at https://github.com/tonywenuon/Transformer_ED

Generative Models. Given a large-scale open-domain data set, many architectures have been proposed to learn to generate conversational responses. Basically, RNNs ([16]) have enjoyed the most popularity in the generative model community in the past. Boosting from the Long Short-Term Memory (LSTM, [13]) and the Gated Recurrent Units (GRUs, [3]), the Sequence-to-Sequence (Seq2Seq, [30]) model has become a popular generative model backbone. Not until attention mechanism [1] is proposed, Seq2Seq becomes omnipresent. The attention mechanism allows the decoder to look back to all the words in the encoder, taking their similarities as weights to incorporate the input sequences. [19, 37] adopt the Seq2Seq with attention mechanism as a generative model to achieve the machine translation task. Evolving from the Seq2Seq model, [25] introduces the copy mechanism to decide whether the current word is generated by the decoder or copied from the question. They observe that the generative mechanism tends to be more important than the copy mechanism when it encounters an uncertain situation, such as the beginning of the generative process. [31] builds a dual copy mechanism on top of [25] by incorporating external knowledge. Whenever the decoder generates a new word, it checks whether the word comes out from the generative model, the question or the external knowledge.

Different from the Seq2Seq model, the memory neural networks (MemNNs, [29]) are able to reason through the inference component, with accessing to the long-term memory. It is a practical model, building memories into a query vector from the query's and the memories' bag-of-words representation. An offline supervised method and an online reinforcement learning based method are adopted in the work [15], in which the authors use the MemNN as the generative model to predict an answer. They also build a MemNN binary classifier to decide which answer ought to be responded. Various prior studies [2, 21, 34] employ the MemNN to form End-to-End dialogue models.

Both of the attention based Seq2Seq model and the MemNN adopt attention to effectively select useful information. To make full use of the attention mechanism, the Transformer model ([33]) proposes the self-attention which calculates each word's representation by all the other words' representations. By introducing multi-head to the word representation, it computes attention on each head. In this way, when the model converges, the representation captures multi-relation between every two words. Some variant works are proposed on top of the vanilla Transformer model. [7] makes improvement by bringing adaptive computation time (ACT, [12]) which decides how many steps to run for a word representation automatically instead of the previously fixed number of steps. Transformer-XL ([4]) captures the longer-term dependency that the original Transformer can not incorporate and it is 1800+ times faster than the vanilla one.

Knowledge-Infusing Methods. Whereas the vanilla Seq2Seq model tends to generate generic responses such as "I am not sure what you mean" ([14]), many adaptive models are proposed for more informative and content-abundant responses. Retrieving background knowledge to add extra information to the question is one of the popular ways. [36] proposes a retrieve and refine model, retrieving top-ranked knowledge from the external knowledge base and refining the question by truncating the retrievals or using all of the retrievals. Akin to this, [38] builds two matrices on the word-level

similarity and semantic vector-level similarity as the foundation of a convolution neural network. Extra terms from the top-ranking documents of the retrieval model are expanded to the question. Another method for injecting knowledge is operated on the semantic vector-level. [32] summarises various vector representation merging methods, including summing and concatenating the question and the knowledge vectors. They conclude that hierarchical models are generally better than non-hierarchical models.

MemNN is initially designed for infusing memory to questions. [11] uses the MemNN model as an encoder after the retrieval process, and then leverages an RNN decoder for generating responses. Built on top of the MemNN, multi-task learning ([20]) is employed to improve the model's generalization ability, leaning shared decoder parameters jointly from several independent tasks. [18] injects personal information by using a person's non-conversational data (from Twitter) as the autoencoder and sharing parameters of the decoders in a two-tasks framework.

Since the Transformer and BERT ([9]) come out, many studies consider BERT as a pre-trained model. [39] adopts BERT, GloVe ([24]) and RNNs to build a complex architecture and gains the best results for now on the CoQA corpus. [10] takes the Transformer as its basic unit to jointly train a knowledge selection model and an utterance prediction model. A generative transformer memory network is built upon the joint-trained model. Inspiring by the Universal Transformer, [6] proposes a two layers model. A Transformer layer takes responsibility for generating a vector representation for each question and each retrieved document. A Universal Transformer layer is in charge of reasoning from the question and all of the retrieved documents.

Even though some previous works have been conducted to compare different RNN approaches on context-injecting models ([32]), they still lack a comprehensive evaluation of existing knowledge-infusing methods. It is also not clear what is the right amount of knowledge that should be injected. In this paper, we categorize existing methods into three categories: RNN based, MemNN based and Transformer based models; and adopt two types of metrics to evaluate them: quality and diversity. Our work shed light on existing knowledge-injecting models and provide guidance on choosing the appropriate amount of knowledge to infuse. We also propose a TED model that is able to more effectively incorporate the background facts. TED distinguishes itself from the previous Transformer models by introducing two functional modules to the decoder. It draws into trainable parameters as the external knowledge weights, which provides a reasonable way for weighting difference sources of evidence.

3 KNOWLEDGE-GROUNDED MODELS

We formulate the problem of infusing knowledge into generative models and provide an overview of current solutions.

3.1 Problem Formulation

Traditionally, the conversational dialogue models try to address the problem that given a question S_i, the model generates a response R_i. In this paper, we have a set of external knowledge that consists of a series of facts corresponding to the question.

The problem can be formulated as follows: given a conversational dialogue set $D = \{S_i, R_i\}_{i=1}^{N}$ (N is the total number of samples) and an external facts knowledge base F, we retrieve a fact list $\{f_i^1, f_i^2, \ldots, f_i^j\}$ based on each question S_i, where i is the i-th question, j refers to the j-th retrieved fact. Finally, we generate R_i as the response to S_i.

3.2 Overview of Approaches

Currently, there are several methods for incorporating background knowledge. We categorize them to three classes (see Table 2 for details).

A typical RNN-based approach is the Seq2Seq model. By adopting the attention mechanism, Seq2Seq models are able to make use of each term in the question. *Retrieve and refine* and *Pointer generator* ([36, 38]) are two Seq2Seq variant models. The former one retrieves external knowledge and extends the question with top-ranked words of the retrieval for refining. The latter model introduces a copy mechanism into the Seq2Seq model. The output words' probabilities come from the generative model and the question's words. The copy-mechanism, theoretically, can improve the diversity of the responses because the final word probabilities come from two sources and thus we choose this model as the typical RNN-based model. Even though LSTM or GRU can gain long-term memory, it is still difficult to tackle a very long sequence. *Generative hierarchical networks* ([26, 28]) adopt a traditional hierarchical recurrent encoder-decoder architecture (HRED, [28]) by combining with pre-trained embedding. It builds an utterance-level RNN layer on top of a term-level layer, by which the model can gain an interrelation between the facts and the question. The limitation of this approach lies in its inefficiency because it poses more computation in the model. The study [32] proclaims that their proposed hierarchical models perform better than other context-injecting methods, and thus we implement them in our work. These variant HRED models are represented as SR-Sum and SR-Concat in this paper.

On the other hand, *memory network* is able to reason between facts and the question, and to some extent handle long-term memory because it chooses memory by the semantic similarity between the question and facts. *Memory-to-sequence* ([22]) incorporates the pointer generator into an *end-to-end memory network* ([29]), potentially improving the memory network's performance. However, both of them fail to effectively distinguish noise from the truly useful information. Multi-task learning ([11, 18]) takes end-to-end memory network as a basic task to jointly train multiple tasks. When different tasks converge together, the shared parameters in the decoder can be affected implicitly by the model that injects facts. Like transfer learning, the three models can learn from each other. Because of these potential advantages, we chose it as the typical MemNN-related work to be implemented. It is also the baseline model of the DSTC-7 task.

Transformer based models manage to reason within a sequence by calculating the current word's attention with all of the other words of the sequence. As soon as it was proposed, it became the state-of-the-art model for Sequence-to-Sequence tasks. Nevertheless, the vanilla Transformer can not incorporate external knowledge. *Generative Transformer Memory Networks* ([10]) use two Transformer

Table 2: Three categories of knowledge-infusing models. '✓' means we implement this model in this paper.

Category		Methods	Advantage	Disadvantage
RNN Based		Retrieve and refine [36, 38]	Easy to implement	Limited question length
	✓	Pointer generator [25]	Flexible to add knowledge	Limited question and knowledge length
		Generative hierarchical network [26, 28]	Latent semantic operation	Computation costing
	✓	Hierarchical RNN [32] (refers to as **SR-Sum** and **SR-Concat**)	Latent semantic operation	Computation costing
MemNN based		Memory-to-sequence [22]	Memory saving	Noise unfriendly
		End-to-end memory network [29]	Memory saving	Noise unfriendly
	✓	MemNN and Multi-task learning [11, 18] (refers to as **Multi-Task**)	Learn from multi-tasks	Memory costing
Transformer based		Generative Transformer Memmory Networks [10]	Combine knowledge-selection	Inject only 1 knowledge
		SDNet [39]	Use all potential useful states	Too complex and computation costing
		Multi-hop Transformer [6]	Multi-hop reasoning	Memory costing
	✓	Transformer with Expanded Decoder (Our proposed model, refers to as **TED**)	Assign weights to facts	Memory costing

units to choose the best knowledge and generate a response simultaneously, while it can only inject one unit of external knowledge. *SDNet* ([39]) is a complex neural network architecture, which contains all of the history hidden states. Benefiting from all of the states, it generally achieves competitive performance. The limitation of this model lies in that it is very memory-consuming. The *Multi-hop Transformer* ([6]) method proposes a multi-hop reasoning layer which is based on Universal Transformer. It can incorporate multiple facts but just operate on the document level, not the token level. As a result, it is used to the factoid task such as SearchQA and Quasar-T data sets, not for open chitchat tasks that generate a sequence. Inspired by the previous works, we implement a novel Transformer with Expanded Decoder architecture, which operates on term level and is able to inject multiple external knowledge.

3.3 Fundamental Models

To be self-contained, we briefly introduce the fundamental models that will be used in our work. The Seq2Seq model will be used in the Pointer-Generator model and the hierarchical RNN model. The multi-task learning model employs the MemNN model and the Seq2Seq model as well. Our proposed TED employs the Transformer architecture.

Sequence-to-Sequence Model In the generative model community, the Seq2Seq model ([1]) is widely adopted as the baseline. Feeding a question S_i into the Seq2Seq model, the encoder encodes it to the vector representation (by the last hidden state of an RNN), and the decoder is used to interpret the vector to the target sequence R_i. Normally, the encoder and decoder are RNNs with LSTM cell or GRU cell. In this paper, we use the GRU cell.

End-to-End Memory Network Memory neural network (MemNN) is first proposed in [35]. [29] proposes a more practical version, the End-to-End Memory Networks. E2E MemNN is used as one of the injecting methods in this paper.

The facts set $\{f_i^1, f_i^2, \ldots, f_i^j\}$ that is related to a specific question S_i is available in our settings. Here, we consider the facts as memory. Each fact would be changed to a d-dimensional vector $\{m_i\}$ by f_i's embedding representation which is a trainable embedding matrix A($d \times V$, where V is the vocabulary size) ([29]). The same process will also be done to the question S_i, but with a different embedding matrix B (B has the same dimension as A) and then S_i will be converted to an internal state u_i. Following [11], we use t_i and r_i^j to represent the bag of words representation of S_i and f_i^j, and thus the MemNN can be formulated as follows.

$$u_i = Bt_i \tag{1}$$

$$m_i^j = Ar_i^j \tag{2}$$

$$c_i^j = Cr_i^j \tag{3}$$

$$p_i^j = \text{softmax}(u_i^T m_i^j) \tag{4}$$

$$o_i = \sum_{j=1} p_i^j c_i^j \tag{5}$$

$$\hat{u_i} = o_i + u_i \tag{6}$$

where, $A, C \in \mathbb{R}^{d \times V}$ should be trained in the MemNN. Originally, A and C are different matrices: A is the input embedding matrix for S_i while C is the output embedding matrix. We follow the "Adjacent" method shown in [29] where $A^{k+1} = C^k$ and $B = A^1$ (k is the k-th layer). p_i is a softmax similarity based on [29], and it is used to choose which part in the memory is most relevant to the question sequence. Then we employ o_i to summarise the potentially useful content in the memory, and finally, it is added to the question vector u_i. For more details about MemNN, we refer readers to the paper [29].

In comparison with the original MemNN model, we do not use MemNN to predict the target directly. Instead, we take the MemNN model as the encoder to generate the hidden states and then input it to an RNN decoder with the GRU cell. The facts are infused one by one to the question, which is memory-friendly and easy to be implemented.

Transformer The transformer is built upon the self-attention mechanism which calculates semantic similarities between the current word and all of the other words in the input sequence. Traditionally, the attention is computed by the dot product, but here in the Transformer, it uses a scaled dot product which means the dot product is divided by $\sqrt{d_k}$, where d_k is the dimension. The attention equation is shown below.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

where, Q, K, V are matrices which are a set of queries, keys and values respectively. Unique to the transformer, multi-head attention is employed. Rather than using d_k dimension to calculate the attentions, the multi-head attention separates d_k to several sub-dimensions. In this way, when generating a word representation, the multi-head attention makes it possible for each word to get access to several sub-spaces of the other words. After computing attentions at different heads, the Transformer concatenates all of the sub-space attentions again for the next step.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{8}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (9)$$

where W^O, W_i^Q, W_i^K, W_i^V are trainable parameters.

3.4 Hierarchical RNN Generative Model

Semantic Representation Summation and Concatenation Traditionally, through the RNNs, $(\{f_i^1, f_i^2, \ldots, f_i^j\}, S_i)$ are converted to an inter vector v_i which contains semantic information from the utterance-level. However, it fails to obtain the inter-relation between the facts and the question. This is where the hierarchical model comes out. We introduce two hierarchical models which are Semantic Representation Sum (**SR-Sum**) and Semantic Representation Concatenation (**SR-Concat**). Summing or concatenating the RNN hidden states of facts and the question and taking as input to another RNN. We build two hierarchical models ([32]) which are shown in Figure 1a and 1b. For each f_i^j and S_i, we obtain v_i^{fj} and v_i^s (the term-level layer's hidden states). In the inter-utterance level RNN, consecutively, they are turned to inter-semantic representations $(\{h_i^{f1}, h_i^{f2}, \ldots, h_i^{fj}, h_i^s\})$. The final encoder hidden states are summed and concatenated as follows.

$$v_i = \sum_{j=0}^{n} h_i^{fj} + h_i^s \quad (10)$$

$$v_i = [h_i^{f1}; h_i^{f2}; \ldots; h_i^{fj}; h_i^s] \quad (11)$$

Compared to the summing method, the concatenating method requires n (n is the facts number) times more RNN cells for the decoder, which results in large memory-cost and low computational efficiency.

3.5 Memory Networks based Infusing Method

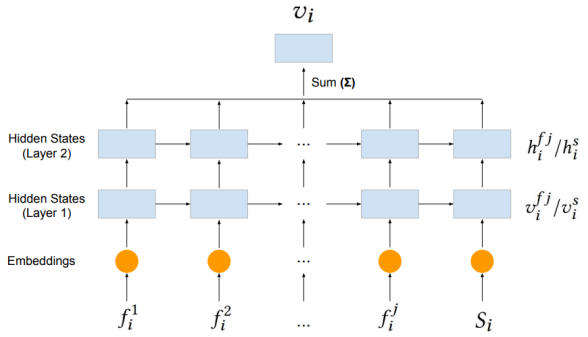
Multi-Task Learning Method To leverage the facts to the generative model, the multi-task learning simultaneously makes use of the facts, questions and responses by sharing decoder parameters. The facts for this task are unconstrained and are applicable for any topics. The multi-task, therefore, does not share the encoder parameters [18]. Its structure can be seen in Figure 2, which is a three-tasks model.

We introduce the following tasks from [11].

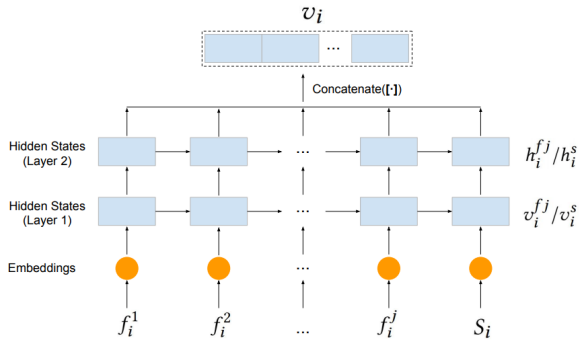
- No Facts Task: Taking (S, R) as training set samples;
- Facts Task: Taking $(\{f^1, f^2, \ldots, f^j; S\}, R)$ as training set samples;
- Autoencoder Task: Taking $(\{f^1, f^2, \ldots, f^j; S\}, f^j)$ as training set samples. This task has j times more samples than the Facts Task. We choose the f^1 which is the top ranking fact as the response to train the autoencoder because theoretically the f^1 is the most relevant knowledge, i.e. $(\{f^1, f^2, \ldots, f^j; S\}, f^1)$.

The autoencoder here helps to inject facts into the generative model indirectly by affecting the decoder parameters. For the three tasks, we choose different encoders that are shown below.

(1) RNN encoder: In the No Facts Task, we adopt a GRU cell for the RNN encoder.
(2) MemNN encoder: In both of the Facts Task and Autoencoder Task, we employ the MemNN as the encoder model.



(a) Hierarchical Summation



(b) Hierarchical Concatenation

Figure 1: Hierarchical RNN architecture: Hidden states summation and concatenation. v_i is the final encoder outputs.

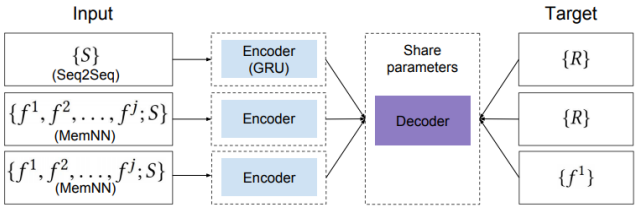


Figure 2: Multi-Task architecture. The dashed frame indicates a set of parameters and the decoder shares parameters between three different tasks.

3.6 The TED Model

Inspired by the Transformer, we propose a Transformer with Expanded Decoder model that is able to automatically tune different attention weights to the question and external knowledge. The overall framework is analogue to the Transformer architecture. The encoder part is the same as the vanilla Transformer. We expand the decoder module to fit with our requirements. Figure 3a illustrates the vanilla Transformer encoder and Figure 3b is the TED's decoder. In Figure 3b, 'Extra Info' means the external knowledge which contains all of the facts. 'Probabilities Layer (PL)' and 'Merge Attention Layer (MAL)' are two functional layers to incorporate extra information. The PL module takes responsibility for generating 'Weights Parameters' that can automatically learn weights

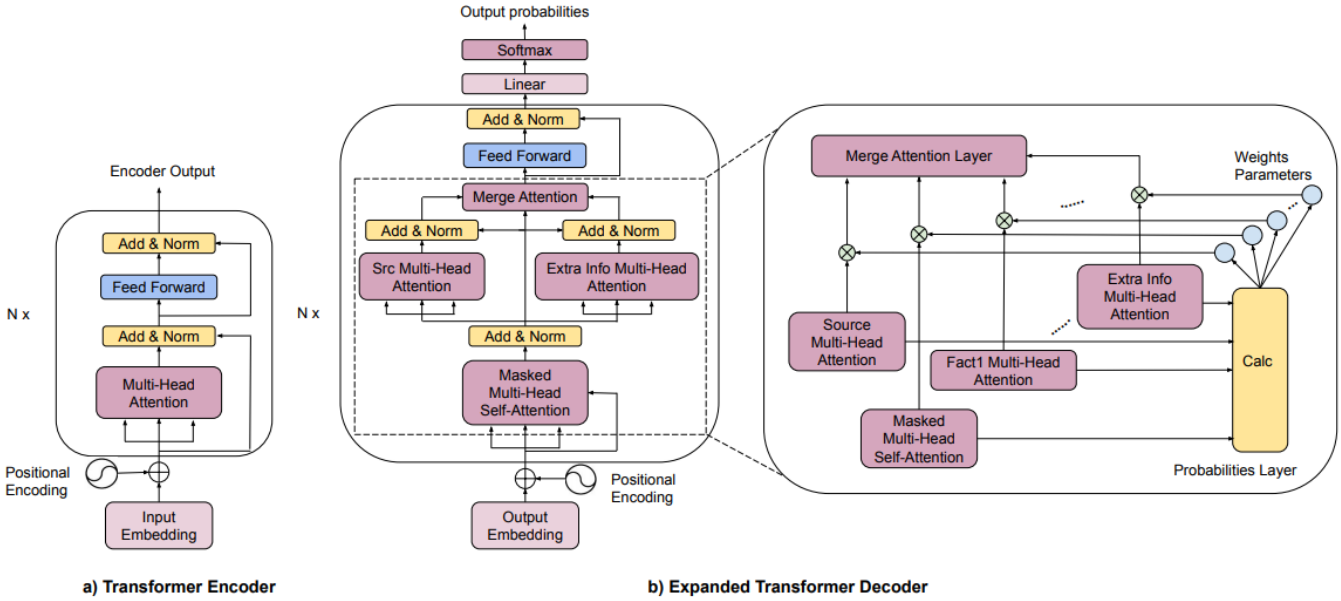


Figure 3: The TED model: a) a Transformer encoder (the same as the vanilla Transformer); b) an expanded decoder. The right-hand side sub-figure is the detailed inner modules. For simplification, some modules are omitted in the right-hand side sub-figure. In b), the 'Extra Info' stands for the external knowledge and the 'Calc' means the probabilities calculating layer.

for each additional information. The parameters are successively inputted into the MAL module to be merged. We provide more detailed formulations as below.

Word-Level Attention As shown in Figure 3a, the encoder part is the same as the vanilla Transformer. We conduct word-level Transformer process for the question S and facts $(\{f^1, f^2, \ldots, f^j\}$. For each word in the sentences, we use the trainable embeddings rather than using certain pre-trained embedding models (e.g., [39] uses the GloVe).

Inner-Attentions Multi-Head attentions between the *question and decoder inputs*, and the *facts and decoder inputs* are incorporated in the decoder part (Figure 3b). The motivation here is that we assume that the question and facts exert influences on the generated word separately. The formulas for these multi-head mutual attentions are shown below.

$$v_{S\_dec} = Attention(v_{dec}, v_S, v_S) \tag{12}$$

$$v_{f\_dec} = Attention(v_{dec}, v_f, v_f) \tag{13}$$

where v_{S_dec} and v_{f_dec} means multi-head attentions between S, f and the decoder inputs respectively. v_S, v_f and v_{dec} are their self-attention representations.

Expanded Transformer Decoder Each Transformer decoder, originally, has a self-attention layer, a multi-head mutual attention layer, and a feed-forward layer. Here in the TED, we have two extra functional layers: PL and MAL, as shown in the right-hand side of Figure 3b. After we get inner-attentions, v_{S_dec} and v_{f_dec}, they are inputted into the PL to get the weights parameters P which will be multiplied with v_{S_dec}, v_{f_dec} and v_{dec} itself. These weights are trainable parameters in the PL.

$$P = W_S * v^T_{S\_dec} + \sum W_{f_j} * v^T_{f_j\_dec} + W_{dec} * v^T_{dec} + b \tag{14}$$

where W_S, W_{f_j}, W_{dec}, and b are trainable parameters. It is flexible to inject several facts because the weights parameters here are calculated automatically. The MAL merges different weighted attentions as follows:

$$v_{merge} = p_S * v_{S\_dec} + \sum p_{f_j} * v_{f_j\_dec} + p_{dec} * v_{dec} \tag{15}$$

Here, p_S, p_{f_j} and p_{dec} come from the Equation 14. The MAL is just a functional layer for merging different attentions together.

4 EXPERIMENTS

4.1 Data sets

We use two data sets for our experiments: Reddit and Wizard of Wikipedia. All of them contain a set of questions, answers and background knowledge. Given that the background knowledge provided in the original Reddit dataset is very noisy, we also obtain a filtered Reddit dataset to evaluate only the extent when the provided background knowledge contains useful information. The statistics of the datasets are shown in Table 3.

Wizard of Wikipedia Data Set Wizard of Wikipedia is a data set from the Facebook AI Research ([10]). It consists of a set of questions, answers and the retrieved background knowledge. The author crowd-sourced a diversity of 1431 natural, open-domain conversational topics. The human takes the roles of the Apprentice and the Wizard. The Wizard answers the Apprentice's questions based on either the retrieved knowledge of the last utterance or

Table 3: Statistics of the Reddit and the Wizard of Wikipedia data sets.

Data set	Train	Test
Wizard of Wikipedia	83245	3865
Original Reddit	1482419	13440
Filtered Reddit	91344	5000

his/her own knowledge, which means all of the answers are generated by humans. In total, this data set has 83245 training samples and 3865 for testing. We take this data set as our main data set for experiments because of its high quality. The data set can be obtained from ParlAI[5].

Original Reddit Data Set This conversational data set is released by Dialog System Technology Challenges 7 (DSTC-7) and is extracted from the Reddit. For each web page of the original data set, there is a link below the title, which might provide background knowledge for the current topic. We download the data from Reddit dump and Common Crawl [6] as our experiment data, following DSTC-7[7]. There are more than 1.4 million samples in the training set, and 13440 samples in the test set.

Filtered Reddit Data Set We find that the external link of Reddit is very noisy and many of those external knowledge articles do not contain any useful information. To measure quantitatively the noise level, we define 'useful-words-rate' (*UWR*) as our metric. After deleting stop words, if a word appears in both of the target and the corresponding facts (external knowledge article) but does not appear in the question, we deem a word as useful. *UWR* is simply defined as the frequency of useful words, divided by the total number the questions. Note that if the same useful word appears n times in the facts, we deem the number of useful words as n. Empirically, we find that the *UWR* of Wizard of Wikipedia dataset is 193%, while it is only 46% for the Original Reddit data set. To evaluate only the extent when the background knowledge article contains useful information, we generate a filtered subset from the Original Reddit data set by only selecting the top Reddit posts based on the *UWR* metric. We obtain 91344 samples for training and 5000 for testing for this Filtered Reddit data set.

4.2 Metrics

Metrics For our experiments, we report a series of standard metrics for this task: Meteor score ([8]), BLEU scores ([23]) and Distinct scores (refers to as Div-1 score and Div-2 score) ([14]). Among them, the BLEU scores calculate the similarity between two sentences by computing the co-occurrence n-gram words frequency. It has many advantages: convenient, fast, and its results have a positive reference value. Meteor introduces external synonym sets, considering the stemmed form of the terms at the same time. According to [17], the BLEU-2 is the closest metric to the human evaluation. Therefore, we take the BLEU-2 score as our main metric for evaluating relevance. Differing from the BLEUs and the Meteor score, the Distinct scores measure the diversity level. For a given sequence, the Distinct scores are distinct unigrams (Div-1) and bigrams (Div-2) divided by the total number of the generated words.

4.3 Experiments Setup

For the data sets, we adopt a similar pre-processing step. We filter the samples by the response sentence length which is limited from 8 to 20 terms (following [31]). 50k vocabulary is chosen by the ranking of the term frequency and it is shared between the questions and the targets. For the Reddit data set, after doing statistical analysis of the length distribution of the questions and the targets, we set the maximum question length to 100, maximum fact length to 100 and maximum target length to 30, which can cover more than 80% sequence lengths of the questions, facts and targets. On the other hand, in terms of the Wizard of Wikipedia data set, to cover 80% sample lengths, the maximum question, target and fact length are set to 30, 30, 30 respectively.

To retrieve potential relevant facts to infuse into the model, we use different retrieval methods in the training and testing processes. In the training process, we use an 'oracle retrieved fact set'. That is, firstly we select 'useful words' (mentioned in Sec. 4.1) to be as the query for each question. And then all of the facts are retrieved by this query and sorted by the number of useful words. By doing this, the top-ranked facts are more likely to contain useful words that can benefit the model through the training process. When it comes to the testing, we adopt a simple TF-IDF weighting scheme to retrieve from all the facts by ranking based on the similarity score between the question (as a query) and facts (as documents).

As for the models, we follow [11] and build a 3-layers gated GRUs Seq2Seq model. 100 hidden state dimension is set for each layer. The encoder and decoder have the same state dimension. The word embedding's dimension is also set to 100. We use Adam optimizer with a learning rate of 0.001. In terms of the MemNN, 100 is given to the memory embedding size and hop steps is set to 3. For the Transformer models, we keep the default settings[8]. The Transformer encoder and decoder stack numbers are set to 3. 4 heads and 100 attention dimension are given for the multi-head attention.

Based on the experimental setup, we conduct all of the experiments on a single GPU (GeForce GTX 1080). For the Original Reddit data set, the TED takes 2 days for training, while on the Filtered Reddit data set and the Wizard of Wikipedia data set, it takes around 3 hours for training. When predicting, if an end token is predicted or the maximum length is reached, the process terminates.

4.4 Experimental Results

4.4.1 Analysis of Knowledge-Infusion Models. Table 4 presents the performance of all models on the Filtered Reddit and the Wizard of Wikipedia. This represents the scenario when the background knowledge contains potentially useful information and is less noisy. Looking at the Table 4, there are two groups to discuss: (1) RNN-based models vs. MemNN-based models; (2) Transformer-based models vs. Other models. RNN-based models mainly include the Pointer-Generator model and the hierarchical RNN model. The MemNN-based models contain the multi-task learning memory network. What's more, the Seq2Seq-Attn model, the Pointer-Generator

[5]http://parl.ai
[6]http://files.pushshift.io/reddit/comments/, http://commoncrawl.org/
[7]https://github.com/mgalley/DSTC7-End-to-End-Conversation-Modeling/tree/master/data_extraction
[8]https://github.com/kpot/keras-transformer

Table 4: For the Filtered Reddit and Wizard of Wikipedia data sets: Compare different types of knowledge-grounded models on 7 metrics. We take the TED as the base model to do the significant test. * and ** stand for significant test value, $p < 0.05$ and 0.01 respectively

Data sets	Filtered Reddit data set							Wizard of Wikipedia data set						
Metrics	BLEU-1(%)	BLEU-2(%)	BLEU-3(%)	BLEU-4(%)	Meteor(%)	Div-1(%)	Div-2(%)	BLEU-1(%)	BLEU-2(%)	BLEU-3(%)	BLEU-4(%)	Meteor(%)	Div-1(%)	Div-2(%)
Seq2Seq-Attn	7.59**	2.06**	0.66**	0.2**	3.7**	0.08**	0.17**	12.73**	4.41**	1.89**	0.9**	4.6**	0.3**	0.89**
Pointer-Generator	11.73*	3.55**	1.84**	0.49**	2.8**	2.0**	4.0**	17.51**	7.08**	3.45**	1.86**	6.8**	2.8**	8.3**
SR-Sum	**15.34****	4.39**	1.39**	0.48**	5.39*	0.49**	1.43**	18.95**	7.69**	3.68**	1.94**	6.98**	3.79**	12.96**
SR-Concat	15.25**	4.2**	1.28**	0.45**	5.15**	0.6**	1.75**	18.56**	7.07**	3.35**	1.81**	6.47**	3.84**	12.7**
Multi-Task	14.73	4.16**	1.25**	0.43**	5.04**	0.13**	0.36**	18.51**	6.78**	2.9**	1.41**	6.26**	2.4*	8.0*
Transformer	11.95**	3.04**	0.91**	0.35**	4.15**	0.04**	0.06**	18.09**	7.15**	3.38**	1.76**	6.6**	2.1**	7.3**
TED	14.18	**5.01**	**2.11**	**0.95**	**5.6**	**2.2**	**9.4**	**20.27**	**9.47**	**5.33**	**3.35**	**8.45**	**3.95**	**16.2**

model and the Transformer model are three models that do not infuse any knowledge.

RNN-based models vs. MemNN-based models Two consistent patterns can be observed from Table 4. First, as the main performance evaluation metrics, the BLEU-2 and Meteor scores are higher on almost all of the knowledge-injecting models even though for the Wizard of Wikipedia, the Pointer-Generator model performs slightly better than the Multi-Task model. That means that injecting external knowledge to questions helps to improve the performance. This conclusion is consistent with the previous works [11, 18, 32, 36]. Second, from the diversity perspective, the SR models and the multi-task model get relatively better results on the Wizard of Wikipedia data set, but on the Filtered Reddit data set, they are worse than the Pointer-Generator model. This indicates that the knowledge-injecting models are generally sensitive to the data set. The Wizard of Wikipedia data set which is generated by humans has a high quality in comparison with the Filtered Reddit data set.

In terms of knowledge-infusing models, two SR models perform similar on the two data sets and are generally better than other RNN-based and MemNN-based models. The Multi-Task model does not beat the SR models but shows better performance than models without infusing knowledge. This may be explained by its parameters sharing mechanism. Just like the transfer learning, when the three tasks are trained together, each of them learns parameters from itself and the other two models. This mechanism makes it possible to learn new extra information from the other models.

Transformer-based models vs. Other models On both of the Filtered Reddit and Wizard of Wikipedia data set, compared to other models without infusing knowledge, the Transformer model outperforms the Seq2Seq-Attn model but perform worse than the Pointer-Generator model. In terms of diversity on the Filtered Reddit data set, Transformer is the worst.

As for our proposed TED model, we can see that except for the BLEU-1 score on the Filtered Reddit data set, almost all of the metrics improve significantly on both of the two data sets. As the anecdotal examples presents (Table 1), our proposed TED model is able to make use of the background facts to generate an informative answer. We conduct significant tests (two-tailed student t-test) on all metrics and it reveals that the TED improves performance significantly in terms of both quality and diversity.

4.4.2 Analysis of The Right Amount of Knowledge. We obtain the top 2 performing models (i.e., TED and SR-Sum) on the Wizard of Wikipedia data set from Table 4 and analyze for both models the optimal amount of external knowledge to infuse by experimenting with injecting the different number of top retrieved facts. The comparative results on the Wizard of Wikipedia data set for TED and SR-Sum are respectively shown in Table 5 and Table 6.

In terms of quality (measured by Meteor and BLEU), we can observe that for both models, the performance first increases when injecting more facts (e.g., top-1-3 sentence for TED). This demonstrates that obtaining more relevant knowledge is beneficial for improving response generation. However, the performance plateaus when reaching a certain number of facts (12 for the SR-Sum model, 3 for the TED). This might be due to that when infusing more facts into the question, it incorporates useful information as well as more noise. As a result, the performance drops when more noise are incorporated. In terms of diversity, it also shows a similar trend to the quality, although the point of the plateau may differ. For example, it can be witnessed in Table 6 that SR-Sum model gets the highest BLEU scores on 12-facts, but the highest diversity on 14-facts.

Note that we also compare the best performing 3-facts TED and the 12-facts SR-Sum model, while the former is significantly better than the latter. To summarize, on the Wizard of Wikipedia data set, the proper injecting fact number for the SR-Sum model is 12, while for the TED, the best number is 3. We demonstrate empirically that those models can only be enhanced with the right amount of external knowledge.

4.4.3 The Impact of Noisy External Knowledge. In Section 4.1, we have illustrated that the *UWR* of the Wizard of Wikipedia data set is much higher than that of the Original Reddit data set. We also demonstrate in our previous section that the amount of injected knowledge can be crucial while more noisy knowledge can degrade the model performance. Next, we further demonstrate whether noisy external knowledge has any impacts on various models.

Table 7 present the performance of all models on the noisy Original Reddit dataset[9]. We can observe that when the injected facts are noisier, the proposed TED does not always obtain the best results. Especially the diversity score, it is worse than the Pointer-Generator and the Multi-Task model. Even though in terms of BLEU-3 and BLEU-4 scores, the TED model performs the best, it is not significantly better than the Pointer-Generator model. This demonstrates that comparatively speaking, Point-Generator is more robust to noise, compared to our proposed TED approaches, especially from

[9]Due to the fact that the Original Reddit data set adopts multi-reference as ground-truths (as in DSTC-7 evaluation scripts), the metric values are generally higher than the other datasets with only one reference. Therefore, the absolute values of different metrics are not comparable across different datasets.

Table 5: Injecting more external knowledge to the TED on the Wizard of Wikipedia data set. The TED gets a peak performance at 3 facts, so we take 3-fact model as the base model to do the significant test. * and ** stand for significant test value, $p < 0.05$ and 0.01 respectively.

Method	TED										
Fact numbers	1	2	3	4	5	10	11	12	13	14	15
BLEU-1	20.27*	20.25*	**20.28**	20.02	20.23	20.16	20.05	20.07	20.36	20.21	20.36
BLEU-2	9.47	8.94**	**9.62**	8.96*	9.09*	8.88*	8.94*	8.95*	8.88**	8.98*	9.13
BLEU-3	5.33	4.64**	**5.5**	4.78**	4.83**	4.56**	4.72**	4.64**	4.51**	4.63**	4.76**
BLEU-4	3.35	2.63**	**3.53**	2.81**	2.86**	2.56**	2.74**	2.65**	2.51**	2.61*	2.77**
Meteor	**8.45****	7.73**	8.04	7.71*	7.71*	7.78	7.82	7.7**	7.7*	7.75**	7.8
Div-1	3.95**	3.55**	**5.46**	4.09**	4.0**	3.73**	4.14**	3.8**	3.3**	3.53**	3.5**
Div-2	16.2**	15.2**	**22.2**	18.0**	18.76**	16.3**	17.4**	16.0**	13.7**	14.7**	14.4**

Table 6: Injecting more external knowledge to the SR-Sum model on the Wizard of Wikipedia data set. The SR-Sum model gets a peak performance at 12 facts, so we take 12-fact model as the base model to do the significant test. * and ** stand for significant test value, $p < 0.05$ and 0.01 respectively.

Method	SR-Sum										
Fact numbers	1	2	3	4	5	10	11	12	13	14	15
BLEU-1	18.95**	19.18**	18.91**	19.13**	19.21**	19.5	19.12**	**19.81**	18.9**	19.24	19.36**
BLEU-2	7.69**	7.9**	7.68**	7.87**	7.96*	8.29	8.1*	**8.39**	7.91*	8.23	8.16
BLEU-3	3.68**	3.82**	3.64**	3.83**	3.85*	4.13	3.97	**4.14**	3.85*	4.14	3.91
BLEU-4	1.94**	2**	1.91**	2.05*	2.05*	**2.31**	2.11	2.24	2.07	2.33	2.09
Meteor	6.98**	7.0**	6.9**	7.07**	7.2**	7.44	7.27*	**7.49**	7.31	7.46	7.35
Div-1	3.79	4.6	4.9**	4.78**	5.09**	5.4	5.29	5.35	5.45**	**5.63****	5.3
Div-2	12.96	15.8	16.75**	16.75**	17.59**	19.48	18.71	19.24	19.46	**19.96***	18.96

Table 7: For the Original Reddit data set: Compare different types of knowledge-grounded models on 7 metrics. We take the TED as the base model to do the significant test. * and ** stand for significant test value, $p < 0.05$ and 0.01 respectively.

Metrics	BLEU-1(%)	BLEU-2(%)	BLEU-3(%)	BLEU-4(%)	Meteor(%)	Div-1(%)	Div-2(%)
Seq2Seq-Attn	34.79**	8.62**	2.39**	0.59**	6.7**	0.49**	1.28**
Pointer-Generator	**37.49**	**12.8****	4.52	1.72	**8.76***	**6.2****	**15.2****
SR-Sum	31.6**	8.52**	2.17**	0.63**	7.7**	0.7**	2.0**
SR-Concat	32.07**	8.67**	2.39**	0.65**	7.9**	1.0**	2.8**
Multi-Task	34.49**	8.98**	2.53**	0.91**	7.1**	1.5	5.29**
Transformer	34.79**	8.62**	2.39**	0.59**	6.7**	0.49**	1.3**
TED	37.36	11.89	**4.55**	**1.89**	8.4	1.1	3.8

the diversity perspective. Our proposed TED model performs the best when infusing only a small amount of relevant knowledge.

5 CONCLUSIONS

There exist various approaches to infuse external knowledge to the generative models. We conduct a set of systematical experiments on the Reddit data set and the Wizard of Wikipedia data set to evaluate their effectiveness. We first categorize current state-of-the-art knowledge-injecting models to three classes: RNN-based model, MemNN-based model and Transformer-based model. With respect to existing models, we find that (1) the knowledge-injecting models perform generally better than the models that do not infuse knowledge; (2) when comparing the RNN-based with the MemNN-based models, the hierarchical RNN models perform significantly better on both of the relevance and diversity; (3) the knowledge-infusing models are generally sensitive with the noise within the knowledge, which implies that high-quality knowledge can help to achieve better performing models.

In addition to reviewing existing models, we propose a novel Transformer with Expanded Decoder model (TED), a method for incorporating extra information to the questions. TED is built on top of the vanilla Transformer and empowers the architecture with weights tuning across different sources of evidence, which are trainable parameters for the question and facts representations to the decoder. We have illustrated the TED model is a highly effective knowledge-infusing model, especially with the small amount of high-quality knowledge.

We intend to explore the following potential directions to further improve the effectiveness of our knowledge-infusing models. First of all, we have demonstrated that most of the current models are sensitive to the noise levels of injected knowledge. We would like to explore from the model perspective various ways to automatically select relevant and high-quality words to cope with the noise. Besides, our approach is an end-to-end generative model with relevant knowledge retrieved using the simple TF-IDF ranking function on sentences. We can not only explore more advanced retrieval methods to retrieve highly relevant knowledge for the generative model but also further establish principled ways to combine retrieval and generative models (e.g., a retrieve and refine model [36]).

ACKNOWLEDGMENTS

This work is partly supported by Engineering and Physical Sciences Research Council (EPSRC Grant No. EP/S515528/1, 2102871). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. http://arxiv.org/abs/1409.0473 cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[2] Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2017. Learning End-to-End Goal-Oriented Dialog.. In *ICLR*. OpenReview.net. http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#BordesBW17

[3] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[4] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv e-prints* (Jan. 2019). arXiv:1901.02860

[5] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 681–689.

[6] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019. Learning to Transform, Combine, and Reason in Open-Domain Question Answering.. In *WSDM*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 681–689. http://dblp.uni-trier.de/db/conf/wsdm/wsdm2019.html#DehghaniAKR19

[7] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal Transformers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HyzdRiR9Y7

[8] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*. 376–380.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] 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*. https://openreview.net/forum?id=r1l73iRqKm

[11] 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*.

[12] A. Graves. 2016. Adaptive Computation Time for Recurrent Neural Networks. *arXiv e-prints* (March 2016). arXiv:1603.08983

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *HLT-NAACL*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). The Association for Computational Linguistics, 110–119. http://dblp.uni-trier.de/db/conf/naacl/naacl2016.html#LiGBGD16

[15] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through Dialogue Interactions by Asking Questions.. In *ICLR*. OpenReview.net. http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#LiMCRW17a

[16] Zachary Chase Lipton. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *CoRR* abs/1506.00019 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1506.html#Lipton15

[17] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael 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. *arXiv preprint arXiv:1603.08023* (2016).

[18] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-Task Learning for Speaker-Role Adaptation in Neural Conversation Models.. In *IJCNLP(1)*, Greg Kondrak and Taro Watanabe (Eds.). Asian Federation of Natural Language Processing, 605–614. http://dblp.uni-trier.de/db/conf/ijcnlp/ijcnlp2017-1.html#LuanBDGG17

[19] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206* (2014).

[20] Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task Sequence to Sequence Learning. In *International Conference on Learning Representations*.

[21] Fenglong Ma, Radha Chitta, Saurabh Kataria, Jing Zhou, Palghat Ramesh, Tong Sun, and Jing Gao. 2017. Long-Term Memory Networks for Question Answering. *CoRR* abs/1707.01961 (2017). http://dblp.uni-trier.de/db/journals/corr/corr1707.html#MaCKZRSG17

[22] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1468–1478. https://www.aclweb.org/anthology/P18-1136

[23] 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 on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[25] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1073–1083. https://doi.org/10.18653/v1/P17-1099

[26] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[27] Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models.. In *EMNLP*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 2210–2219. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2017.html#ShaoGBGSK17

[28] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 553–562.

[29] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. 2440–2448.

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[31] Yik-Cheung Tam, Jiachen Ding, Cheng Niu, and Zhou Jie. [n. d.]. Cluster-based Beam Search for Pointer-Generator Chatbot Grounded by Knowledge. ([n. d.]).

[32] 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)*, Vol. 2. 231–236.

[33] 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*. 5998–6008.

[34] Anran Wang, Anh Tuan Luu, Chuan-Sheng Foo, Hongyuan Zhu, Yi Tay, and Vijay Chandrasekhar. 2018. Holistic Multi-modal Memory Network for Movie Question Answering. *CoRR* abs/1811.04595 (2018). http://dblp.uni-trier.de/db/journals/corr/corr1811.html#abs-1811-04595

[35] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. *MEMORY NETWORKS*. Technical Report. arXiv:1410.3916v11 https://arxiv.org/pdf/1410.3916.pdf

[36] 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*. Association for Computational Linguistics, Brussels, Belgium, 87–92. https://www.aclweb.org/anthology/W18-5713

[37] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ÅĄukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* abs/1609.08144 (2016). http://arxiv.org/abs/1609.08144

[38] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems.. In *SIGIR*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 245–254. http://dblp.uni-trier.de/db/conf/sigir/sigir2018.html#YangQQGZCHC18

[39] C. Zhu, M. Zeng, and X. Huang. 2018. SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. *arXiv e-prints* (Dec. 2018). arXiv:cs.CL/1812.03593