Coupled Context Modeling for Deep Chit-Chat: Towards Conversations between Human and Computer

Rui Yan[1,2]
[1]Institute of Computer Science and Technology
Peking University
Beijing 100080, China
ruiyan@pku.edu.cn

Dongyan Zhao[1,2]
[2]Beijing Institute of Big Data Research
Beijing 100871, China
zhaody@pku.edu.cn

ABSTRACT

To have automatic conversations between human and computer is regarded as one of the most hardcore problems in computer science. Conversational systems are of growing importance due to their promising potentials and commercial values as *virtual assistants* and *chatbots*. To build such systems with adequate intelligence is challenging, and requires abundant resources including an acquisition of big conversational data and interdisciplinary techniques, such as content analysis, text mining, and retrieval. The arrival of big data era reveals the feasibility to create a conversational system empowered by data-driven approaches. Now we are able to collect an extremely large number of human-human conversations on Web, and organize them to launch human-computer conversational systems. Given a human issued utterance, i.e., a *query*, a conversational system will search for appropriate *responses*, conduct relevance ranking using *contexts* information, and then output the highly relevant result. In this paper, we propose a novel context modeling framework with end-to-end neural networks for human-computer conversational systems. The proposed model is general and unified. In the experiments, we demonstrate the effectiveness of the proposed model for human-computer conversations using p@1, MAP, nDCG, and MRR metrics.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; • **Computing methodologies → Natural language processing**; *Discourse, dialogue and pragmatics*;

KEYWORDS

Context modeling; conversational system; retrieval model

ACM Reference Format:
Rui Yan[1,2] and Dongyan Zhao[1,2]. 2018. Coupled Context Modeling for Deep Chit-Chat: Towards Conversations between Human and Computer. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220045

1 INTRODUCTION

Conversational systems have great potentials and promising commercial values as personal assistants, agent systems, and social chatbots as real world applications. Since the Web 2.0 is so popular nowadays, people are getting used to having conversations—in public—on a variety of websites, such as forums, social medium (e.g., *Facebook*, *Twitter*) and community question-answering platforms (e.g., *Baidu Zhidao*, *Yahoo! Answers*). There are abundant resources of natural conversations occurring on these websites, which indicate a unique opportunity to launch conversational systems based on the massive data repository.

The big data era accelerates fast progress of conversational research in the open domain based on information retrieval technology. Owing to the huge data on the Web, a conversational system is able to find at least some appropriate results for any user inputs. The data-driven system can learn to flow conversations by analyzing the pattern of the large volume of human-to-human conversations. In this paper, we focus on research for the human-computer conversational system in the open domain by retrieval techniques.

Generally speaking, there are two conversation scenarios between human and computer. We have either single-turn conversations, or multi-turn conversations. Single-turn conversations indicate the basic assumption using only the *query* to find a *response*. A more practical scenario is the multi-turn conversations with *contexts* (a.k.a., previous utterances within the current conversation session). For multi-turn conversations, the contexts can provide auxiliary information to identify responses which are appropriate to respond. We show a motivation case to use contexts to understand the query q and to find a response r in Table 1.

Here we see that contexts (s_1-s_4) are useful to understand a query q to talk about "Disney in Shanghai". To make good use of contexts requires fine-grained strategies because 1) some contexts are irrelevant to q (e.g., s_1 and s_2) and 2) relevant contexts are not equally important (e.g., s_3 and s_4).

The most important challenge for context modeling is to maintain relevant and important information while filter out other information so that proper ranking evidence is used to find appropriate responses [46]. Previously, context

KDD '18, August 19–23, 2018, London, United Kingdom
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5552-0/18/08...$15.00
https://doi.org/10.1145/3219819.3220045

Table 1: A motivation example for context modeling from human-to-human conversations.

s_1	A: 中午吃太撑了 (I ate too much for lunch)
s_2	B: 晚上少吃点 (Don't eat too much for dinner then)
s_3	A: 我突然想起来一个上海有个好玩的地方 (I suddenly think of an interesting place in Shanghai)
s_4	B: 哪儿? (where?)
q	A: 那边迪士尼刚开业 (Disney just started business there)
r	B: 要不咱们下个月去上海玩吧? (Shall we go to Shanghai to have fun next month?)

modeling is conducted via sentence sequences [22, 29, 46, 47]. In this paper, we provide a novel angle to model contexts as utterance matching sequences with external memory propagation. It is better to distinguish relevant/irrelevant information for multi-turn human-computer conversations.

To be more specific, we propose a series of coupled recurrent neural network chains with Long-Short Term Memory (LSTM) units while each coupled LSTM chain indicates the semantic matching between two adjacent utterances (which could be the contexts, the query or a candidate response). Within each coupled LSTM chain, the information across the words is propagated through chains and literally denotes *local*-level matching from two adjacent sentences. We also propose another LSTM chain to record the sentence matching sequences, which in contrast indicates the *global*-level matching. In this way, relevant and important information will be "remembered" while otherwise will be "forgotten" through an end-to-end learning framework. We tackle the challenge by Coupled Context Modeling (CCM) through the LSTM chains from both *local* and *global* levels. The CCM model offers a novel insight.

We build a human-computer conversational system upon a large conversation resource (in millions). The system shortlists multiple candidate responses given a query using the standard retrieval technique. Then we match the response with the query and the contexts through the deep neural network-based CCM; the CCM model thereafter tells how each candidate is likely to respond the query under the context scenario. We conduct extensive experiments in a variety of human-computer conversation setups and evaluate the performance in terms of p@1, MAP, nDCG and MRR metrics. We run experiments against several rival algorithms.

To sum up, we have several contributions as follows:

- We characterize contexts as a sequence of sentence matchings rather than a sequence of sentences. To the best of our knowledge, we are the first to investigate the novel coupled context modeling strategy.
- In the coupled context modeling, we propose an end-to-end neural network learning framework to organize the functional components together to rank all evidence from local-level and global-level matchings.

We introduce the task statement in Section 2. Next, we describe the model, devise experimental setups, and discuss results in Sections 2 and 3. Related work is reviewed in Section 4. We draw conclusions in Section 5.

2 TASK STATEMENT AND MODELS

2.1 Problem Formulation

Over time, notable accomplishments in conversational systems have been achieved with tremendous efforts devoted. Concretely there is a well-defined paradigm for human-computer conversations and the classic paradigm has been applied to most of existing conversational systems [37, 46]: given a human utterance as the *query*, the computer returns a *response*. Given a query q from the human, the computer would retrieve several candidate responses r to respond. Especially for multi-turn conversations, we have the context information available which comes from the previous utterances within the current conversation context $\mathcal{C} = \{s_1, s_2, \ldots, s_n\}$. Using the contexts, we are aware of the background information of the query, and accordingly rank responses better.

For the query q, a response r and the context $\mathcal{C}$, we feed them into the proposed neural network framework for sentence representation learning and semantics matching. The ranking score $\mathcal{F}(.)$ is a learned function from all evidence through an end-to-end learning process. The calculation of $\mathcal{F}(.)$ is elaborated in Section 2.5. The best-matched r will be output to respond q. In contrast to the simple single-turn conversation formulation as $r^\star$=argmax$_r$ $\mathcal{F}(r|q)$, we formulate the multi-turn conversation problem as follows:

$$\begin{aligned} r^* &= \underset{r}{\text{argmax}}\, \mathcal{F}\big(r|q, \mathcal{C}\big) \\ &= \underset{r}{\text{argmax}}\, \mathcal{F}\big(r|q, s_1, s_2, \ldots, s_n\big) \end{aligned}$$

To learn the representation and matching is the core component in the conversational system. We establish a coupled context modeling framework to integrate all ranking evidence through an end-to-end learnable metric. We will first introduce the big picture of the framework and then elaborate the details.

2.2 Model Overview

In general, the proposed coupled context modeling framework can be divided into two hierarchies: 1) local-level matchings between two adjacent utterances, and 2) a global-level matching to integrate the sentence matching sequence, which is illustrated in Figure 1.

For matchings between two utterances, two sentences can be matched in two ways: 1) one is to match via a transformation of an affinity matrix through parameters trained and learned from the network; 2) the other one is to use the chain-based matching through information propagated in the recurrent neural networks (RNNs) sequence with Long-Short Term Memory (LSTM) units [49]. Here we use the second option. Information is propagated along the word sequence. RNNs keep a hidden state vector, which changes

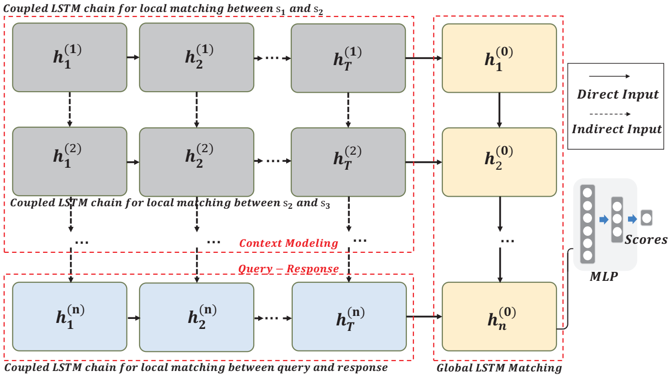

Figure 1: Model overview: coupled LSTM chains for *local* matchings between adjacent utterances from contexts, the query and a candidate response, a *global* LSTM chain to integrate the matching sequence, and a multi-layer perceptron (MLP). To be concise, we omit some model details and will illustrate more in Figure 2.

according to the input at each time step. Such an LSTM chain-based method has been proved effective for matching and alignments [10, 21]. More details will be introduced in the next subsection.

On the lower hierarchy of local matchings, each coupled LSTM chain denotes matching between two adjacent utterances. The last coupled LSTM chain measures the matching between the query and a candidate response. If no other coupled LSTM chain is taken into account, the model degenerates to the *single-turn* conversation scenario, which is connected to a chain-based matching method in [10].

As to *multi-turn* conversations, the contexts are all modeled as coupled LSTM chains. Yet, these coupled LSTM chains should not be isolated. Two consecutive coupled LSTM chains share part of the chain: for instance, coupled LSTM chain concatenating s_{i-1} and s_i (from head to tail) and the coupled LSTM chain concatenating s_i and s_{i+1} (from head to tail) have s_i in common. As a result, the antecedent chain can have a positive impact (when appropriately matched), or negative penalty (when mismatched), on the subsequent chain. In other words, the next LSTM chain will be influenced by an external "memory" from the previous LSTM chain and the memory is a soft alignment mechanism for the information propagation. Such an alignment is an indirect influence from the last chain to the next chain. Hence, we use dotted arrows in Figure 1.

On the higher hierarchy of the global matching, we have a sequence of matching scores with the utterance order preserved, we use a standard LSTM chain to record these matchings and decide relevant information to "remember" and irrelevant information to "forget". With the last hidden state of the global LSTM obtained, the information of the context modeling is updated. We feed the vector to an ensuing network for further information mixing.

2.3 Local-Level Coupled LSTM Chains

We move on to the coupled LSTM chains for the local-level matchings. The gist is similar to chain-based LSTM sequence. The antecedent coupled LSTM chain has an impact by propagating information through a soft gating mechanism on the subsequent coupled LSTM chain as mentioned. We will first explain the external "memory" from the previous LSTM chain. We formulate a relevance vector through a gating cell into the neural network units in the next LSTM chain so as to control information fusion.

When concatenating the first two context sentences s_1 and s_2 into a coupled LSTM chain in this way, we obtain a series of state $\{\mathbf{h}_t^{(1)}\}$ with a superscript $^{(1)}$ from the first coupled LSTM chain, and the final hidden state $\mathbf{h}_T^{(1)}$ which indicates the information mixing from s_1-s_2. The hidden states $\{\mathbf{h}_t^{(1)}\}$ will be utilized to characterize the next coupled LSTM chain. We take the output vector $\mathbf{h}_T^{(1)}$ as a local-matching feature representation and feed it to the global-level LSTM on the higher hierarchy for future integration and optimization. The gradients are computed using the back-propagation algorithm.

2.3.1 External Memories.

Suppose we obtain external memories constructed by history hidden states $\{\mathbf{h}_t^{(z-1)}\}$ from the (z-1)-th LSTM chain for s_{z-1} and s_z, which is:

$$\mathrm{M}^{(z-1)} = \{\mathbf{h}_1^{(z-1)}, \mathbf{h}_2^{(z-1)}, \ldots, \mathbf{h}_T^{(z-1)}\}$$

where $\mathbf{h}_t^{(z-1)}$ is the hidden state at time t emitted by the (z-1)-th coupled LSTM chain. T means the ending step. The memory blocks M are used to store the external information for matching. Hence, the history information can be read from the memory block. We denote a relevance vector, denoted as $\mathbf{r}^{(z)}$, from external memories as $\mathbf{r}_i^{(z)}$, which can be computed by soft attention mechanisms:

$$\mathbf{r}_i^{(z)} = \sum_1^T \alpha_{ij} \mathbf{h}_j^{(z-1)} \tag{1}$$

where α represents the attention distribution over the external memory states $\mathrm{M}^{(z-1)}$.

The attention signal can be calculated as:

$$\alpha_{ij} = \mathrm{softmax}\big(\phi(\mathbf{h}_j^{(z-1)}, \mathbf{x}_i^{(z)}, \mathbf{h}_{i-1}^{(z)})\big) \tag{2}$$

where

$$\phi(\mathbf{h}_j^{(z-1)}, \mathbf{x}_i^{(z)}, \mathbf{h}_{i-1}^{(z)}) = \mathbf{v}^T \tanh(W \cdot \Big[\mathbf{h}_j^{(z-1)}, \mathbf{x}_i^{(z)}, \mathbf{h}_{i-1}^{(z)}\Big] + \mathbf{b}) \tag{3}$$

W denotes the weight matrix, $\mathbf{v}$ is weight vector and $\mathbf{v}^T$ denotes its transpose. Here we use superscripts $^{(z-1)}$ and $^{(z)}$ to denote the vectors from the (z-1)-th coupled LSTM chain and the (z)-th LSTM chain respectively. The value is based on how to control the information fusion from the previous relevance matchings. The attention schema is parametrized as a neural network which is jointly trained with all the other components [21, 23].

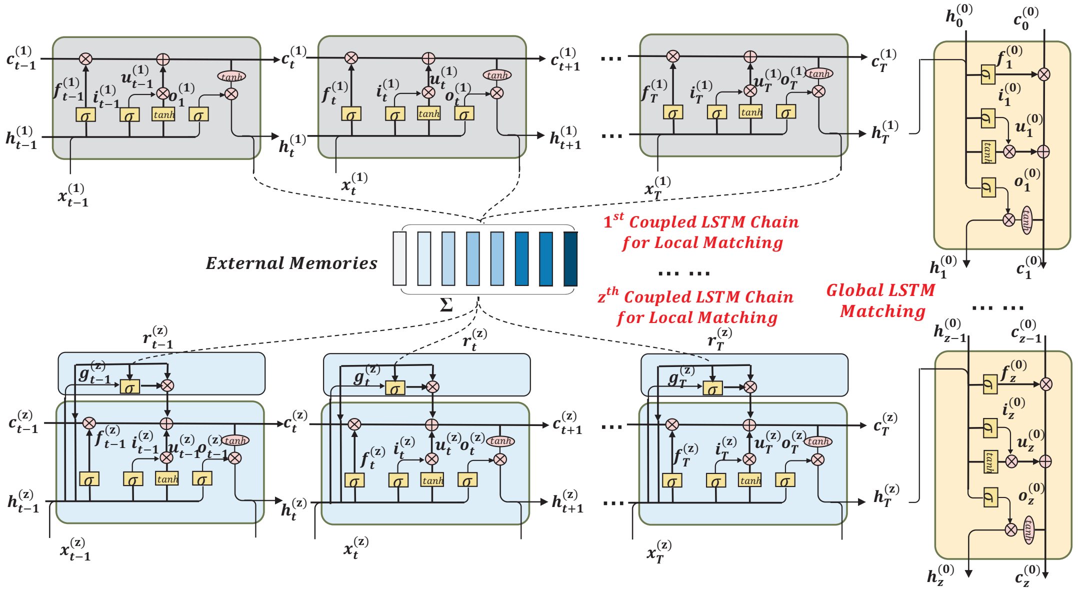
External Memories
Σ
1st Coupled LSTM Chain for Local Matching
… …
zth Coupled LSTM Chain for Local Matching
Global LSTM Matching
… …
tanh
σ

Figure 2: Coupled Context Modeling: coupled LSTM chains indicate *local* matching between two adjacent utterances. The local matching sequence with temporal orders is put into the *global* matching LSTM chain for further information mixing. The output from the global LSTM chain is fed to an MLP shown in Figure 1.

2.3.2 Gating Cell. With additional information of the external memory blocks, we have more evidence to measure the relevance between two utterances. The information from an antecedent LSTM chain is maintained and passed along for further judgments of the subsequent LSTM chain. Our motivation is to propagate "reliable" relevance evidence from the previous coupled LSTM chain to measure the next one. To this end, we design a gating cell to read relevance memories into the next coupled LSTM chain.

The gating cell is to control the information fusion into the subsequent LSTM chain: relevance evidence can pass through the gate and irrelevant evidence shall not [49]. In this way, the semantic relevance from the antecedent chain implicitly performs as the clue for the selection of the subsequent chain. Starting from the original vectors, at each time step the cell decides the alignment with the previous LSTM chain, and decides what information should be retained for future time steps and discards the others. This cell plays the role of gating relevant information fusion, and the semantic clues can be integrated into the next LSTM chain smoothly.

We add a new gating cell, and the relevance information is incorporated into the original LSTM units by fusing vectors of $\mathbf{g}$ and $\mathbf{r}$. The modified LSTM cells read the relevance vector, and decides whether or how to use the external memory from the antecedent coupled LSTM chain into the subsequent one. We model how gates work in LSTMs, illustrated in Equations (4)-(7).

After updating the new LSTM units, the hidden state of $\mathbf{h}_T^{(z)}$ will be passed as the output of the z-th LSTM chain.

$$\begin{bmatrix} \mathbf{i}_t^{(z)} \\ \mathbf{f}_t^{(z)} \\ \mathbf{o}_t^{(z)} \\ \mathbf{g}_t^{(z)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} (W \cdot \left[\ \mathbf{h}_{t-1}^{(z)}, \mathbf{x}_t^{(z)}, \mathbf{r}_t^{(z)} \ \right] + \mathbf{b}) \tag{4}$$

$$\mathbf{u}_t^{(z)} = \tanh(W \cdot \left[\mathbf{h}_{t-1}^{(z)}, \mathbf{x}_t^{(z)}, \mathbf{r}_t^{(z)}\right] + \mathbf{b}) \tag{5}$$

$$\mathbf{c}_t^{(z)} = \mathbf{f}_t^{(z)} \odot \mathbf{c}_{t-1}^{(z)} + \mathbf{i}_t^{(z)} \odot \mathbf{u}_t^{(z)} + \mathbf{g}_t^{(z)} \odot \mathbf{r}_t^{(z)} \tag{6}$$

$$\mathbf{h}_t^{(z)} = \mathbf{o}_t^{(z)} \odot \tanh(\mathbf{c}_t^{(z)}) \tag{7}$$

2.4 Global-Level LSTM Chain

Now we have the final states denoted as $\mathbf{h}_T^{(1)}, \mathbf{h}_T^{(2)}, \ldots, \mathbf{h}_T^{(n)}$ where T denotes their ending steps correspondingly. The final states represent the sequence with the utterance order maintained in the outputs from the *local*-level matching layer (n pairs). On the higher hierarchy, a *global*-level LSTM takes $\mathbf{h}_T^{(1)}, \mathbf{h}_T^{(2)}, \ldots, \mathbf{h}_T^{(n)}$ as its input and encodes the matching sequences as hidden states, namely $\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, \ldots, \mathbf{h}_n^{(0)}$. This global LSTM has two functions: 1) it models the dependency and the temporal relationship of utterances in the conversation session; 2) it leverages the temporal relationship to supervise the accumulation of the *query-response* matching as a context-aware matching. In particular, the global LSTM discards irrelevant context information to the query. The detailed parameterizations are similar to Equations (4)-(7)

with different inputs and different meanings:

$$\begin{bmatrix} \mathbf{i}_t^{(0)} \\ \mathbf{f}_t^{(0)} \\ \mathbf{o}_t^{(0)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (W \cdot \begin{bmatrix} \mathbf{h}_{t-1}^{(0)}, \mathbf{h}_t^{(t)} \end{bmatrix} + \mathbf{b}) \tag{8}$$

$$\mathbf{u}_t^{(0)} = \tanh(W \cdot \left[\mathbf{h}_{t-1}^{(0)}, \mathbf{h}_t^{(t)}\right] + \mathbf{b}) \tag{9}$$

$$\mathbf{c}_t^{(0)} = \mathbf{f}_t^{(0)} \odot \mathbf{c}_{t-1}^{(0)} + \mathbf{i}_t^{(0)} \odot \mathbf{u}_t^{(0)} \tag{10}$$

$$\mathbf{h}_t^{(0)} = \mathbf{o}_t^{(0)} \odot \tanh(\mathbf{c}_t^{(0)}) \tag{11}$$

Through Equation (8)-(11), we can see how the gates control the information from the previous hidden state and the current input flows to the current hidden state. Thus, important matching vectors (corresponding to important and relevant utterances from the conversation context) can be accumulated while noise in the vectors can be filtered out.

The vector of the last state obtained from the global-level LSTM is passed through a 3-layer, fully-connected, feed-forward neural network, also known as *multi-layer perceptron* (MLP) [1], which allows rich interactions. The network enables to extract features automatically, starting from lower-level representations to higher-level ones, till the system provides an overall judgment of the appropriateness for the response given the query and the conversation contexts. The matching process with the ranking scores is denoted as $\mathcal{F}(.)$.

2.5 Training

We have two options to formulate the task-specific output. We can formulate the whole task as a ranking problem. For ranking tasks, the outputs are scalar matching scores. We can also formulate the task as a classification problem, when the outputs are the probabilities of the different classes, which are computed by the softmax function on the matching vectors [49]. In this paper, we use the first formulation of a ranking problem. A single neuron outputs the matching score as $\mathcal{F}(r|q, \mathcal{C})$. As mentioned, $\mathcal{F}(.)$ is in $\mathbb{R}$; the final scoring neuron is essentially a linear regression after the MLP layer.

Here we apply hinge loss to train the proposed network. Given a pair of positive sample (q, r^+) in the training set, we randomly sample a negative instance r^-. The objective is to maximize the scores of positive samples while minimizing that of the negative samples. Concretely, we would like the score of positive samples $\mathcal{F}(r^+|\cdot)$ to be as least the score of negative samples $\mathcal{F}(r^-|\cdot)$ plus a margin Δ. Thus, the training objective is to

$$\min_{\theta} \sum_{(q,r^+)} \max\left\{0, \Delta + \mathcal{F}(r^+|\cdot) - \mathcal{F}(r^-|\cdot)\right\} + \lambda\|\theta\|_2^2 \tag{12}$$

where we add an ℓ_2 penalty with a coefficient λ (empirically set) for all the parameters θ which are weight and bias values optimized by the network from all ranking evidence.

3 EXPERIMENTS AND EVALUATION

3.1 Experimental Setups

3.1.1 Data. We crawled a large number of human-to-human conversations from the open Web [46, 47], where the users publish a *message* visible to the public, and then receive multiple subsequent *replies* to their utterances. We removed messages without any replies, conducted data filtering and cleaning procedures by removing extremely short utterances and those of low linguistic quality [45]. After data pre-processing, we have 7,293,978 (*message-reply*) pairs in all extracted [46, 47]. The data are used for the retrieval purpose only. Some statistics are summarized in Table 2.

Table 2: Statistics of the conversation resources in the retrieval repository and model training.

Source	#Message	#Reply	#Vocabulary
Zhidao	8,915,694	3,705,302	1,499,691
Douban	10,618,981	2,963,226	483,846
Tieba	4,189,160	3,730,248	1,046,130
Weibo	186,963	393,654	119,163
Misc.	3,056	1,548	4,297

	Train	Validation	Test
#conversation samples	1,606,583	357,018	11,097
#responses per sample	2	2	20
#“+” response per sample	1	1	4.05
#max turn per context	59	37	31
#avg turn per context	6.16	5.98	6.01

Moreover, we train the model with 1,606,583 conversation samples, 357,018 for validation, and 11,097 for testing. The data distribution for model training is also shown in Table 2. Note that it is important that the dataset for learning does NOT overlap with the data for retrieval: we strictly comply with the machine learning paradigm. For each training and validation sample, we randomly chose inappropriate responses to obtain negative samples. Validation was based on the model accuracy.

In Table 3, we show what the original data look like. The first message is typically unique. There are many flexible ways to “respond”, which is exactly the nature of real conversations: various responses are all possibly appropriate. The interactions can be of a single turn or multiple turns.

We illustrate how to extract the original data into *message-reply* pairs as the retrieval repository, shown in Table 4 (I). In the retrieval repository, a subsequent *reply* has a responding relationship to the antecedent *message*, and each pair can be regarded as an atomic conversation of two utterances. We can also extract positive training samples with contexts for model learning, shown in Table 4 (II). A data repository from public human conversations is demonstrated to be a rich resource to facilitate human-computer conversations based on retrieval [37, 40, 46, 47].

3.1.2 Hyperparameters. We use 500-dimensional word embeddings, and they were initialized randomly and learned during training. As our dataset is in Chinese, we performed standard Chinese word segmentation based on the language platform. We maintained a vocabulary of 177,044 phrases by choosing those with more than 2 occurrences.

Table 3: An example of the original data of a message and all replies. We anonymize user information.

User 1:我明天要开始自驾横穿中国的旅行 (Tomorrow I'll start a journey driving across China.)
User 2:大神求带! (Oh my God take me with you!) **User 3**:旅行路线规划好啦?大家环行路线都差不多吧 (Have you scheduled your route? Is the route more or less the same for everyone?) **User 4**:太羡慕你了，这也是我最大的梦想，没有之一 (I envy you so much. This could be my biggest dream.) **User 1**:@User 3已经定好了，最好环线能穿越西安和成都 (All set. It is best to travel through Xi'an and Chengdu.)

Table 4: An illustration of how to extract retrieval repository and triples for model training/testing.

(I) Retrieval repository.
Message 我明天要开始自驾横穿中国的旅行 (Tomorrow I'll start a journey driving across China.)
Reply大神求带! (Oh my God take me with you!)
Message我明天要开始自驾横穿中国的旅行 (Tomorrow I'll start a journey driving across China.)
Reply太羡慕你了，这也是我最大的梦想，没有之一 (I envy you so much. This could be my biggest dream.)

(II) Multi-turn samples.
我明天要开始自驾横穿中国的旅行 (Tomorrow I'll start a journey driving across China.)
旅行路线规划好啦?大家环行路线都差不多吧 (Have you scheduled your route? Is the route more or less the same for everyone?)
已经定好了，最好环线能穿越西安和成都 (All set. It is best to travel through Xi'an and Chengdu.)
[The conversation continues...]

The LSTM units have 300 hidden units for each dimension. We used stochastic gradient descent (with a mini-batch size of 100) for optimization, gradient computed by standard back propagation. Initial learning rate was set to 0.8, and a multiplicative learning rate decay was applied. The above parameters were chosen empirically. We used the validation set for early stopping. For fairness, all methods including baselines are tuned in this way.

3.1.3 Evaluation Metrics.

For the test set, we hired annotators on a crowdsourcing platform to judge the appropriateness of candidate responses for each query. Each sample was judged by at least 7 annotators via majority voting based on the *appropriateness* for the response given the query and contexts: "1" denotes an appropriate response and "0" indicates an inappropriate one [46, 49]. The manual evaluation is highly human-oriented. We filter out samples with extremely low agreement. Due to the highly subjective human judgments for conversations, a kappa score between 0.3 and 0.4 of the annotation for all valid samples indicates moderate agreements among annotators.

Given the ranking lists of responses, we evaluated the performance in terms of precision@1 (p@1), mean average precision (MAP) [30], and normalized discounted cumulative gain (nDCG) [5]. Since the system outputs the highest ranked results, p@1 is the precision at the 1st position, and should be the most natural way to indicate the fraction of suitable result among the top-1 response provided. We also provide the top-20 ranking list for all test queries using nDCG and MAP, which test the potential for a system to provide more than one appropriate responses. We aimed at selecting as many appropriate candidates as possible into the top-20 list and rewarding methods that return suitable responses on the top.

Moreover, since we use real conversations for testing, we have the original human response taken from the human-human conversation session as 1 candidate response in the ranking list along with the other 19 retrieved candidates to measure Mean Reciprocal Rank (MRR) scores. Unlike MAP and nDCG, which examine the ranks of all appropriate responses, MRR focuses on evaluating the capability to highly rank the original "ground truth" response. MRR is useful but does not test the full capability because there can be more than one appropriate responses other than the "ground truth" response to fulfill a conversation.

Since p@1, nDCG, MAP and MRR are standard metrics, we omit their definitions and calculations to save spaces in this paper. Interested readers may refer to Manning *et al.* [15] for more details.

3.2 Competing Algorithms

We include several algorithms as baselines to compare. Since our proposed approach is technically a retrieval method, we mainly focus on retrieval-based conversation systems. For completeness, we also include two typical generation methods for comparisons. Given a query during a conversation, different systems provide different response accordingly.

3.2.1 Generation-based Methods.

For this group of algorithms, the conversational system will generate a response from a given input, rather than to retrieve from a repository.

- *Neural Responding Machine* (NRM). We implement the neural responding machine proposed in [23], which is an RNN-based sequence-to-sequence generation approach with no context information considered.
- *Hierarchical Recurrent Encoder-Decoder* (HRED). HRED is a context-aware response generator [22]. Utterances are formulated in two hierarchies of word- and sentence-level.

3.2.2 Retrieval-based Methods.

We focus on more matching algorithms, since we establish a retrieval-based system.

- *Okapi BM25.* We include the standard retrieval technique to rank candidates. For each query, we retrieve the most relevant results using BM25 model [15].
- *ARC-II.* There are many convolutional kernel based sentence matching method [4, 14]. The ARC-II approach is a typical neural network based method with convolutionary layers which construct sentence representations and produce the final matching scores via a MLP layer [4].

Table 5: Overall performance against baselines. '$\star$' indicates that we accept the improvement hypothesis of CCM over the best baseline at a significance test level of 0.01. For generative methods, they generate one response given each query. Other metrics except p@1 are not applicable.

	Methods	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
1	NRM [23]	0.465					
	HRED [22]	0.543					
2	Okapi BM25	0.272	0.253	0.337	0.302	0.368	0.169
	ARC-CNN [4]	0.394	0.294	0.397	0.421	0.477	0.232
	LSTM-RNN [18]	0.338	0.283	0.330	0.371	0.431	0.228
	Chain-LSTM [10]	0.416	0.328	0.413	0.429	0.450	0.301
3	ROCF [47]	0.711	0.412	0.651	0.666	0.702	0.321
	MRS [51]	0.720	0.414	0.655	0.669	0.711	0.328
	DL2R [46]	0.731	0.417	0.663	0.682	0.717	0.333
	SMN [40]	0.738	0.420	0.674	0.688	0.725	0.347
	Coupled Context Modeling	**0.765***	**0.433***	**0.693***	**0.708***	**0.739***	**0.356***

Table 6: Model variations by distinguishing direct/indirect inputs (i.e., removing external memory gatings).

	p@1	MAP	nDCG@5	nDCG@10	nDCG@20	MRR
-Indirect Input	0.750	0.420	0.682	0.699	0.721	0.347
Full Model	0.765	0.433	0.693	0.708	0.739	0.356

- *LSTM-RNN.* A sentence is encoded as a vector representation by the last hidden state from an LSTM-RNN sequence. Two sentences are matched by cosine similarity in pairs [18].
- *Chain-LSTM.* The Chain-LSTM indicates another matching style. When concatenated as a chain, the first sentence helps to model the second one. The information from both sentences interweaves sentence modeling and matching [10].
- *Rank Optimized Conversation Framework* (ROCF). The ROCF model is a rank optimized framework to combine context-insensitive ranking and context-aware ranking [47].
- *Multi-view Response Selection* (MRS). MRS [51] is a hierarchical matching model for retrieval-based conversational systems. The representation learning of utterances is based on two hierarchies of word-level and utterance-level.
- *Deep Learning-to-Respond* (DL2R). DL2R uses a query reformulation framework to add context utterances with different context utilization strategies [46].
- *Sequential Matching Network* (SMN). SMN is a sequential matching network while utterances are formulated as several matchings for evidence integration [40].
- *Coupled Context Modeling* (CCM). In this paper, we propose a novel context modeling framework which incorporates coupled LSTM chains of matching sequences on two hierarchies of the local level and the global level.

3.3 Results

3.3.1 Overall Performance. We compare the performance of all methods including baselines and our proposed CCM method measured in terms of the evaluation metrics. In Table 5 we list the overall results for these methods.

- The first group of baselines are typical generation-based methods, with contexts (i.e., HRED) or without contexts (i.e., NRM). In general, generative methods provide one response with the largest likelihood during the generation process. Many other candidate generations are not always guaranteed to be legitimate sentences [46]. Hence we do not compare MAP or nDCG for this algorithm group. Note that the original response is not likely to be generated; thus it is infeasible to calculate the MRR. In general, the generative algorithms have good p@1 scores, but the generated responses are often ambiguous, universal and lack of diversity or flexibility [23]. HRED is better than NRM when contexts are incorporated, which concurs the observation in [22].
- The second group of baselines are strong matching algorithms for short texts. No context information will be taken into account. *Okapi BM25* represents the standard (and simple) retrieval system by utilizing the shallow representation of terms only. Deep learning systems are demonstrated to have stronger capabilities to learn the abstractive representation [1, 7, 25]. The deep learning algorithms clearly outperform the shallow learning method. Measuring the interactions of each and every term from the two sentences at every position can be ascribed as a classic matching style. Word-to-word matching in every position is better than the matching from a single position (i.e., *LSTM-RNN*). The *Chain-LSTM* method is regarded as another matching style, and is demonstrated to be useful in matching tasks such as sentence entailment [21]. We have similar observations. The MRR score indicates that Chain-LSTM is good at capturing information flows through a conversation.
- The last group is for context-aware matching methods, an algorithm family where our proposed method lie in. From Table 5, we can see that that context information is rather useful for human-computer conversation systems. All context-aware methods outperform context-insensitive baselines. We believe the contexts have rich information to

interpret semantics from the conversation background, while context-insensitive baselines only utilize a single query. It is natural to incorporate contexts for multi-turn conversations.

We compare the context-aware methods in detail. For ROCF, the contexts are used as a whole piece of text while for DL2R, the contexts are used in some reformulation and fusion ways. MRS uses a hierarchical structure to distinguish sentence-level representations from word-level representations. These strategies lead to useful representation learning. SMN learns to match utterances in a sequential matching manner, which performs better than other three baselines.

The CCM model extends the sequential matching SMN, and shows a better performance against ROCF, MRS, DL2R and SMN. We design a new context modeling method by modeling *hierarchical* "matching sequences". The sequential modeling is different from a simple integration in ROCF or a combination of all possible utterances in DL2R since the order of the conversation session is incorporated. Besides, hierarchical modeling from *local*- and *global*-level sequential matchings for information integration also leads to improvements in contrast to SMN and MRS, which do not have such matching architectures.

3.3.2 Analysis. We have multiple LSTM chains, and we check the model with different components and variants. For the CCM variants, the first variant is to degenerate the full model without any contextual LSTM chains, which is actually the basic Chain-LSTM model included in the baselines. We omit this variant due to strict page limits. Moreover, we degenerate the model by removing the gating cell with external memories, which means removing the indirect inputs from the external memory (please refer to Figure 1). We observe that performance drops compared to the full model. This phenomenon indicates the contextual LSTM chains and the influence between the coupled LSTM chains are both useful for the learning task. The performance drops when we remove the gating cell, which explains that cells work as soft attention from external memories.

4 RELATED WORK

During the past years, people have worked on conversational systems between human and computer. In early days, researchers generally focus on dialogue systems established by rules or templates [36, 38]. Rule-based ideas are simple and such methods require no data or few data for model learning. Instead, these methods require huge human expertise to build handcrafted rules or templates to launch the conversational system. Therefore, to build rule-based systems is time consuming. Still a conversation might go out of scope from time to time. Data-driven methods have attracted the attention from the research community.

From human-driven conversation systems to data-driven conversation systems, much more data are in need. Nowadays, with the prosperity of social media (such as microblogs), forums and other Web 2.0 resources, people are getting used to have conversations with each other in public on the Internet. It is now possible to collect a very large amount of human-to-human conversation data [37].

With substantially increasing data, it is straightforward to build a retrieval-based conversational system as information retrieval techniques are developing fast. The system takes a user utterance (a.k.a., a query), and then searches for candidate responses by certain matching metrics. Leuski *et al.* build such a systems to select the most suitable response to the query from the question-answer pairs using a statistical language model as cross-lingual information retrieval [8]. The database consisting of a number of question-answer pairs is a key to success [9]. Researchers propose to augment the database with question-answer pairs retrieved from plain texts [2, 17].

A second way to launch a conversational system is to make use of language generation techniques. Previously, Higashinaka *et al.* propose to combine language template generation with the search-based methods [3]. Ritter *et al.* have investigated the feasibility of conducting short text conversation by using statistical machine translation (SMT) techniques, learning from millions of naturally occurring conversation data in Twitter [20]. In these approaches, a response is generated from a model, not retrieved from a repository.

In recent years, deep neural networks (DNNs, also known as *deep learning*) brings huge impacts for natural language related techniques. DNNs can extract underlying abstract features of data automatically by exploring multiple layers of non-linear transformation [1]. Prevailing DNNs for sentence-level modeling include convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs have a fixed-size sliding window to capture local patterns of successive words [7], whereas RNNs keep one or a few hidden states, and collect information along the word sequence in an iterative fashion [42]. Due to a well-known problem of gradient vanishing/explosion in vanilla RNNs, Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are proposed to address the issue [18, 31].

Due to deep learning, generation-based conversation systems are greatly advanced. In general, the conversational system applies the sequence-to-sequence generation manner [31]. A neural responding machine is proposed for single-turn conversation [23]. Soon, the conversation is extended into multi-turns, with plain contexts [29] and hierarchical contexts [22]. Researchers gradually introduce various elements into conversation generation, such as diversity [12, 27, 28, 32], topics [41], proactive suggestions [48], additional contents [16, 50], and quality control [24], which is calibrated with learnable evaluation [33, 35].

Retrieval-based conversation systems are also greatly advanced using neural networks. A series of information retrieval-based methods are applied to short-text conversations using microblog data, either for single-turn conversation [6, 11, 14] or multi-turn conversation [46, 47]. Basically, these methods model sentences using convolutional [4, 14] or recurrent [18] units to construct abstractive representations. Besides, many matching metrics are proposed for retrieval using deep neural networks. Palangi *et al.* have proposed sentence matching based on vector similarities [18]. Usually, sentences are compared in a pairwise matching style via word-by-word matchings, known as sentence pair modeling [4, 44]. The chain-based matching is also demonstrated to be useful, where the first sentence's information is available when modeling the second one [10, 21]. In this paper, we use the chain-based matching with recurrent units. Auxiliary information such as topics [39] and contents [13] can also be incorporated for better retrieval. Although not all of these methods are originally designed for conversation, they are very effective for short-text matching tasks in general. We have included some of these methods as strong baselines in the experimental setups.

There are some interesting studies for conversational systems as well. Researcher ensemble generation-based and retrieval-based conversational system so that two systems could be better than one [19, 26]. A recent survey paper introduces more about conversational systems [43].

In particular, we revisit all the mentioned context-aware method from both retrieval-based systems or generative conversation systems. They are all based on sentence sequences with orders maintained [22, 40, 47] or without orders [29, 46]. Contextual utterances are modeled with hierarchies on word- and sentence-levels [22, 34, 51] or without hierarchies at all [29, 46, 47]. The difference compared with related work is quite clear: we model contexts as a sequence of local sentence matchings, and integrate such a sequence by a global sequence. To this end, our proposed CCM model lie on two hierarchies from local/global-levels rather than sentence/word levels, which is also a new mechanism.

5 CONCLUSION

To sum up, we propose a novel context modeling of matching sequences via coupled LSTM chains for multi-turn human-computer conversational systems. Each coupled LSTM chain indicates a semantic matching between two adjacent utterances. The information through chains denotes local-level matchings. The matching information from the lower-level is propagated to a higher-level integration for further mixing. We tackle the challenge in context modeling through the (coupled) LSTM chains from local and global levels using an end-to-end learnable framework. The learning model is effective: we examine the effect of the CCM method against a series of baselines. Our model consistently outperforms the baselines in terms of p@1, MAP, nDCG, and MRR.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC No. 61672058). Rui Yan was sponsored by CCF-Tencent Open Research Fund and Microsoft Research Asia (MSRA) Collaborative Research Program.

REFERENCES

[1] Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2, 1 (2009), 1–127.

[2] Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding Question-answer Pairs from Online Forums. In *SIGIR*. 467–474.

[3] Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open domain conversational system fully based on natural language processing. In *COLING*.

[4] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. 2042–2050.

[5] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[6] Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An Information Retrieval Approach to Short Text Conversation. *CoRR* abs/1408.6988 (2014).

[7] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[8] Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2009. Building effective question answering characters. In *SIGDIAL*. 18–27.

[9] Anton Leuski and David Traum. 2011. NPCEditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine* 32, 2 (2011), 42–56.

[10] Chaozhuo Li, Yu Wu, Wei Wu, Chen Xing, Zhoujun Li, and Ming Zhou. 2016. Detecting Context Dependent Messages in a Conversational Environment. In *COLING'16*. 1990–1999.

[11] Hang Li and Jun Xu. 2014. Semantic matching in search. *Foundations and Trends in Information Retrieval* 8 (2014), 89.

[12] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL'16*. 110–119.

[13] Xiang Li, Lili Mou, Rui Yan, and Ming Zhang. 2016. Stalemate-Breaker: A Proactive Content-Introducing Approach to Automatic Human-Computer Conversation. In *IJCAI'16*. 2845–2851.

[14] Zhengdong Lu and Hang Li. 2013. A Deep Architecture for Matching Short Texts. In *NIPS*. 1367–1375.

[15] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[16] 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'16*. 3349–3358.

[17] Elnaz Nouri, Ron Artstein, Anton Leuski, and David R Traum. 2011. Augmenting Conversational Characters with Generated Question-Answer Pairs. In *AAAI Fall Symposium: Question Generation*.

[18] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2015. Deep Sentence Embedding Using the Long Short Term Memory Network: Analysis and Application to Information Retrieval. *arXiv preprint arXiv:1502.06922* (2015).

[19] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. Alime chat: A sequence to sequence and rerank based chatbot engine. In *ACL'17*. 498–503.

[20] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven Response Generation in Social Media. In *EMNLP*.

[21] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. In *ICLR*.

[22] 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'16*. 3776–3783.

[23] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL-IJCNLP'15*. 1577–1586.

[24] Mingyue Shang, Zhenxin Fu, Nanyun Peng, Yansong Feng, Dongyan Zhao, and Rui Yan. 2018. Learning to Converse with Noisy Data: Generation with Calibration. In *IJCAI'18*.

[25] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*.

[26] 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'18*.

[27] Yiping Song, Zhiliang Tian, Dongyan Zhao, Ming Zhang, and Rui Yan. 2017. Diversifying Neural Conversation Model with Maximal Marginal Relevance. In *IJCNLP'17*. 169–174.

[28] Yiping Song, Rui Yan, Yansong Feng, Yaoyuan Zhang, Dongyan Zhao, and Ming Zhang. 2018. Towards a Neural Conversation Model with Diversity Net Using Determinantal Point Processes.. In *AAAI'18*.

[29] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *NAACL'15*. 196–205.

[30] Hiroaki Sugiyama, Toyomi Meguro, Ryuichiro Higashinaka, and Yasuhiro Minami. 2013. Open-domain Utterance Generation for Conversational Dialogue Systems using Web-scale Dependency Structures. In *SIGDIAL*. 334–338.

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

[32] Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism. In *IJCAI'18*.

[33] Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. RUBER: An unsupervised method for automatic evaluation of open-domain dialog systems.. In *AAAI'18*.

[34] 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'17*. 231–236.

[35] Xiaowei Tong, Zhenxin Fu, Mingyue Shang, Dongyan Zhao, and Rui Yan. 2018. One "Ruler" for All Languages: Multi-Lingual Dialogue Evaluation with Adversarial Multi-Task Learning. In *IJCAI'18*.

[36] Marilyn A. Walker, Rebecca Passonneau, and Julie E. Boland. 2001. Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems. In *ACL*. 515–522.

[37] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A Dataset for Research on Short-Text Conversations.. In *EMNLP*.

[38] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *SIGDIAL*. 404–413.

[39] Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2016. Topic Augmented Neural Network for Short Text Conversation. *arXiv preprint arXiv:1605.00090* (2016).

[40] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL'17*. 496–505.

[41] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic Aware Neural Response Generation.. In *AAAI'17*, Vol. 17. 3351–3357.

[42] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*.

[43] Rui Yan. 2018. "Chitty-Chitty-Chat Bot": Deep Learning for Conversational AI. In *IJCAI'18*.

[44] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline Generation Through Evolutionary Trans-temporal Summarization. In *EMNLP'11*. 433–443.

[45] Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet Recommendation with Graph Co-ranking. In *ACL'12*. 516–525.

[46] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR'16*. 55–64.

[47] Rui Yan, Yiping Song, Xiangyang Zhou, and Hua Wu. 2016. Shall I Be Your Chat Companion?: Towards an Online Human-Computer Conversation System. In *CIKM'16*. 649–658.

[48] Rui Yan and Dongyan Zhao. 2018. Smarter Response with Proactive Suggestion: A New Generative Neural Conversation Paradigm. In *IJCAI'18*.

[49] Rui Yan, Dongyan Zhao, and Weinan E. 2017. Joint Learning of Response Ranking and Next Utterance Suggestion in Human-Computer Conversation System. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. 685–694.

[50] Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards Implicit Content-Introducing for Generative Short-Text Conversation Systems. In *EMNLP'17*. 2190–2199.

[51] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view Response Selection for Human-Computer Conversation.. In *EMNLP'16*. 372–381.