Response-Anticipated Memory for On-Demand Knowledge Integration in Response Generation

Zhiliang Tian,[*1,4] **Wei Bi,**[†2] **Dongkyu Lee,**[1] **Lanqing Xue,**[1]
Yiping Song,[3] **Xiaojiang Liu,**[2] **Nevin L. Zhang**[1,4]
[1]Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong SAR, China
[2]Tencent AI Lab, Shenzhen, China
[3]Department of Computer Science School of EECS, Peking University, Beijing, China
[4]HKUST Xiao-i Robot Joint Lab, Hong Kong SAR, China
{ztianac,dleear,lxueaa,lzhang}@cse.ust.hk
{victoriabi,kieranliu}@tencent.com songyiping@pku.edu.cn

Abstract

Neural conversation models are known to generate appropriate but non-informative responses in general. A scenario where informativeness can be significantly enhanced is Conversing by Reading (CbR), where conversations take place with respect to a given external document. In previous work, the external document is utilized by (1) creating a context-aware document memory that integrates information from the document and the conversational context, and then (2) generating responses referring to the memory. In this paper, we propose to create the document memory with some anticipated responses in mind. This is achieved using a teacher-student framework. The teacher is given the external document, the context, and the ground-truth response, and learns how to build a response-aware document memory from three sources of information. The student learns to construct a response-anticipated document memory from the first two sources, and the teacher's insight on memory creation. Empirical results show that our model outperforms the previous state-of-the-art for the CbR task.

1 Introduction

Neural conversation models have achieved promising performance in response generation. However, it is widely observed that the generated responses lack sufficient content and information (Li et al., 2016a). One way to address this issue is to integrate various external information into conversation models. Examples of external information include document topics (Xing et al., 2017), commonsense knowledge graphs (Zhou et al., 2018), and domain-specific knowledge bases (Yang et al., 2019). Conversing by reading (CbR) (Qin et al., 2019) is a recently proposed scenario where external information can be ingested to conversations. In CbR, conversations take place with reference to a document. The key problem in CbR is to learn how to integrate information from the external document into response generation on demand.

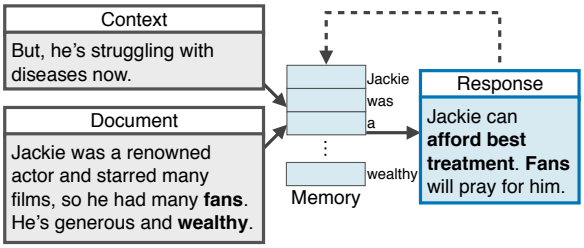

Figure 1: A motivating example of constructing a response-anticipated document memory for response generation. Details are provided in the introduction.

To exploit knowledge from documents for conversations, a conventional way is to extend the sequence-to-sequence (Seq2Seq) model (Sutskever et al., 2014) with Memory Networks (Sukhbaatar et al., 2015), which store knowledge representations accessible to their decoder (Ghazvininejad et al., 2018; Parthasarathi and Pineau, 2018). Dinan et al. (2018) propose to encode the dialogue context as well as a set of retrieved knowledge by Transformer (Vaswani et al., 2017) to construct the memory. However, these methods only use sentence-level representations of the documents in the memory, which cannot pinpoint accurate token-level document information.

To discover token-level document information, researchers borrow models from other generation tasks, which are adept at extracting segments of sentences for given questions. Moghe et al. (2018) explore the pointer generator network (See et al., 2017) for abstractive summarization and the bi-directional attention flow model (Seo et al., 2017), which is a QA model to predict a span of the

[*]This work was partially done when Zhiliang Tian was an intern at Tencent AI Lab.
[†]Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 650–659
July 5 - 10, 2020.

document to be contained in the response. Qin et al. (2019) follow the stochastic answer network (SAN) (Liu et al., 2018) in machine reading comprehension (MRC), integrating both context and document information to form the context-aware document memory. This approach obtains the state-of-the-art performance on the CbR task.

However, we should notice the difference between existing generation tasks and CbR. For summarization, QA, and MRC, they require models to extract exact answers from documents, where documents cover all requisite knowledge. Meanwhile, CbR expects to output a general utterance relevant to both context and document. As the example in Fig. 1, the document refers to *actor, films, fans, wealthy* and the context mentions *disease*. Document and context discuss the same person but have no topic overlap; thus we cannot pinpoint document information from the context. If we use SAN as in Qin et al. (2019), SAN can hardly acquire helpful information from context-document interaction. To ingest useful knowledge for response generation, we argue that processing documents should consider not only the interaction between context and document but also the target response. As in the example, the document should attend more on *fans, wealthy* by considering the response.

In this work, we propose a method to construct a response-anticipated memory to contain document information that is potentially more important in generating responses. Particularly, we construct a teacher-student framework based on Qin et al. (2019). The teacher model accesses the ground-truth response, context, and document. It learns to construct a weight matrix that contains information about the importance of tokens in the document to the response. The student model learns to mimic the weight matrix constructed by the teacher without access to the response. That is, the teacher learns to build a response-aware memory, while the student learns to build a response-anticipated memory. During inference on testing data, the student will be applied. Our experiments show our model exceeds all competing methods.

2 Related Work

Most neural conversation models in open domain chit-chat scenarios are based on the Seq2Seq model (Sutskever et al., 2014; Shang et al., 2015). A critical issue of these models is the safe response problem, i.e., generated responses often lack enough content and information. To address this issue, previous work encourages response diversity and informativeness by introducing new training objectives (Li et al., 2016b; Zhao et al., 2017), refining beam search strategies (Li et al., 2016a; Vijayakumar et al., 2018; Song et al., 2017), exploiting information from conversational contexts (Serban et al., 2016, 2017; Tian et al., 2017), or incorporating with retrieval-based conversation systems (Song et al., 2018; Wu et al., 2019b; Tian et al., 2019).

Some researchers augment information in generating responses by external resources. Zhou et al. (2018) utilize the commonsense knowledge graph by their designed graph attention. Agarwal et al. (2018) propose a knowledge encoder to encode query-entity pairs from the knowledge base. Wu et al. (2019a) enrich response generation with knowledge triplets. These work all uses knowledge information in structured formats.

External unstructured text information has also been investigated to improve conversation models. Some researchers directly build "document memory" by using distributed representations of the knowledge sentences into conversation models (Ghazvininejad et al., 2018; Parthasarathi and Pineau, 2018). Dinan et al. (2018) make use of the Transformer (Vaswani et al., 2017) to encode the knowledge sentences as well as the dialogue context. Ren et al. (2020) design a knowledge selector to construct the document memory on selective knowledge information. As stated in the introduction, some other researchers borrow models from other generation tasks, including abstractive summarization models (Moghe et al., 2018), QA models (Moghe et al., 2018) and MRC models (Meng et al., 2020; Qin et al., 2019). Especially, Qin et al. (2019) get the state-of-the-art performance. However, they all construct the document memory relying on connections between context and document without consideration of the response. If context or document contains a lot of noise tokens irrelevant to the response, which is indeed the case in CbR, the constructed memory may be misled by these noise information (as the case in Fig. 1). Therefore, we propose to involve the consideration of responses in the memory construction, which can benefit generating a more desired response.

3 Methodology

In this section, we will first give an overall description of the proposed teacher-student architecture for CbR, then briefly describe the base model. The detailed teacher model and student model are presented in Sec 3.3 and 3.4. Lastly, we summarize the training updates of the two models in Sec 3.5.

3.1 Model Architecture

The CbR task provides a conversation context X and a document D as inputs, requiring the model to generate a response R to X by referring to D. In the rest of the paper, we use $|X|$, $|D|$, and $|R|$ to denote the number of tokens in X, D, and R respectively. To pinpoint accurate document information for response generation, we design a teacher-student framework to construct document memory as follows:

- The teacher model learns a response-aware document memory $\mathbf{M}$ used in our base conversation model. Specifically, we construct a response-aware weight matrix $\mathbf{G} \in \mathbb{R}^{|D|\times|D|}$, which considers the correlation between context-aware document representations and response representations, and then impose $\mathbf{G}$ on the memory matrix $\mathbf{M}$. The teacher model is optimized to reconstruct the response with the use of response-aware memory $\mathbf{M}$.
- The student model learns to construct a response-anticipated weight matrix to estimate $\mathbf{G}$ used in the teacher model but without access to the response. It is a feed-forward neural network with document and context as its input.

The teacher model and the student model are jointly optimized with training data, while only the student model is applied to testing data.

3.2 Base Model

Following Qin et al. (2019), we use SAN (Liu et al., 2018) as our base model, which mainly consists of three components:

- Input encoder: We use two bi-directional LSTM encoders to extract token-level representations of the document D and the context X.
- Memory construction: We build the document memory $\mathbf{M} \in \mathbb{R}^{|D|\times k}$ (k is the hidden size of the memory) which will be used in the decoder. A cross-attention layer is first applied to the outputs of the two encoders to integrate information from the context to the document. Then, we obtain a set of context-aware document representation $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{|D|}]$. Since each $\mathbf{d}_i$ corresponds to a document token, we treat it as the context-aware token representation of the i-th token. Next, a self-attention layer is employed to ingest salient information of the context-aware document representations:

$$\mathbf{M} = \text{SelfAttn}(\mathbf{D}) = \mathbf{A}\mathbf{D}^T, \mathbf{A} = \text{softmax}(\mathbf{D}^T\mathbf{D}) \quad (1)$$

where the softmax conducts the normalization over each row of the matrix.
- Output decoder: We use an attentional recurrent decoder to generate response tokens by attending to the memory $\mathbf{M}$. The initial hidden state is set as the summation of token-level context representations. For each decoding step t, we get a hidden state $\mathbf{h}_t$:

$$\mathbf{z}_t = \text{GRU}(\mathbf{e}_{t-1}, \mathbf{h}_{t-1}), \quad (2)$$

$$\mathbf{h}_t = \mathbf{W_1}[\mathbf{z}_t; \text{CrossAttn}(\mathbf{z}_t, \mathbf{M})] \quad (3)$$

where $[;]$ indicates concatenation, and the cross-attention layer here integrates information from the memory to the recurrent outputs. $\mathbf{e}_{t-1}$ is the word-embedding at step $t-1$. Finally, we generate a token y_t by a softmax on $\mathbf{h}_t$.

Our model modifies the memory construction by refining its self-attention layer so that the memory represents more accurate and on-demand knowledge that helps generating the response.

3.3 Teacher Model

To ingest accurate memory information for response generation under the aforementioned base model, our teacher model builds a response-aware weight matrix $\mathbf{G} \in \mathbb{R}^{|D|\times|D|}$ given the context-aware document representation $\mathbf{D}$ and the response R, then refines the document memory $\mathbf{M}$ with $\mathbf{G}$. Elements in $\mathbf{G}$'s indicate the importance of tokens or token pairs in the document, with consideration of the response information.

First, we describe how to modify the memory matrix $\mathbf{M}$ when $\mathbf{G}$ is given. The original memory $\mathbf{M}$ is constructed by a self-attention operation as Eq. 1. To facilitate response awareness, we update the attention weight matrix $\mathbf{A}$ by element-wise multiplying $\mathbf{G}$, and then get the refined memory $\widetilde{\mathbf{M}}$ as

$$\mathbf{A} = \text{softmax}(\mathbf{D}^T\mathbf{D}), \widetilde{\mathbf{M}} = (\mathbf{G} \odot \mathbf{A})\mathbf{D}^T. \quad (4)$$

In the following, we describe two methods to construct the response-aware weight matrix $\mathbf{G}$: (1)

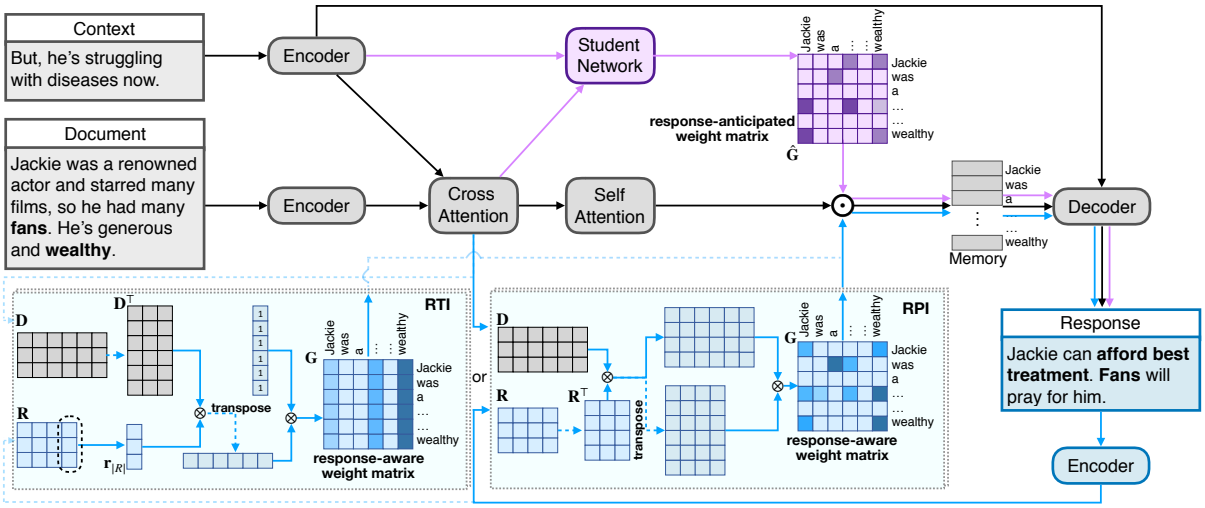

Figure 2: The architecture of our model. Blocks and lines in gray color compose the base model. Blue and gray parts compose the teacher model, while purple parts compose the student model. All components work for training, while only the student model and the decoder works for inference. In the response-aware/anticipated weight matrix, darker grids indicate higher weights. ($\otimes$: matrix multiplication; $\odot$: element-wise matrix multiplication.)

We measure the response-aware token importance (RTI) considering the ground-truth response to construct $\mathbf{G}$. (2) We measure the response-aware pairwise importance (RPI) of each token pair (i, j), which can be directly assigned to the element G_{ij} in $\mathbf{G}$. For both methods, matrix elements can be either continuous or binary.

Response-Aware Token Importance (RTI)

We denote the response-aware token importance of document tokens as $\boldsymbol{\beta} \in \mathbb{R}^{|D|}$, and measure it by response R and context-aware token representation $\mathbf{D}$. To obtain $\boldsymbol{\beta}$, we first apply an encoder to obtain the token-level representations of the response as $[\mathbf{r}_1, \ldots, \mathbf{r}_{|R|}]$ and use its last hidden state $\mathbf{r}_{|R|}$ as the sentence-level response representation. The response-aware token importance of token i is defined as the similarity between its context-aware token representation $\mathbf{d}_i$ and the response representation $\mathbf{r}_{|R|}$. Next, we adjust each attention distribution (i.e., each column of $\mathbf{A}$) with each of its attention weight multiplied by the token importance β_i. Therefore, the resulting $\mathbf{G}$ can be obtained as:

$$\beta_i = \mathbf{d}_i^T \mathbf{r}_{|R|}, \quad \mathbf{G} = \mathbf{1}\boldsymbol{\beta}^T, \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^{|D|}$ represents an identity vector with all elements as 1. By plugging the above $\mathbf{G}$ in Eq. 5, we can construct a memory matrix with plagiarized signals from the response. In this way, the self-attention distributions can adjust to emphasize important tokens, and their corresponding context-aware document token representations become more important in the memory matrix.

Recall that the document contains a large amount of noise information in CbR. Thus the attention distributions may become long-tailed due to the existence of many redundant document tokens. Hence, we can further construct a binary weighting vector based on $\boldsymbol{\beta}$. We keep the weight of each element as 1 with the probability of β_i calculated in Eq. 5. If the weight of a token turns to 0, this token is deactivated in calculating the attention distributions. However, the binary weight sampled from the Bernoulli distribution is not differentiable. To enable back-propagation of our model, we apply the Gumbel-Softmax (Jang et al., 2016) to approximate the Bernoulli distribution in the training phase, and sample the binary value from the Bernoulli distribution in the prediction phase as:

$$\mathbf{G} = \mathbf{1}g(\boldsymbol{\beta})^T, \tag{6}$$

where $g(\boldsymbol{\beta})$ is defined as:

$$\begin{cases} g(\beta_i) = \text{GumbelSoftmax}(\beta_i) & \text{Training}, \\ g(\beta_i) \sim \text{Bernoulli}(\beta_i) & \text{Prediction}. \end{cases} \tag{7}$$

The objective function of the teacher model is to maximize the log-likelihood of responses generated by the response-aware memory constructed with $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = f_{\theta_t}^t(D, X, R), \mathcal{J}_t = \mathop{\mathbb{E}}_{D,X,R\sim\mathcal{D}} \log P_\phi(R|D, X, \boldsymbol{\beta}), \tag{8}$$

where f^t denotes operations in Eq. 5 and its preorder operations. θ_t consists of all parameters in the layers of f^t. ϕ denotes parameters in Eq. 1 to Eq. 3. Both ϕ and θ_t are learning parameters for $\mathcal{J}_t$.

Response-Aware Pairwise Importance (RPI) Instead of using token importance, we can construct $\mathbf{G}$ by the pairwise importance of token pairs. After obtaining the token representations $[\mathbf{r}_1, \ldots, \mathbf{r}_{|R|}]$ from the response encoder similarly as in RTI, we can calculate the similarity of each $\mathbf{d}_i$ towards all $\mathbf{r}_j$'s, denoted as $\mathbf{n}_i \in \mathbb{R}^{|R|}$. Each element in $\mathbf{G}$ can be associated with a weight $\mathbf{B}_{ij}$ defined as the inner-product between $\mathbf{n}_i$ and $\mathbf{n}_j$. Thus, we can treat $\mathbf{B}$ as the response-aware pairwise importance, and directly set each element in $\mathbf{G}$ as $\mathbf{B}_{ij}$:

$$\mathbf{n}_i = [\mathbf{r}_1, \ldots, \mathbf{r}_{|R|}]^T \mathbf{d}_i, \mathbf{B}_{ij} = \mathbf{n}_i^T \mathbf{n}_j, \mathbf{G} = \mathbf{B}. \tag{9}$$

Compared with response-aware token importance in which the designed $\mathbf{G}$ has identical column values, response-aware pairwise importance allows different values of different index (i, j)'s in $\mathbf{G}$ (but (i, j) and (j, i) have the same value since $\mathbf{G}$ is symmetric). Thus, the space of $\mathbf{G}$ is larger.

Notice that, the aforementioned binary processing with each β_i can also be applied on each $\mathbf{B}_{ij}$ here and the resulting $\mathbf{G}$ is binary. By using a binary $\mathbf{G}$ in our model, the memory construction can be considered as passing through a Graph Attention Network (GAT) (Veličković et al., 2018), which also constructs a graph and updates its representations relying on the information from itself and neighbors on the graph. However, our neighborhood matrix (i.e. $\mathbf{G}$ in our model) is not predefined as in GAT but dependant on the inputs $\mathbf{d}_i$'s and $\mathbf{r}_j$'s, which involve parameters to be estimated.

The objective of the teacher model for RPI can be modified from Eq. 8 by replacing $\boldsymbol{\beta}$ with $\mathbf{B}$ obtained in Eq. 9.

3.4 Student Model

The student model learns to construct a response-anticipated weight matrix to estimate the weight matrix $\mathbf{G}$ in the teacher model without access to the ground-truth R. If we employ RTI, the estimated target of the student model is $\boldsymbol{\beta}$ in Eq. 5. For RPI, the estimated target is $\mathbf{B}$ in Eq. 9.

Given $\mathbf{D}$ and $\mathbf{X}$ as inputs, we apply a bilinear attention layer to obtain a hidden representation matrix $\mathbf{H}$. We apply a two-layer multi-layer perceptron (MLP) with ReLU activation to estimate $\boldsymbol{\beta}$; we combine two attention outputs by $\mathbf{W_a}$ to estimate $\mathbf{B}$ in the RPI:

$$\mathbf{H} = \text{softmax}(\mathbf{D}^T \mathbf{W} \mathbf{X}) \mathbf{X}^T, \tag{10}$$

$$\begin{cases} \hat{\boldsymbol{\beta}} = \text{MLP}(\mathbf{H}) & \text{for RTI,} \\ \hat{\mathbf{B}} = \mathbf{H} \mathbf{W_a} \mathbf{H}^T & \text{for RPI.} \end{cases} \tag{11}$$

The objective function of the student model is to maximize the log-likelihood of generating responses based on the estimated $\hat{\boldsymbol{\beta}}$ or $\hat{\mathbf{B}}$, and diminish the gap of the weighting vector or matrix between the student model and the teacher model by a mean square loss. Taking the RTI strategy as an example, we optimize the following objective:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= f^s_{\theta_s}(D, X), \\ \mathcal{J}_s &= \mathop{\mathbb{E}}_{D,X,R \sim \mathcal{D}} \log P_\phi(R|D, X, \hat{\boldsymbol{\beta}}) - \lambda \mathcal{L}_{\text{MSE}}(\boldsymbol{\beta}, \hat{\boldsymbol{\beta}}), \end{aligned} \tag{12}$$

where f^s denotes the operation in Eq. 11 and its preorder operations. θ_s consists of the layer parameters in f^s. λ balances the two loss terms. For RPI, we replace to optimize with $\mathbf{B}$ and $\hat{\mathbf{B}}$.

3.5 Model Training

We first train the teacher model until it converges, and then train the student model with the use of $\boldsymbol{\beta}$ or $\mathbf{B}$ from the converged teacher model. Next, we repeat the above processes iteratively. In the training of the teacher model, we fix parameters in θ_s (except parameters shared with θ_t) and train the model subject to $\mathcal{J}_t$; for the student model, we fix ϕ and θ_t (except parameters shared with θ_s) and train the model subject to $\mathcal{J}_s$. For inference, only the student model will be used to infer the response-anticipated weight matrix and the decoder applies it for generating the output response.

As stated in RPI, it has better model capacity by allowing a larger space of $\mathbf{G}$ with the use of the weight matrix $\mathbf{B}$ instead of the token importance vector $\boldsymbol{\beta}$ in RTI. In terms of optimization, we need to estimate more parameters by using RPI, which requires higher training difficulty.

4 Experiment Setting

4.1 Dataset

We use the dataset for the CbR task released by Qin et al. (2019). The dataset contains crawled articles and discussions about these articles from Reddit. The articles act as the documents, while the discussions serve as conversational contexts and responses. In total, we have 2.3M/13k/1.5k samples for training/testing/validation.

	Appropriateness			Grounding						Informativeness			
	NIST	BLEU	Meteor	P	R	F1	P_{GT}	R_{GT}	$F1_{GT}$	Ent4	Dist1	Dist2	Len
Human	2.650	3.13%	8.31%	2.89%	0.45%	0.78%	0.44%	0.09%	0.14%	10.445	0.167	0.670	18.8
Seq2Seq	2.223	1.09%	7.34%	1.20%	0.05%	0.10%	0.89%	0.05%	0.09%	9.745	0.023	0.174	15.9
MemNet	2.185	1.10%	7.31%	1.25%	0.06%	0.12%	0.91%	0.05%	0.10%	9.821	0.035	0.226	15.5
GLKS	2.413	1.34%	7.61%	2.47%	0.13%	0.24%	0.84%	0.05%	0.10%	9.715	0.034	0.213	15.3
CMR	2.238	1.38%	7.46%	3.39%	0.20%	0.38%	0.91%	0.05%	0.10%	9.887	0.052	0.283	15.2
CMR+Copy	2.155	1.41%	7.39%	5.37%	0.28%	0.54%	0.92%	0.06%	0.11%	9.798	0.044	0.266	14.4
RAM_T	**2.510**	**1.43%**	**7.74%**	4.46%	0.26%	0.49%	**1.04%**	**0.08%**	**0.15%**	**9.900**	**0.053**	**0.290**	15.1
RAM_P	2.353	1.40%	7.59%	3.89%	0.21%	0.41%	0.97%	0.07%	0.13%	9.891	0.049	0.279	14.9
RAM_T+Copy	2.467	1.41%	7.64%	**6.14%**	**0.32%**	**0.61%**	0.65%	0.04%	0.08%	9.813	0.045	0.265	14.9
RAM_P+Copy	2.342	1.41%	7.51%	5.83%	0.30%	0.57%	0.84%	0.06%	0.10%	9.798	0.045	0.267	14.6

Table 1: Automatic evaluation results on all competing methods. *Len* denotes the length of the generated responses.

4.2 Implementation Details

For all methods, we set word embedding dimension to 300 with the pre-trained GloVe (Pennington et al., 2014). Following Qin et al. (2019), our vocabulary contains top 30k frequent tokens. We use bi-LSTMs with the hidden dimensions of 512 and the dropout rate of 0.4 in our encoders. We optimize models by Adam with an initial learning rate of 0.0005 and the batch size of 32. All conversation contexts/responses/documents are truncated to have the maximum length of 30/30/500. For training, we set λ as 1 in the loss of student models after tuning. For inference, we apply a top-k random sampling decoding (Edunov et al., 2018) with k=20. The validation set is for early stopping. Aforementioned implementation details can be found in our codes [1].

4.3 Competing Methods

1. **Seq2Seq** (Sutskever et al., 2014). The standard Seq2Seq model that leverages only the conversational context for response generation.
2. **MemNet** (Ghazvininejad et al., 2018). A knowledge-grounded conversation model that uses a memory network to store knowledge facts.
3. **GLKS** (Ren et al., 2020). It applies a global knowledge selector in encoding and a local selector on every decoding step.
4. Conversation with Machine Reading (**CMR**) (Qin et al., 2019). The state-of-the-art model on the CbR task, which is also our base model (Sec 3.2). Here, we use the full model of CMR (called CMR+w in (Qin et al., 2019)), since the full model outperforms other CMR's variants on most metrics. We further apply the copy mechanism (See et al., 2017) to this base model (**CMR+Copy**).
5. Four variants of our proposed models: **RAM_T** denotes our **R**esponse-**A**nticipated **M**emory-based model with R**TI**, and **RAM_T+Copy** denotes its copy version. **RAM_P** and **RAM_P+Copy** denote our model with R**PI** and its copy variant .

[1] https://github.com/tianzhiliang/RAM4CbR

4.4 Evaluation Metrics

Following all metrics in Qin et al. (2019), we evaluate all methods by both automatic and human evaluations. For automatic evaluations, we evaluate the responses in three aspects:

1. **Appropriateness**.
We use three metrics to evaluate the overall quality of a response: BLEU-4 (Papineni et al., 2002), Meteor (Banerjee and Lavie, 2005), and NIST (Doddington, 2002). NIST is a variant of BLEU that measures n-gram precision weighted by the informativeness of n-grams.
2. **Grounding**. We measure the relevance between documents and generated responses to reveal the effectiveness of responses exploiting the document information. We define #overlap as the number of non-stopword tokens in both the document D and the generated response $\hat{R}$ but not in contexts X. We calculate the precision P and recall R as

$$\#\text{overlap} = |(D \cap \hat{R}) \backslash X \backslash S|, \quad (13)$$

$$\text{P} = \frac{\#\text{overlap}}{|\hat{R} \backslash S|}, \text{R} = \frac{\#\text{overlap}}{|D \backslash S|}, \quad (14)$$

where S denotes the stopword list. *F1* is the harmonic mean of precision P and recall R.
We further propose to measure the effectiveness of exploiting the document information considering the ground-truth. In this way, we evaluate how many ground-truth information models can exploit from the document. We define $\#\text{overlap}_{GT}$ as the number of non-stopword tokens in the document D, the generated response $\hat{R}$ and the ground-truth R but not in contexts X. The precision and recall

	Appropriateness			Grounding						Informativeness			
	NIST	BLEU	Meteor	P	R	F1	P_{GT}	R_{GT}	$F1_{GT}$	Ent4	Dist1	Dist2	Len
RAM_T	2.510	1.43%	7.74%	4.46%	0.26%	0.49%	1.04%	0.08%	0.15%	9.900	0.053	0.290	15.1
RAM_P	2.353	1.40%	7.59%	3.89%	0.21%	0.41%	0.97%	0.07%	0.13%	9.891	0.049	0.279	14.9
RAM_T (Teacher)	2.539	1.43%	7.85%	4.47%	0.26%	0.49%	**1.05%**	0.08%	0.15%	**9.904**	0.053	**0.290**	15.1
RAM_P (Teacher)	**2.551**	**1.47%**	**7.88%**	**4.56%**	**0.27%**	**0.50%**	0.99%	**0.08%**	**0.16%**	9.900	**0.053**	0.287	15.1
RAM_T_Binary	2.560	1.63%	7.91%	3.75%	0.21%	0.40%	0.87%	0.07%	0.12%	9.890	0.052	0.283	15.1
RAM_P_Binary	2.403	1.51%	7.63%	3.55%	0.18%	0.38%	0.85%	0.07%	0.12%	9.887	0.046	0.274	14.6

Table 2: Performance comparison on our model variants. Line1&2: our models trained by the full teacher-student framework. Line3&4: our models trained with the teacher model only. Line5&6: our models with binary weight matrices. Bold values are the best results among the first four lines; underlines mark the best ones among the first two and last two lines.

	H-Appr	H-Ground	H-Info
Human	2.986	2.521	3.007
Seq2Seq	1.902	1.564	2.040
MemNet	1.872	1.574	2.105
GLKS	2.073	1.593	2.071
CMR	2.188	1.678	2.219
CMR+Copy	2.063	1.773	2.075
RAM_T	**2.259**	1.714	**2.312**
RAM_P	2.213	1.682	2.231
RAM_T+Copy	2.109	**1.861**	2.240
RAM_P+Copy	2.114	1.775	2.115

Table 3: Human annotation results.

are as following,

$$\#\text{overlap}_{GT} = |(D \cap \hat{R} \cap R) \backslash X \backslash S|, \quad (15)$$

$$\text{P}_{GT} = \frac{\#\text{overlap}_{GT}}{|\hat{R} \backslash S|}, \text{R}_{GT} = \frac{\#\text{overlap}_{GT}}{|D \backslash S|}, \quad (16)$$

where $F1_{GT}$ is the harmonic mean of precision P_{GT} and recall R_{GT}.

3. **Informativeness**. *Ent-n* (Mou et al., 2016) measures responses' informativeness with the entropy of the n-gram count distribution. *Dist-n* (Li et al., 2016a) evaluates the diversity of responses via the proportion of unique n-grams among all responses.

For human evaluations, we hire five annotators from a commercial annotation company to evaluate 200 randomly selected test samples, and results from different models are shuffled. The annotators evaluate on a 5-point scale in three aspects: overall quality (*H-Appr*), relevance with documents (*H-Ground*), and informativeness (*H-Info*).

5 Experimental Results and Analysis

In this part, we first show the performance of all methods in Sec 5.1. Then, we validate the effectiveness of response anticipation on CbR in Sec 5.2 by comparing the top similar tokens with the response using their representations in the memory. We also compare more variants of our model in Sec 5.3, including the token importance versus pairwise importance, and each method with continuous weights versus their variants with binary weights. At last, we conduct a case study in Sec 5.4.

	Top10 tokens		Top20 tokens	
	Emb-M	Emb-B	Emb-M	Emb-B
CMR	0.482	0.356	0.571	0.420
RAM_T_Soft	**0.745**	**0.520**	**0.867**	**0.616**
RAM_P_Soft	0.518	0.441	0.634	0.493

Table 4: Similarity between important document tokens picked by gold responses and the accumulated attention weights in the models.

5.1 Overall Performance

Results of all models on automatic and human evaluations are shown in Table 1 and Table 3. MemNet outperforms Seq2Seq on most metrics, which validates that it is important to utilize document information in CbR. However, MemNet only slightly improves on Grounding. Both GLKS and CMR outperform MemNet on most metrics, indicating that it matters how to construct the document memory used in conversation models for CbR. Compared with CMR, CMR+Copy is more competitive on Grounding but weaker on other metrics.

Our proposed models outperform other competing methods on all metrics, including automatic and human evaluations. For models without the copy mechanism, RAM_T performs the best. For models with copy, RAM_T+Copy and RAM_P+Copy excel CMR+Copy on most metrics. Overall, our proposed strategy works well on both the model with and without copy mechanism. We will compare RAM_T and RAM_P in details in Sec 5.3.

5.2 Effectiveness of Response Anticipation

In this section, we investigate whether anticipating response contributes to building a better document memory. We first calculate the semantic similarity between each document token and the response us-

	Case 1	Case 2
Document	fa premier league was the fourth season of the competition, since its formation in 1992. due to the decision to reduce the number of clubs in the premier league from 22 to 20, only two clubs were promoted instead of the usual three , middlesbrough and bolton wanderers.	darko milicic. darko milicic (serbian cyrillic. serbian pronunciation. born june 20, 1985) is a serbian former professional basketball player . he is 2.13 m (7 ft 0 in) , and played center .
Context	at least we qualified for a european competition we're capable of winning now	that darko milicic, who was drafted 2nd overall in the 2003.nba draft is currently an apple farmer in serbia.
Seq2Seq	i do n't really need to take a time and a bit more and i think he 's saying it was n't in an accident .	he is so happy when i 'm not in 0ame universe as the first time.
MemNet	i am not saying i was a kid .	you know what ? is there anything in a book ?
GLKS	i have a pretty good chance of being the first person i know !	i think a lot of people are still able to get in a hour
CMR	well , at what point do you think about how they are getting play for ?	i remember my comment on my post. and i am not sure why but my point is that he has the best score that will always get a good
CMR+Copy	they are , but not the same as the first one .	he also played the same game, is there title to be a team
RAM_T	i think we have num teams playing the premier league team. in my opinion he was not a good player, but the united kingdom was in the europa	i love him the next time i play for num years, so that is probably the only option i understand.
RAM_T +Copy	they are the best player in the world.	he also played the second one, but that doesn't mean it was num years ago.

Figure 3: Test samples with generated responses of all models. A colored word in the responses indicate that it has similar words with documents or contexts, which are marked in the same color.

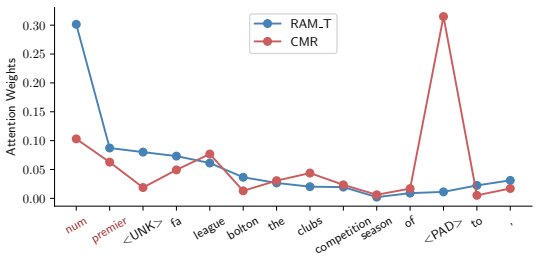

Figure 4: The accumulated attention weights of documents tokens on RAM_T and CMR on Case 1 in Fig. 3. We only show top tokens in both methods here.

ing their Glove embeddings, and select top K document tokens. Next, we accumulate the attention weights of each token in all attention distributions in the self-attention weights $\mathbf{A}$ in Eq. 1, i.e. summation over each column of $\mathbf{A}$. Then we select the top K tokens according to their accumulated attention weights. Here, we set $K = 10, 20$. We apply metrics in Liu et al. (2016) to calculate the similarity of two token sets extracted above, including maximal tokens-tokens embedding similarity (Emb-M) and bag-of-word embedding similarity (Emb-B). A higher similarity score indicates more response information anticipated by the model. Table 4 shows the results of our two models RAM_T and RAM_P as well as CMR (We use the original self-attention matrix $\mathbf{A}$ for the above calculation for CMR). Results demonstrate that our model is able to output more response-anticipated self-attention distributions, which benefits generating a response close to the ground truth.

5.3 Analysis on Different Model Variants

Token importance vs Pairwise importance. We compare our model variants with different strategies to construct the response-aware/anticipated weight matrix , i.e. RAM_T (Eq. 5) and RAM_P (Eq. 9). We not only compare their overall performance by the teacher-student framework (Eq. 8 & 12) but also the teacher model only (Eq. 12).

The first four rows in Table 2 shows the results. We have an interesting finding that RAM_P underperforms RAM_T in the full teacher-student framework, but outperforms RAM_T on the mode with teacher model only on most metrics. This result is actually consistent with our discussion in Sec 3.5 that RAM_P has a higher capacity to carry more information in $\mathbf{G}$, thus its teacher model yields better performance. However, for the student model, RAM_P is more difficult to converge to a good local optimum due to more parameters to be estimated, resulting in that its overall performance may not exceed that of RAM_T.

Continuous weight vs Binary weight. We also compare the model variants with continuous weight (Eq. 5) and binary weight (Eq. 6). The last two rows in Table 2 give the results of the variants of RAM_T and RAM_P with a binary $\mathbf{G}$. We can see that both RAM_T and RAM_P with a binary weight matrix performs better on Appropriateness, which means a sparse $\mathbf{G}$ on the attention matrix can help select more concise information to construct the memory. Nevertheless, models with a continuous

weight matrix can generate more informative responses owing to their ability to access broader and more information from the document.

5.4 Case Study

Table 3 shows two test samples with generated responses of all models. For Case 1, Seq2Seq and MemNet cannot generate responses relevant to either the document or context. CMR catches the topic "sports", while GLKS and CMR+Copy use "first person" and "first one" to reflect "only two" mentioned in the document. The response of RAM_T contains information related to both document ("num teams" and "premier league") and context ("europa"). RAM_T+Copy is also highly relevant to the document and the context, and copies "player" from the document. For Case 2, the first four methods have little relation to the document or the context. CMR+Copy mentions "played". Our models mention "played" and "num years". By examining the cases, our method shows promising improvements over existing methods. However, generation on the CbR task is very challenging and there is still a huge space to improve.

We plot the accumulated attention weights of RAM_T and CMR as in Sec 5.2 of the document tokens on Case 1. Fig. 4 shows that RAM_T's attention highlights "num" and "premier", and thus it generates the above words in its response.

6 Conclusion

Focusing on the CbR task, we propose a novel response-anticipated document memory to exploit and memorize the document information that is important in response generation. We construct the response-anticipated memory by a teacher-student framework. The teacher accesses the response and learns a response-aware weight matrix; the student learns to estimate the weight matrix in the teacher model and construct the response-anticipated document memory. We verify our model on both automatic and human evaluations and experimental results show our model obtains the state-of-the-art performance on the CbR task.

Acknowledgments

Research on this paper was supported by Hong Kong Research Grants Council under grants 16202118 and 16212516 and Tencent AI Lab Rhino-Bird Focused Research Program (No. GF202035).

References

Shubham Agarwal, Ondřej Dušek, Ioannis Konstas, and Verena Rieser. 2018. A knowledge-grounded multimodal search-based conversational agent. In *EMNLP*, pages 59–66.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Workshop on ACL*, pages 65–72.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLTCon*, pages 138–145.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*, pages 489–500.

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 *AAAI*, pages 5110–5117.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*, pages 110–119.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *EMNLP*, pages 1192–1202.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*, pages 2122–2132.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. In *ACL*, pages 1694–1704.

Chuan Meng, Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Refnet: A reference-aware network for background based conversation. In *AAAI*.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *EMNLP*, pages 2322–2332.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING*, pages 3349–3358.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In *EMNLP*, pages 690–695.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *ACL*, pages 5427—5436.

Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking globally, acting locally: Distantly supervised global-to-local knowledge selection for background based conversation. In *AAAI*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, pages 1073–1083.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

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 *AAAI*, pages 3776–3783.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL*, pages 1577–1586.

Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *IJCAI*, pages 4382–4388.

Yiping Song, Zhiliang Tian, Dongyan Zhao, Ming Zhang, and Rui Yan. 2017. Diversifying neural conversation model with maximal marginal relevance. In *IJCNLP*, pages 169–174.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *NIPS*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Zhiliang Tian, Wei Bi, Xiaopeng Li, and Nevin L Zhang. 2019. Learning to abstract for memory-augmented conversational response generation. In *ACL*, pages 3816–3825.

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 *ACL*, 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 *NLPS*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models. In *AAAI*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019a. Proactive human-machine conversation with explicit conversation goal. In *ACL*, pages 3794–3804.

Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019b. Response generation by context-aware prototype editing. In *AAAI*, pages 7281–7288.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*.

Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. In *CIKM*, pages 1341–1350.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, pages 654–664.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.