A Neural Conversation Generation Model via Equivalent Shared Memory Investigation

Changzhen Ji[1], Yating Zhang[2], Xiaozhong Liu[3],
Adam Jatowt[4], Changlong Sun[2], Conghui Zhu[1] and Tiejun Zhao[1]
[1]Harbin Institute of Technology, Harbin, China
[2]Alibaba Group, Hangzhou, China
[3]Worcester Polytechnic Institute, Worcester, Massachusetts, USA
[4]University of Innsbruck, Innsbruck, Austria
czji_hit@outlook.com, ranran.zyt@alibaba-inc.com, liu237@indiana.edu,
adam.jatowt@uibk.ac.at, changlong.scl@taobao.com, {conghui,tjzhao}@hit.edu.cn

ABSTRACT

Conversation generation as a challenging task in Natural Language Generation (NLG) has been increasingly attracting attention over the last years. A number of recent works adopted sequence-to-sequence structures along with external knowledge, which successfully enhanced the quality of generated conversations. Nevertheless, few works utilized the knowledge extracted from similar conversations for utterance generation. Taking conversations in customer service and court debate domains as examples, it is evident that essential entities/phrases, as well as their associated logic and inter-relationships can be extracted and borrowed from similar conversation instances. Such information could provide useful signals for improving conversation generation. In this paper, we propose a novel reading and memory framework called Deep Reading Memory Network (DRMN) which is capable of remembering useful information of similar conversations for improving utterance generation. We apply our model to two large-scale conversation datasets of justice and e-commerce fields. Experiments prove that the proposed model outperforms the state-of-the-art approaches.

CCS CONCEPTS

• **Information systems → Information retrieval**; *Retrieval tasks and goals*; Question answering.

KEYWORDS

Conversation Generation, Equivalent Shared Memory, Deep Reading Memory Network

ACM Reference Format:
Changzhen Ji[1], Yating Zhang[2], Xiaozhong Liu[3],, Adam Jatowt[4], Changlong Sun[2], Conghui Zhu[1] and Tiejun Zhao[1]. 2021. A Neural Conversation Generation Model via Equivalent Shared Memory Investigation. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482407

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482407

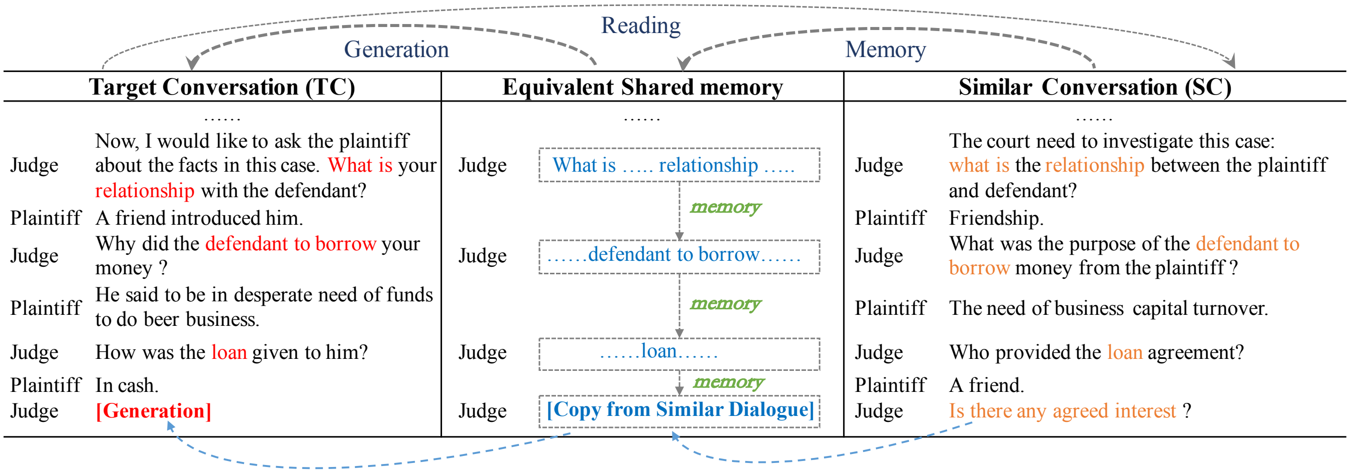
1 INTRODUCTION

Over the past years, chatbot technologies and conversation mining approaches have been actively explored and applied in many tasks and applications for a variety of purposes including supporting users in their decision making, e.g., in e-commerce customer service and legal justice consulting. These groundbreaking developments typically utilize big data and deep learning technologies, which harness in-depth semantic and discourse information of conversations and apply sophisticated learning models.

In contrast to classical rule-based [53] and template-based [43, 56] approaches applied for conversation generation, sequence-to-sequence models [2, 30, 49] are able to understand sequential dependencies between conversation utterances which is crucial for content generation [46, 52]. Recently, external knowledge has been also utilized to further improve the performance of utterance generation. For instance, open-domain, unstructured knowledge has been employed [12, 40, 58, 61], while in other cases, domain-specific knowledge bases (organized based on triples) have been used in task-oriented conversations [7, 28, 31, 57]. Furthermore, large-scale knowledge graphs have also recently been employed [17, 35, 59]. Although the above-mentioned works allowed achieving superior results, constructing external knowledge bases can be quite an expensive and arduous task. Additionally, the low adaptability and transferability of domain knowledge restrict their real-world applications. Therefore, further efforts are required in order to address these challenges.

Equivalent Shared Memory (ESM) is a phenomenon that can be observed in conversation corpora, especially, between conversations belonging to the same domain. While the detailed contents of different conversations vary, there are certain common patterns which can be considered as the common backbone memory. Such shared memory can be then useful for supporting the utterance generation task. As portrayed in the example in Figure 1, a target legal conversation discourse (one on the left) and a selected reference prior conversation (on the right) share certain common patterns (i.e., ESM). Then the utterance generator is able to copy the utterance (in this case the judge's question) from the reference conversation and paste it into the Target Conversation (TC) context. This scenario can also occur in other conversation corpora, e.g., customer service, where an agent may repeat the same or a similar responses as ones in past dialogues.

Motivated by these observations, we propose a novel model to discover ESM by extracting critical words, phrases, and discourse



Target Conversation (TC)		Equivalent Shared memory		Similar Conversation (SC)	
	……		……		……
Judge	Now, I would like to ask the plaintiff about the facts in this case. What is your relationship with the defendant?	Judge	What is ….. relationship …..	Judge	The court need to investigate this case: what is the relationship between the plaintiff and defendant?
Plaintiff	A friend introduced him.			Plaintiff	Friendship.
Judge	Why did the defendant to borrow your money ?	Judge	……defendant to borrow……	Judge	What was the purpose of the defendant to borrow money from the plaintiff ?
Plaintiff	He said to be in desperate need of funds to do beer business.			Plaintiff	The need of business capital turnover.
Judge	How was the loan given to him?	Judge	……loan……	Judge	Who provided the loan agreement?
Plaintiff	In cash.			Plaintiff	A friend.
Judge	**[Generation]**	Judge	**[Copy from Similar Dialogue]**	Judge	Is there any agreed interest ?

Figure 1: A toy case in our judicial dataset. We can generate the next utterance by reading similar conversations and memorizing related entities, phrases as well as sentences.

information from similar conversations. The proposed model, the Deep Reading Memory Network **(DRMN)**, tries to reproduce the human decision-making process, in resemblance to a human brain which ”retrieves” similar memories to utilize them for generating the next utterance in an ongoing conversation. **DRMN** treats the last utterance[1] [64] as a query and Similar Conversations (SC) as a search database. The query is issued to retrieve from the search database the information most relevant for the next utterance to be issued.

To verify this hypothesis and to validate the proposed model, we conduct experiments on two large-scale conversation datasets from different domains - *court debate dataset* from a legal field and *customer service dataset* from an e-commerce field. We apply **DRMN** to the above two datasets and assess the model performance based on both automated and human evaluation. The experimental results indicate that **DRMN** significantly outperforms the baseline models.

Our study contributes to the growing body of research in exploring conversation generation. The particular contributions of this paper are as follows:

- We propose a novel end-to-end model, called the Deep Reading Memory Network, for the conversation generation task to explore Equivalent Shared Memory from past conversations.
- We demonstrate that the proposed model has sufficient domain adaptability and generalization ability by experimenting on two datasets that originate from different domains - *court debate* and *customer service* conversation datasets. Experimental results show that our model produces the best results on both the datasets compared to the state-of-the-art methods.
- To support and motivate other scholars for further investigating this novel and an important research problem, we make the experimental datasets and code publicly available[2].

2 RELATED WORK

2.1 Conversation Generation

Maintaining intelligent conversations to facilitate life in the real world has been the long-term goal of Artificial Intelligence (AI). In this regard, recently, the research in conversation generation, which is an important task in Natural Language Processing (NLP), has generated impressive achievements.

In the early years, researchers mostly adopted rule-based and template-based methods: for example, Joseph *et al.* [53] proposed to generate responses by reassembling rules associated with selected decomposition rules. Jost *et al.* [43] attempted at building systems that learn what constitutes a good conversation strategy through trial-and-error interaction.

In the recent years, due to the capabilities of deep neural networks, sequence-to-sequence models [2, 18, 20, 30, 36, 49] become popular. They are widely used in conversation generation allowing to achieve significant improvements. For example, [46, 52] supplement the classic attention model with contextual information. [6, 8, 65] solve the problem of the lack of diversity and boring responses being generated for open-ended utterances. The copy mechanism [15, 44] and hierarchical LSTM [60] led to increased research in text generation. EED [38] used example vectors to guide the generation of dialogue. CCN [19] applied a hierarchical encoder and cross-copying method to the field of conversation generation. The introduction of Transformers [51] brought text generation to a new level. Tranformer-based models are also widely used in conversation generation [3, 62, 64].

More recently, employing external knowledge to conversation generation became a popular approach: [12, 40, 58, 61] utilized unstructured knowledge for conversation generation. Additionally, structured knowledge triples have also been widely employed in the conversation generation task [55, 57]. Knowledge graphs as larger-scale external knowledge sources can also be utilized in conversation generation tasks [35, 59, 63].

[1] The last utterance is usually most relevant to the forthcoming utterance to be generated.
[2] https://github.com/jichangzhen/DRMN

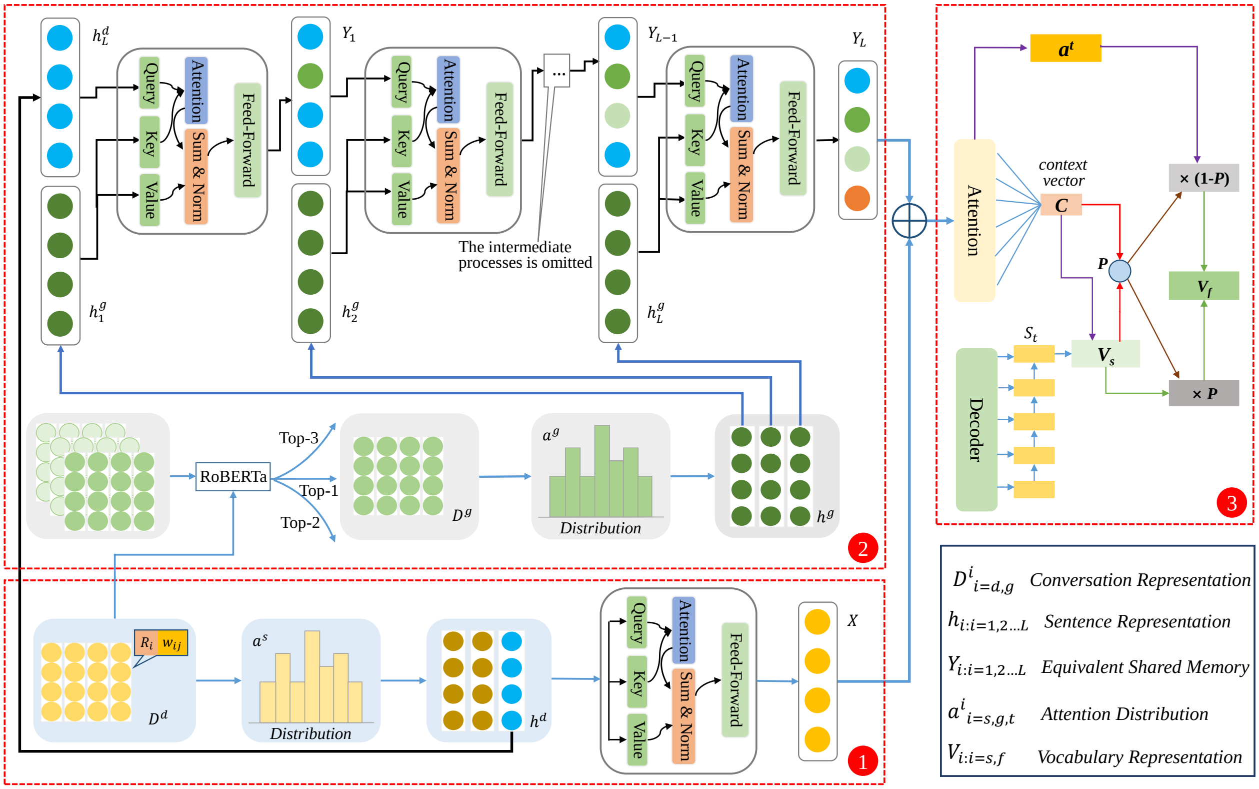


Figure 2: The overall structure of DRMN. The model is divided into three components, (1) Conversation Representation: it is used for encoding the target conversation; (2) Equivalent Shared Memory: it is used to retrieve the similar conversations and construct ESM and (3) Utterance Generation: it is used for generating utterances.

Existing works have achieved superior results, but the extensive external knowledge construction and difficulty of domain adaptation restrict their real-life applications. We note that the model proposed in this paper does not use any external knowledge and can achieve good results in the general-domain conversation generation.

2.2 Memory Networks

Sequence-to-sequence models rely on RNN [9] and LSTM [18] to improve the word-dependency memory in the sentence. However, the memory capacity of RNN [9] and LSTM [18] is very limited. They usually only remember a dozen time steps at most. Therefore, when the length of utterance increases or the number of utterances in a dialogue grows, the sequence-to-sequence models cannot satisfy the requirements of conversation generation systems.

Memory network [54] was originally proposed by Facebook AI. It was initially used for reasoning in question answering systems. Later, the end-to-end memory network was proposed [48, 54] to solve the problem of the memory being too short in traditional sequence-to-sequence models. Afterwards, [14] proposed storage memory using stack and queue structure. [34] introduced key-value memory networks, increasing the scale of memory based on an end-to-end memory network.

Memory networks have been also widely used in recent conversation systems: [23] proposed a selectively overwriting mechanism for more efficient Dialogue State Tracking (DST) by the memory network; [50] proposed to create the document memory with some anticipated responses. [27] introduced a Heterogeneous Memory Networks (HMNs) to simultaneously utilize user utterances, conversation history, and background knowledge tuples. [5] proposed to use neural models to learn personal embeddings in conversation. In addition, memory networks have been also widely used in other natural language processing tasks, for instance in: text classification [11, 37], question answering system [16, 22], information extraction [24, 42], text generation [25, 32] and language models [41].

Different from the previous memory network structures, **DRMN** uses the last utterance of the target conversation as a query, while every utterance in the equivalent shared memory is used as key and value. It also adopts self-attention structure in the memory module, and it allows multiple loops' filtering of effective information in the memory process.

3 DEEP READING MEMORY NETWORK

In this section, we introduce the details of the proposed Deep Reading Memory Network **(DRMN)** model, in which we establish an ESM module between the context of the target conversation and other historical conversations which have similar semantics to help generate the next utterance. The overall framework of the model is shown in Figure 2. The proposed framework has three major components:

(1) **Conversation Representation**: We propose to encode each conversation fragment based on hierarchical infrastructure consisting of *utterance representation layer* and *conversation representation layer* (Section 3.1).
(2) **Equivalent Shared Memory**: We introduce the method for discovering similar conversations by using the pre-trained RoBERTa model to retrieve candidates and then constructing Equivalent Shared Memory based on these candidates (Section 3.2).
(3) **Utterance Generation**: Based on the extracted contextual information of the target conversation as well as the equivalent shared memory constructed from similar conversations, we further employ pointer generation mechanism to generate the next utterance (Section 3.3).

3.1 Conversation Representation

In order to represent the conversation fragment, we make the following definition: given a conversation $D = \{(U, R)^L\}$ containing L utterances, U and R represent the utterance and the role of a speaker, respectively, where each utterance in the conversation is expressed as $U_i = \{w_{i1}, w_{i2}, ..., w_{il}\}$, with w being a word and l denoting the length of the utterance. Note that we use D^d to represent target conversation, and the similar conversation will be denoted by D^g.

It should be pointed out that the role information can be critical for conversation generation since different characters taking part in the conversation may not necessarily share the same vocabulary space (e.g., plaintiff, defendant, customer or service staff). Therefore, we take role information into consideration in conversation representation. We concatenate the role R_i with each utterance U_i as a sentence: $S_i = [[R_i, w_{i1}], [R_i, w_{i2}], ..., [R_i, w_{il}]]$. The conversation can be expressed as $D = \{S_1, S_2, ..., S_L\}$. For utterance information, we utilize word2vec [33] to construct the initial word vectors. For role information, we take the randomly initialized vectors.

3.1.1 *Utterance layer.*

The Bidirectional Long-Short Term Memory Network (Bi-LSTM) [18] has superior performance in representing sequential text. We apply it to hierarchically encode each conversation. For the utterance layer, the encoder represents a sentence with hidden representation $h_i^d = \{h_{i1}^d, h_{i2}^d, ..., h_{il}^d\}$, as defined below:

$$h_{ij}^d = \mathbf{Bi-LSTM}(e(S_{ij}), h_{ij-1}^d) \tag{1}$$

where $e(S_{ij})$ is the embedding of S_{ij}. i represents the i-th utterance in the conversation, while j represents the j-th word in the current utterance.

We then use the attention mechanism [2] to estimate the importance of words in the sentence expressed as word-level attention distribution a^s:

$$h_i^d = \sum_{j=1}^{l} a_j^s h_{ij}^d \tag{2}$$

$$a_j^s = \frac{exp(tanh(W^d h_{ij}^d + b^d)^{\mathrm{T}} h_{ij}^d)}{\sum_{j=1}^{l} exp(tanh(W^d h_{ij}^d + b^d)^{\mathrm{T}} h_{ij}^d)} \tag{3}$$

3.1.2 *Conversation layer.*

Similarly, for the conversation layer, in order to obtain the dependencies between sentences, we again use Bi-LSTM to encode h_i^d:

$$h_i^d = \mathbf{Bi-LSTM}(e(S_i), h_{i-1}^d) \tag{4}$$

We also obtain the importance of different sentences in the conversation expressed as sentence-level attention distribution a^u:

$$h^d = \sum_{j=1}^{l} a_i^u h_i^d \tag{5}$$

$$a_i^u = \frac{exp(tanh(W^u h_i^d + b^u)^{\mathrm{T}} h_i^d)}{\sum_{i=1}^{l} exp(tanh(W^u h_i^d + b^u)^{\mathrm{T}} h_i^d)} \tag{6}$$

W^d, b^d, W^u and b^u are learnable parameters and $tanh$ is hyperbolic tangent function. Thereby, we finally obtain the target conversation representation h^d.

3.2 Equivalent Shared Memory

As mentioned before, we are motivated by the observation of Equivalent Shared Memories (ESM) existing across different conversations. The proposed model is established based on the hypothesis that ESM can be discovered from similar conversations and employed to generate the next utterance of the target conversation.

3.2.1 *Similar Conversations Discovery.*

Similar conversations play a key role in **DRMN**. First, we introduce how similar conversations are obtained. Our goal is to find similar conversations to the target conversation. Due to a typically large number of samples in real-world datasets, to assure high efficiency of retrieving similar conversations we use ElasticSearch[3] to retrieve the top 50 similar conversations as candidates by leveraging the target conversation as a query and the other samples as documents. To capture semantic information, we fine-tune the pre-trained RoBERTa model [29]. Then, we add a dense layer with softmax as a classifier to obtain the semantic similarity score between the target conversation and each candidate.

Next, we describe how the similar conversations are represented. We obtain the representation of a similar conversation by using the same way as representing the target conversation, with the difference that the similar conversation fragment is encoded only with the utterance layer. This is because, to construct the ESM, the context of the target conversation interacts with each sentence in the similar conversations. Then the similar conversation D^g is represented in the way as illustrated in Eqs. 1, 2 and 3. We then obtain the attention distribution a^g, as well as the representation of each sentence in the similar conversation, which is expressed as: $h^g = [h_1^g, h_2^g, ..., h_L^g]$.

3.2.2 *Equivalent Shared Memory Construction.*

Equivalent Shared Memory (ESM) refers to the backbone patterns that commonly appear in similar conversations, which are closely related to the sentences to be generated.

Note that the sentences in the target conversation and ones in each selected similar conversation are not in one-to-one correspondence; for example, the utterance to be generated in the target conversation could be similar to the third or fifth utterance in the similar conversation, or to any other subset of sentences. So we need to read every utterance in the similar conversation to construct ESM.

[3] https://www.elastic.co/products/elasticsearch

Take a hypothetical judicial scenario as an example. The construction of ESM could be viewed as simulating the way in which the experience of a judge gradually grows through learning from similar cases. When a judge takes part in a trial case, he/she may first mentally go over the memories (or even physically check the related documents) to find similar conversations, and then to recall (or read) their entire trial process. In the end, he/she formulates his/her own utterances for the target conversation based on the common words/logic borrowed from the similar conversation fragments.

To explore the connection between the target conversation and the fragments of similar conversations, we propose to let the last sentence appearing in the context of the target conversation interact with each utterance in the similar conversations (see procedure 2 depicted in Figure 2). This is due to the observation that, in most cases in the conversation generation task, the utterance to be generated is more related to the last utterance in the context [64]. The representation of the last sentence in the target conversation context is denoted as h_L^d (using Equation 2) while each sentence in the similar conversation is represented as $h^g = [h_1^g, h_2^g, ..., h_L^g]$.

Unlike in the case of the traditional memory networks, to obtain the dependency between h_L^d and h_i^g, the model reads h_i^g in order, and at every step adopts the self-attention [51] module as memory structure. The self-attention can be expressed as:

$$SA(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = softmax(\frac{\mathcal{Q} \cdot \mathcal{K}^T}{\sqrt{d}}) \cdot \mathcal{V} \tag{7}$$

with three inputs: the query $\mathcal{Q}$, the key $\mathcal{K}$ and the value $\mathcal{V}$. This module first takes the query to attend to the key via Scaled Dot-Product Attention, then applies those attention results upon the value.

Inspired by the concept of self-attention, we use the last sentence h_L^d as a query $\mathcal{Q}$, and each sentence in the similar conversation h_i^g as the key $\mathcal{K}$ and value $\mathcal{V}$. The memory module is defined as follows:

$$SA(h_L^d, h_1^g, h_1^g) = softmax(\frac{h_L^d \cdot h_1^{gT}}{\sqrt{d}}) \cdot h_1^g \tag{8}$$

To prevent vanishing or exploding gradients, we adopt a layer normalization operation [1] which refers to a feed-forward network $\mathcal{F}$ with RELU activation function [13, 66]:

$$\mathcal{F}(x) = max(0; xW^f + b^f)W^h + b^h \tag{9}$$

where W^f, b^f, W^h, and b^h are learnable parameters.

We represent the memory in a time step t as Y_t, and the first memory Y_1 can be represented as:

$$Y_1 = \mathcal{F}(SA(h_L^d, h_1^g, h_1^g)) \tag{10}$$

Our reading and memory update proceeds iteratively, and the previous memory is used as the input of the next memory. The memory content will continue to be enriched as the amount of information increases. Thus, this process can be described as:

$$Y_t = \mathcal{F}(softmax(\frac{Y_{t-1} \cdot h_t^{gT}}{\sqrt{d}}) \cdot h_t^g) \tag{11}$$

since the length of the conversation is L. Finally, we can obtain the integrated memory which is denoted as Y_L.

3.3 Utterance Generation

To solve the long-dependency problem which often occurs in a multi-turn conversations, we apply self-attention mechanism to get the final representation of the target conversation, expressed as X. We merge the contextual distribution of the target conversation X, and the ESM discovered from the similar conversation Y_L as the input for the decoder.

During the decoding process, we selectively copy ESM. This further solves the Out-Of-Vocabulary (OOV) [44] problem. We learn a probability pointer P, which is used to determine whether to generate or copy at the current time step.

At the time step t, the decoder state, attention distribution, and context vector are represented as s_t, a^t, and C, respectively. The target vocabulary distribution is expressed as:

$$V_s = softmax(W^v(W^w[C^t, s_t] + b^w) + b^v) \tag{12}$$

In addition, P is computed as:

$$P = \sigma(W^p(X + Y_L) + W^c C_t^t + W^s s_t + b^p) \tag{13}$$

Therefore, we get the final vocabulary distribution V_f:

$$V_f = P * V_s + (1 - P) * \sum_{i:w_i=w}^{l*L} a^t \tag{14}$$

W^w, b^w, W^v, b^v,W^p and b^p in the three equations above are learnable parameters, σ is sigmoid function and the $tanh$ is hyperbolic tangent function.

Finally, we use a cross-entropy loss as the objective function:

$$\begin{aligned} loss &= -\log P(U \mid D) \\ &= -\sum_{j=1}^{l} \log P(w_{ij} \mid w_{i1:j-1}, D) \end{aligned}$$

4 EXPERIMENTAL SETTINGS

In this section, we introduce the datasets used for experiments, baselines, evaluation metrics, and training details.

4.1 Datasets

We conduct our experiments on both the court debate dataset in judicial field and the customer service dataset in e-commerce field to prove the effectiveness of our proposed model. Both the datasets are constructed from real-world conversations. We have chosen the datasets belonging to quite different domains in order to prove that our model has good domain adaptability.

4.1.1 *Court Debate Dataset.*

The Court Debate Dataset (CDD) is provided by the High People's Court of one province in China. It contains 260, 190 trial multi-role conversations. All the court transcripts are manually recorded by a legal professional. In this work, we aim at generating the words spoken by the judge. Therefore, we consider the judge's utterance as the output of the model, and the previously-spoken context in the conversation is regarded as input. The details of the dataset are shown in Table 1.

4.1.2 *Jing Dong Dialogue Corpus.*

The Customer Service Dataset, Jing Dong Dialogue Corpus (JDDC) [4], has been published as a part of JD contest[4]. Analogously to the CDD dataset, the tested

[4] http://jddc.jd.com/auth_environment

Table 1: Statistics of Court Debate Dataset

	Conversation	Utterance	length(avg)
Train	208,152	2,869,794	7.2
Dev	26,018	364,345	7.0
Test	26,020	371,554	7.6
Total	260,190	3,605,693	——

* The length represents the length of the utterance equal to the number of its words.

Table 2: Statistics of Jing Dong Dialogue Corpus

	Conversation	Utterance	length(avg)
Train	261,282	3,135,377	9.4
Dev	32,660	391,983	9.1
Test	32,661	391,480	9.8
Total	326,603	3,918,840	——

* The length represents the length of the utterance equal to the number of its words.

models will learn to generate the customer service employee's utterances based on the input represented by the prior part of the conversation. The details of JDDC dataset are shown in Table 2.

To motivate other scholars to investigate this novel and challenging problem we release our code as well as the datasets.

4.2 Baselines

To test our model we select several representative and state-of-the-art works in text generation as the baseline methods.

- CNN-based models:
  - **ByteNet** [20]: a one-dimensional Convolutional Neural Network.
  - **ConvS2S** [10]: This approach uses the convolutional neural network as an encoder to solve the problem of long sequence training.
- RNN-based models:
  - **LSTM** [18]: a unidirectional Long Short Term Memory network.
  - **S2S+attention** [36]: a model in which the encoder encodes the input sequence, while the decoder produces the target sequence. The attention mechanism is added to force the model to learn focus on specific parts of the input sequence when decoding.
  - **PGN** [44]: a commonly used method in the tasks of text generation and automatic summarization, which utilizes the copy mechanism in the decoder to effectively solve Out-Of-Vocabulary problem.
- Transformer-based models:
  - **Transformer** [51]: a neural network using positional encoding and multi-head self-attention mechanism.
  - **DAM** [66]: a multi-turn conversation model which matches a response with its multi-turn context using dependency information based entirely on attention.
  - **ReCoSa** [62]: a multi-turn conversation model in which the self-attention mechanism is utilized to update both the context and masked response representation.
  - **Retrieval-guided** model [3]: a model in which the skeleton extraction is made by an interpretable matching model, and the following skeleton-guided response generation is accomplished by a separately trained generator.
- Hierarchical LSTM-based models:
  - **HRED** [45]: a hierarchical RNN structure which enables to simultaneously model the sentence-layer information and the conversation-layer information in multi-turn conversation.
  - **EED** [38]: a model which retrieves responses to create exemplar vectors and uses the vector to decode response.
  - **CCN** [19]: a model which uses the combination of copying the current context and the content from similar conversation.

It should be noted that the **Retrieval-guided** approach is a fusion of extractive and generative based models. In addition, the methods **EED** and **CCN** also utilize similar dialogues for modeling during training as in case of our model. It needs to be also pointed out that **CCN** and **EED** use multiple similar conversations during the training, and we chose the best result as the baseline; they are marked as **CCN**$_{best}$ and **EED**$_{best}$, respectively.

Furthermore, all the baselines were trained in the same way as the proposed model **DRMN** was trained to make fair comparison across the models.

4.3 Evaluation Measures

We adopt two evaluation methods to evaluate the performance of all the tested models.

4.3.1 Automatic Evaluation. Automatic evaluation adopts quantitative evaluation metrics commonly used in text generation tasks: BLEU [39] and ROUGE [26]. We regard BLEU and ROUGE scores as objective evaluation, serving as the measures of method performance [47]. We report ROUGE-1, ROUGE-L and BLEU to understand the performance of each model and their advantages as well as disadvantages.

4.3.2 Human Evaluation. In order to ensure the fluency and rationality of the generated utterances, we also qualitatively analyzed data through human evaluation. We hired five well-educated NLP researchers to evaluate the quality of the generated utterances. We evaluated the effect of independent evaluation from two aspects: Relevance and Fluency [21, 67]:

- Relevance: the generated utterance is logically relevant to the conversation context and can provide meaningful information.
- Fluency: Generated utterance is fluent and grammatical.

We randomly selected 300 examples from the test set for each model. For either aspect, we set three levels with scores: +2, +1, 0, in which higher score stands for excellent. To compute the final scores from 5 annotators, we remove the highest score and the lowest score given by the annotators and then calculated the average of the remaining three scores. We report the average score and coefficient κ which indicates the consistency of evaluation among annotators.

4.4 Training Details

For representing utterances, we set the dimensions of word embedding as 300 and use word2vec to build the initial word vectors. For the role information, the dimension of the role embedding is set to 100 with random initialization. In the encoder, the **DRMN** is implemented by two-layer LSTM networks with a hidden size of 300. In this case, a combination of forward and backward LSTM

Table 3: Quantitative Evaluation. We report ROUGE-1 (R-1), ROUGE-L (R-L), and BLEU scores for each tested method.

model	CDD			JDDC		
	R-1	R-L	BLEU	R-1	R-L	BLEU
ByteNet [10]	33.68	32.99	16.91	22.19	18.35	11.55
ConvS2S [20]	35.92	31.48	16.34	26.53	21.08	11.64
LSTM [18]	30.28	28.02	9.77	19.45	18.74	9.52
S2S+attention [36]	36.91	33.12	18.52	28.44	22.34	13.42
PGN [44]	37.03	34.25	18.75	29.78	24.06	14.37
Transformer [51]	37.59	34.93	18.58	27.25	22.75	11.29
DAM [66]	38.28	35.27	20.83	28.86	23.79	13.95
ReCoSa [62]	38.53	35.38	20.95	30.83	24.67	14.94
Retrieval-guided [3]	37.27	34.75	19.26	28.75	22.28	12.89
HRED [45]	38.22	35.74	20.71	28.01	23.28	13.86
EED$_{best}$ [38]	39.28	37.55	22.43	32.18	30.07	18.11
CCN$_{best}$ [19]	41.10	39.82	24.75	34.17	32.37	19.53
DRMN$_{top-1}$	43.79	39.23	23.11	35.98	32.71	22.08
DRMN$_{top-2}$	44.68	40.51	27.27	**36.31**	33.19	23.37
DRMN$_{top-3}$	**45.03**	**43.09**	**28.96**	36.15	**33.35**	**23.42**

Table 4: Qualitative Evaluation. We report the average scores (Avg) and calculate the κ values for relevance and fluency.

model	CDD				JDDC			
	Relevance		Fluency		Relevance		Fluency	
	Avg	κ	Avg	κ	Avg	κ	Avg	κ
ByteNet [10]	0.63	0.62	1.01	0.71	0.59	0.55	1.19	0.67
ConvS2S [20]	0.64	0.51	1.05	0.82	0.67	0.71	1.13	0.56
LSTM [18]	0.54	0.48	0.93	0.61	0.53	0.52	1.09	0.59
S2S+attention [36]	0.89	0.55	1.32	0.69	0.88	0.48	1.26	0.57
PGN [44]	1.06	0.64	1.47	0.72	0.96	0.69	1.52	0.53
Transformer [51]	1.02	0.71	1.41	0.65	0.83	0.56	1.42	0.73
DAM [66]	1.04	0.77	1.47	0.57	0.88	0.58	1.51	0.68
ReCoSa [62]	1.06	0.67	1.55	0.65	0.91	0.71	1.59	0.55
Retrieval-guided [3]	0.96	0.72	1.43	0.59	0.92	0.64	1.48	0.63
HRED [45]	1.01	0.49	1.43	0.62	0.73	0.71	1.47	0.54
EED$_{best}$ [38]	1.11	0.63	1.62	0.73	1.07	0.65	1.65	0.54
CCN$_{best}$ [19]	1.12	0.66	1.69	0.68	1.01	0.72	**1.77**	0.70
DRMN$_{top-1}$	1.13	0.75	1.68	0.74	1.01	0.64	1.69	0.79
DRMN$_{top-2}$	1.12	0.64	1.71	0.69	1.05	0.68	1.73	0.65
DRMN$_{top-3}$	**1.15**	0.62	**1.74**	0.62	**1.09**	0.67	1.72	0.63

gives us 600 dimensions. The dropout is set to 0.8. Based on these settings, we optimize the objective function with the learning rate of $5e-4$. We perform the mini-batch gradient descent with a batch size of 32. During the experiment, we adopted cross-validation to ensure the rationality of the model.

5 EXPERIMENTAL RESULTS

5.1 Overall Performance

In this section we conduct the analysis of results from diverse perspectives to thoroughly evaluate the performance of the proposed model. One thing to note before delving into details is that to prove the impact of ESM module in our **DRMN** model, we applied different numbers of similar conversations, i.e., utilizing the most similar conversation (top-1), the top two similar ones (top-2), and the top three similar ones (top-3).

Quantitative Comparison against baselines. The quantitative performance of all the tested methods is reported in Table 3. As Table 3 shows, the proposed approach **DRMN**$_{top-3}$ significantly outperforms all the baselines in ROUGE and BLEU metrics over the two datasets.

Compared with the first three groups of baselines (CNN-based, RNN-based and Transformer-based), **DRMN** with its variants perform significantly better than these models by a large margin. The group of Hierarchical LSTM-based models shows better performance than the the first three groups. It demonstrates the effectiveness of the hierarchical infrastructure for modeling conversations by capturing word-level and sentence-level dependencies, which can further improve the quality of the generated text.

Moreover, we notice the higher performance of the two baseline methods **EED**$_{best}$ and **CCN**$_{best}$ compared to the other baselines. It indicates the advantage of leveraging similar conversations in the task of text generation. Compared with the infrastructures of these two best baselines, **DRMN** can gradually memorize keywords, phrases, and sentences from similar dialogues to assist the generation of the target conversation. **CCN** highly relies on the copy mechanism on the decoder side, which tends to cause the problem of copy position error during the generation process, as well as its limitation in copying the keywords from remote sentences. On the other hand, **EED** uses only similar dialogues as a knowledge-assisted generation, and cannot extract key words, phrases, and sentences from similar dialogues.

Qualitative Comparison against baselines. The quantitative performance of all the tested methods is reported in Table 4. To be fair, for each input, we shuffled the output generated by all the models and then let the annotators evaluate them. As noted earlier, κ indicates the consistency of the annotator's evaluation. The observed κ coefficient values that range between 0.48 and 0.82 indicate middle and upper agreement. We found that the relevance and fluency compared to the best performing baseline model (CCN) increased by 2.7% and 2.9% in the CDD, respectively. For the JDDC dataset, although the fluency of our qualitative evaluation is lower than that of CCN dataset, the relevance was still improved by 7.9%.

The impact of the number of similar conversations used. We observe the increasing performance as the number of referred similar conversations increases (see the results of **DRMN**$_{top-1}$, **DRMN**$_{top-2}$, **DRMN**$_{top-3}$ in Table 3). As mentioned above, the prediction of the model appears to improve along with the increase in the number of similar conversations, indicating that similar conversations play an important role in reading and memory. However, the increase of the number of similar conversations also adds a certain degree of complexity to train the model. It makes the time cost of model training higher as well as results in larger space cost. In order to balance the effectiveness and the training cost, we choose at most three similar conversations (top-3) for experiments to verify the validity of our model.

Comparison based on datasets. In addition, we compare the results obtained for the two datasets. We observe that the results on the CDD dataset are better than the ones on JDDC both in the quantitative and qualitative evaluations. After manually investigating the contents of our datasets, we concluded the following three possible reasons. First, compared with the court debate scenario, customer service conversations are much more open due to a large number of types and aspects of the commodities. It makes the task of text generation more difficult. Second, the utterances

Table 5: Ablation study: quantitative evaluation.

model	CDD			JDDC		
	R-1	R-L	BLEU	R-1	R-L	BLEU
TC+SC	38.71	37.09	21.86	31.75	24.92	15.65
-ESM	37.53	35.29	19.02	29.06	23.72	13.74
DRMN$_{top-1}$	43.79	39.23	23.11	35.98	32.71	22.08
DRMN$_{top-2}$	44.68	40.51	27.27	36.31	33.19	23.37
DRMN$_{top-3}$	**45.03**	**43.09**	**28.96**	**36.15**	**33.35**	**23.42**

Table 6: Ablation study: qualitative evaluation.

model	CDD				JDDC			
	Relevance		Fluency		Relevance		Fluency	
	Avg	κ	Avg	κ	Avg	κ	Avg	κ
TC+SC	1.09	0.81	1.63	0.48	0.96	0.61	1.66	0.62
-ESM	1.03	0.58	1.51	0.71	0.79	0.53	1.49	0.66
DRMN$_{top-1}$	1.13	0.75	1.68	0.74	1.01	0.64	1.69	0.79
DRMN$_{top-2}$	1.12	0.64	1.71	0.69	1.05	0.68	**1.73**	0.65
DRMN$_{top-3}$	**1.15**	0.62	**1.74**	0.62	**1.09**	0.67	1.72	0.63

of customers and customer service staff tend to be more colloquial, while the judge's utterances are more strict and formal. We note that colloquial sentences may cause difficulties for language model training due to a large variety of non-standard expressions. Last, in the JDDC dataset, there is higher number of the same or similar utterances. For example, the phrases like "Welcome back again!", "Can I help you?" appear quite repetitively.

5.2 Ablation test

To assess the contribution of ESM module and similar conversations, we next conduct the ablation tests. To prove the effectiveness of ESM module, we remove it from the **DRMN** model (removing the entire second part in Figure 2), denoted as **-ESM**. To justify our way of integrating the similar conversations with the target one, we simply concatenate them together as input (denoted as **TC+SC**), rather than modeling them in an interactive way as we have done for **DRMN**. Tables 5 and Table 6 report the evaluation scores in terms of the quantitative and qualitative analysis, respectively.

According to the results shown in Tables 5 and Table 6, we notice a dramatic decrease in the performance of **-ESM** (decrease of 34.3% on CDD and 41.3% on JDDC, as measured by BLEU score). Similarly, the variant **TC+SC** has also experienced a large decrease in performance but less than that in **-ESM**. It shows that similar dialogues still play a certain role, but compared with the way of interactive modeling in **DRMN**, a straightforward concatenation **TC+SC** has limited effect. This confirms the effectiveness of the proposed circular reading and memory module.

5.3 Case study

To help with better understanding of our model's performance, we demonstrate two cases in Figure 3. The figure shows the results obtained by different models such that the left side represents the target conversation, and the right side represents its similar conversation in the dataset. We show the ground truth utterance (in purple color) as well as the utterances generated by different models. We also highlight the utterance delivered by our model and the relevant context in the similar case (in red color). We can first observe that ESM can be extracted either from a single sentence (such as the CDD example in Figure 3, for which the ESM comes from the third last sentence of a similar conversation), or it can originate from multiple sentences (such as the JDDC example in Figure 3, in which the ESM comes from the last two sentences of the similar conversation).

As shown in the Figure 3, compared with the best performing baselines, especially **CCN** and **EED**, our model can better capture important entities, phrases, as well as the sentences from similar conversations. For example, "tax number" and "need to provide the invoice header" are recalled from similar conversations. Since the memory module in **DRMN** uses a self-attention mechanism, parallel calculations can be performed for long sentences. Hence the long-term memory can be achieved.

We next verify whether the performance improvements are obtained thanks to the detected relevant similar conversations. We analyze the attention weights (Y_L in Figure 2) of the similar conversations for the first example of our case study. As shown in Figure 4, the darker the color, the higher the weight of the word is, and the greater is the impact on the context (it means these words have higher importance). We can observe that **DRMN** selects keywords by assigning them high weights; these words are accurately memorized.

5.4 Error analysis

To explore the limitations of our model, we also analyze the generated utterances, summarize the problems that occur, and explore the optimization solutions.

After conducting statistical analysis, we found that **DRMN** performs worse for low-frequency utterances/keywords or ones that do not appear in the target conversation or similar conversation. In particular, there are 43% errors[5] belonging to this case in the JDDC dataset. For example, in the sentence *"Your order was successfully intercepted"*, the word *"intercepted"* is a low-frequency word, and it has not appeared in the target conversation neither in the similar conversation. Similarly, in the CDD dataset, such a problem caused 45% of errors, e.g., in sentences like: *"Why was the application for investigation and evidence collection not submitted until today?"*, *"application"* and *"collection"* are in low-frequency in legal trial scenario. In addition, 42% of errors in the JDDC dataset occur when specific attributes of products are mentioned which tend to appear sparsely in the dataset, e.g., *"The hard disk capacity of this computer is 500G, and the memory is 16G"*. Finally, it is worth mentioning that the proposed model has worse performance when generating long sentences.

Table 7 shows the statistics of the cases with fluency score equal to 0 for two best performing baselines and our proposed model. We can observe that all the tested models face significant difficulties in generating long sentences. Among all the utterances with a fluency score equal to 0, the proportion of long utterances[6] for **DRMN** takes up to 78.6% (85% for **CCN** and 92% for **EED**). It proves the superiority of **DRMN** even in very hard cases.

[5] The error refers here to the generated text for which either the relevance or the fluency score equals 0.

[6] I.e. the utterances with length greater than 10.

Target Case (JDDC)		Similar Case (Top1)	
	……		……
Servicer:	You need to provide the invoice header and tax number.	Customer:	Can you issue an invoice?
Customer:	Can you issue an invoice?	Servicer:	Yes, Sir.
Servicer:	Yes, sir.	Customer:	Can I only provide my tax number.
Customer:	What information do I need to provide?	Servicer:	Sir, You also need to provide the invoice header.
Ground Truth:	*Sir, you can just write the invoice header and tax number.*		
Ours-DRMN	You need to provide the invoice header and tax number.	Retrieval-guide	Is there anything else that can help you?
CCN	Yes, the invoice header is required.	Transformer	We look forward to your visit.
EED	Please give me the invoice information.	S2S+Attention	Can I help you?
ReCoSa	Is this an electronic invoice?	PGN	Can I help you?
DAM	What information needs to be provided?	Cons2s	Sir, can I help you?
HRED	Our products have a floating policy.	ByteNet	See you.
Target Case (CDD)		**Similar Case (Top1)**	
	……		……
Judge:	Did you have any economic contacts before?	Plaintiff:	The defendant's loan principal is still #number .
Plaintiff:	No.	Judge:	Has the defendant repaid you interest?
Judge:	How much did the defendant loan?	Plaintiff:	He paid two times last year.
Plaintiff:	The defendant's loan about number.	Judge:	What about the subsequent interest?
Ground Truth:	*Whether the defendant has repaid principal and interest?*		
Ours-DRMN	Has the defendant repaid you interest?	Retrieval-guide	Defendant <personname> remittance location?
CCN	How much the defendant's loan?	Transformer	What is the defendant's occupation?
EED	Whether the plaintiff provided evidence?	S2S+Attention	Does the plaintiff have anything to add?
ReCoSa	Whether the defendant has questions to ask the plaintiff?	PGN	Does the plaintiff have anything to add?
DAM	Plaintiff questioned the evidence?	Cons2s	Final statement.
HRED	Plaintiff continues to provide evidence.	ByteNet	Does the plaintiff have anything to add?

Figure 3: Case Study. We take two examples (target cases shown on the left side) from the judicial data and customer service data. We then show the following ground truth utterance as well as the utterances generated by different models. In addition, similar cases are displayed on the right. DRMN can memorize similar conversation information and accurately locate related entity, phrases, as well as sentences in similar conversations through the current conversation logic (the red text represents generated utterances that also appeared in similar conversations, and which our model can accurately remember.).

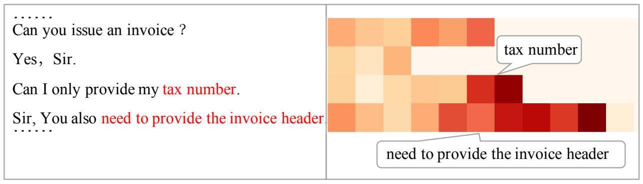

……
Can you issue an invoice ?
Yes，Sir.
Can I only provide my tax number.
Sir, You also need to provide the invoice header
……

Figure 4: Visual analysis graph: the diagram on the right shows the significance of the DRMN model for memorizing words in similar conversations based on the sub-part of the example shown in Figure 3; the depth of the color represents the importance of words and the darker the color, the greater the weight of the word.

Table 7: Error analysis for the tested models on the cases with fluency score equal to 0.

Model	DRMN	CCN	EED
#long utterance/ratio	11/78.6%	17/85%	23/92%
#short utterance/ratio	3/21.4%	3/15%	2/8%

The improvement of the pre-training language models and constructing retrieval database could be promising approaches for future research to address the above-mentioned problems.

6 CONCLUSION AND FUTURE WORK

The motivation behind our work is to improve the efficiency of conversation generation in specific domains. In particular, we propose a novel neural network structure called Deep Reading Memory Networks (DRMN), which enhances the expression of the model by reading and memorizing similar dialogues, as well as improving the quality of the generated text. Unlike the prior research, the proposed approach does not need to leverage any external knowledge, thus maintaining high field adaptability. We conduct experiments on two different datasets with both quantitative and human evaluation to validate the effectiveness of our proposed model. Experimental results indicate DRMN's superiority when compared with a number of existing state-of-the-art text generation models, which suggests that the Deep Reading Memory Networks can successfully improve the conversation generation performance.

In the future, we will further investigate other content generation problems by leveraging multi-granularity memorizing and copying mechanism. The current study serves as the methodological foundation for this goal.

ACKNOWLEDGMENTS

This work has been supported by the National Key R&D Program of China (2018YFC0830200; 2018YFC0830206; 2018YFC0830700 and 2020YFC0832505).

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *stat* 1050 (2016), 21.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[3] Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *EMNLP-IJCNLP*. 1866–1875.

[4] Meng Chen, Ruixue Liu, Lei Shen, Shaozu Yuan, Jingyan Zhou, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. The JDDC Corpus: A Large-Scale Multi-Turn Chinese Dialogue Dataset for E-commerce Customer Service. In *Proceedings of the 12th Language Resources and Evaluation Conference*. 459–466.

[5] Eric Chu, Prashanth Vijayaraghavan, and Deb Roy. 2018. Learning Personas from Dialogue with Attentive Memory Networks. In *EMNLP*. 2638–2646.

[6] Richard Csaky, Patrik Purgai, and Gabor Recski. 2019. Improving Neural Conversational Models with Entropy-Based Data Filtering. In *ACL*. 5650–5669.

[7] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmad, and Li Deng. 2017. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. In *ACL*. 484–495.

[8] Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *ACL*. 38–43.

[9] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *JMLR* (2017), 1243–1252.

[11] Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic Memory Induction Networks for Few-Shot Text Classification. In *ACL*. 1087–1094.

[12] 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*.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[14] Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. 2015. Learning to transduce with unbounded memory. In *NIPS*. 1828–1836.

[15] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACL*.

[16] Moonsu Han, Minki Kang, Hyunwoo Jung, and Sung Ju Hwang. 2019. Episodic Memory Reader: Learning What to Remember for Question Answering from Streaming Data. In *ACL*. 4407–4417.

[17] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. In *ACL*. 1766–1776.

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

[19] Changzhen Ji, Xin Zhou, Yating Zhang, Xiaozhong Liu, Changlong Sun, Conghui Zhu, and Tiejun Zhao. 2020. Cross Copy Network for Dialogue Generation. In *EMNLP*. 1900–1910.

[20] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099* (2016).

[21] Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *ACL*. 1499–1508.

[22] Mahmoud Khademi. 2020. Multimodal Neural Graph Memory Networks for Visual Question Answering. In *ACL*. 7177–7188.

[23] Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In *ACL*. 567–582.

[24] Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative Memory-Based Joint Open Information Extraction. In *ACL*. 5871–5886.

[25] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020. MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning. In *ACL*. 2603–2614.

[26] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[27] Zehao Lin, Xinjing Huang, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Task-Oriented Conversation Generation Using Heterogeneous Memory Networks. In *EMNLP-IJCNLP*. 4558–4567.

[28] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *ACL*. 1489–1498.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[30] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

[31] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *ACL*. 1468–1478.

[32] Sameen Maruf and Gholamreza Haffari. 2018. Document Context Neural Machine Translation with Memory Networks. In *ACL*. 1275–1284.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *NIPS*.

[34] Alexander H Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *EMNLP*.

[35] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *ACL*. 845–854.

[36] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *SIGNLL*.

[37] Dai Quoc Nguyen, Tu Nguyen, and Dinh Phung. 2020. A Relational Memory-based Embedding Model for Triple Classification and Search Personalization. In *ACL*. 3429–3435.

[38] Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *ACL*. 1329–1338.

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

[40] Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. *arXiv preprint arXiv:1809.05524* (2018).

[41] Maarten Sap, Eric Horvitz, Yejin Choi, Noah A. Smith, and James Pennebaker. 2020. Recollection versus Imagination: Exploring Human Memory and Cognition via Neural Language Models. In *ACL*. 1970–1978.

[42] Ilham Fathy Saputra, Rahmad Mahendra, and Alfan Farizki Wicaksono. 2018. Keyphrases Extraction from User-Generated Contents in Healthcare Domain Using Long Short-Term Memory Networks. In *Proceedings of the BioNLP 2018 workshop*. 28–34.

[43] Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review* 21, 2 (2006), 97–126.

[44] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.

[45] Iulian 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*.

[46] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364* (2015).

[47] Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149* (2016).

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

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

[50] Zhiliang Tian, Wei Bi, Dongkyu Lee, Lanqing Xue, Yiping Song, Xiaojiang Liu, and Nevin L. Zhang. 2020. Response-Anticipated Memory for On-Demand Knowledge Integration in Response Generation. In *ACL*. 650–659.

[51] 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 *NIPS*. 5998–6008.

[52] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* (2015).

[53] Joseph Weizenbaum. 1966. ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Commun. ACM* (1966).

[54] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[55] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*.

[56] Jason D Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21, 2 (2007), 393–422.

[57] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive Human-Machine Conversation with Explicit Conversation Goal. In *ACL*. 3794–3804.

[58] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *NIPS*. 1784–1794.

[59] Jun Xu, Haifeng Wang, Zhengyu Niu, Hua Wu, and Wanxiang Che. 2020. Knowledge Graph Grounded Goal Planning for Open-Domain Conversation Generation.. In *AAAI*. 9338–9345.

[60] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*. 1480–1489.

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

[62] Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *ACL*.

[63] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded Conversation Generation as Guided Traverses in Commonsense Knowledge Graphs. In *ACL*. 2031–2043.

[64] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *COLING*.

[65] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *ACL*. 654–664.

[66] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *ACL*. 1118–1127.

[67] Qingfu Zhu, Lei Cui, Wei-Nan Zhang, Furu Wei, and Ting Liu. 2019. Retrieval-Enhanced Adversarial Training for Neural Response Generation. In *ACL*. 3763–3773.