A Hybrid Retrieval-Generation Neural Conversation Model

Liu Yang [*1] Junjie Hu[2] Minghui Qiu[3] Chen Qu [1] Jianfeng Gao[4] W. Bruce Croft[1]
Xiaodong Liu[4] Yelong Shen[5] Jingjing Liu[4]
[1] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
[2] Language Technologies Institute, Carnegie Mellon University [3] Alibaba Group
[4] Microsoft Research Redmond [5] Tencent AI Lab
{lyang,chenqu,croft}@cs.umass.edu,junjieh@cs.cmu.edu,minghui.qmh@alibaba-inc.com
{jfgao,xiaodl,jingjl}@microsoft.com,yelongshen@tencent.com

ABSTRACT

Intelligent personal assistant systems that are able to have multi-turn conversations with human users are becoming increasingly popular. Most previous research has been focused on using either retrieval-based or generation-based methods to develop such systems. Retrieval-based methods have the advantage of returning fluent and informative responses with great diversity. However, the performance of the methods is limited by the size of the response repository. On the other hand, generation-based methods can produce highly coherent responses on any topics. But the generated responses are often generic and not informative due to the lack of grounding knowledge. In this paper, we propose a hybrid neural conversation model that combines the merits of both response retrieval and generation methods. Experimental results on Twitter and Foursquare data show that the proposed model outperforms both retrieval-based methods and generation-based methods (including a recently proposed knowledge-grounded neural conversation model [8]) under both automatic evaluation metrics and human evaluation. We hope that the findings in this study provide new insights on how to integrate text retrieval and text generation models for building conversation systems.

ACM Reference Format:
Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, Jingjing Liu. 2019. A Hybrid Retrieval-Generation Neural Conversation Model. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357881

1 INTRODUCTION

The fast development of artificial intelligence has enabled many intelligent personal assistant systems, such as Amazon Alexa, Apple Siri, Alibaba AliMe, Microsoft Cortana, Google Now and Samsung Bixby.[1] As a natural interface for human computer interaction, conversation systems have attracted the attention of researchers in the Information Retrieval (IR), Natural Language Processing (NLP) and Machine Learning (ML) communities, leading to a rapidly growing field referred to as Conversational AI [7].

Typical task-oriented dialog systems use a modularized architecture which consists of a natural language understanding module, a dialog state tracker, a dialog policy learning module, and a natural language generation module [11]. In recent years, fully data-driven end-to-end conversation models have been proposed to reduce hand-crafted features, rules or templates. These methods could be grouped into two different categories: generation-based approaches [19, 31, 33, 35, 37] and retrieval-based approaches [15, 41–43, 45].

Given some conversation context, retrieval-based models try to find the most relevant context-response pairs in a pre-constructed conversational history repository. Some of these methods achieve this in two steps: 1) retrieve a candidate response set with basic retrieval models such as BM25 [32] or QL [27]; and 2) re-rank the candidate response set with neural ranking models to find the best matching response [39, 41–43, 45]. These methods can return natural human utterances in the conversational history repository, which is controllable and explainable. Retrieved responses often come with better diversity and richer information compared to generated responses [34]. However, the performance of retrieval-based methods is limited by the size of the conversational history repository, especially for long tail contexts that are not covered in the history. Retrieval-based models lack the flexibility of generation-based models, since the set of responses of a retrieval system is fixed once the historical context/response repository is constructed.

On the other hand, the generation-based methods could generate highly coherent new responses given the conversation context. Much previous research along this line was based on the Seq2Seq model [33, 35, 37], where there is an encoder to learn the representation of conversation context as a contextual vector, and a decoder to generate a response sequence conditioning on the contextual vector as well as the generated part of the sequence. The encoder/decoder could be implemented by an RNN with long short term memory (LSTM) [12] hidden units or gated recurrent units (GRU) [3]. Although generation-based models can generate new responses for a conversation context, a common problem with generation-based methods is that they are likely to generate very general or universal responses with insufficient information such as "I don't

[*] Work primarily done during Liu Yang's internship at Microsoft Research. Liu is at Google currently.

[1] For example, over 100M installations of Google Now (Google, http://bit.ly/1wTckVs); 100M sales of Amazon Alexa devices (TheVerge, https://bit.ly/2FbnzTN); more than 141M monthly users of Microsoft Cortana (Windowscentral, http://bit.ly/2Dv6TVT).

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357881

Table 1: A comparison of retrieval-based methods and generation-based methods for data driven conversation models.

Item	Retrieval-based methods	Generation-based methods
Main techniques	Retrieval models; Neural ranking models	Seq2Seq models
Diversity	Usually good if similar contexts have diverse responses in the repository	Easy to generate bland or universal responses
Response length	Can be very long	Usually short
Context property	Easy for similar context in the repository; Hard for unseen context	Easy to generalize to unseen context
Efficiency	Building index takes long time; Retrieval is fast	Training takes long time; Decoding is fast
Flexibility	Fixed response set once the repository is constructed	Can generate new responses not covered in history
Fluency	Natural human utterances	Sometimes bad or contain grammar errors
Bottleneck	Size and coverage of the repository	Specific responses; Long text; Sparse data
Informativeness	Easy to retrieve informative content	Hard to integrate external factual knowledge
Controllability	Easy to control and explain	Difficult to control the actual generated content

know", "I have no idea", "Me too", "Yes please". The generated responses may also contain grammar errors. Ghazvininejad et al. [8] proposed a knowledge-grounded neural conversation model in order to infuse the generated responses with more factual information relevant to the conversation context without slot filling. Although they showed that the generated responses from the knowledge-grounded neural conversation model are more informative than the responses from the vanilla Seq2Seq model, their model is still generation-based, and it is not clear how well this model performs compared to retrieval-based methods. A comparison of retrieval-based methods and generation-based methods for end-to-end data driven conversation models is shown in Table 1. Clearly these two types of methods have their own advantages and disadvantages, it is thus necessary to integrate the merits of these two methods.

To this end, in this paper we study the integration of retrieval-based and generation-based conversation models in an unified framework. The closest prior research to our work is the study on the ensemble of retrieval-based and generation-based conversation models by Song et. al. [34]. Their proposed system uses a multi-seq2seq model to generate a response and then adopts a Gradient Boosting Decision Tree (GBDT) ranker to re-rank the generated responses and retrieved responses. However, their method still requires heavy feature engineering to encode the context/ response candidate pairs in order to train the GBDT ranker. They constructed the training data by negative sampling, which may lead to sub-optimal performance, since the sampled negative response candidates could be easily discriminated from the positive response candidates by simple term-matching based features.

We address these issues by proposing a hybrid neural conversational model with a generation module, a retrieval module and a hybrid ranking module. The generation module generates a response candidate given a conversation context, using a Seq2Seq model consisting of a conversation context encoder, a facts encoder and a response decoder. The retrieval module adopts a "context-context match" approach to recall a set of response candidates from the historical context-response repository. The hybrid ranking module is built on the top of neural ranking models to select the best response candidate among retrieved and generated response candidates. The integration of neural ranking models, which can learn representations and matching features for conversation context-response candidate pairs, enables us to minimize feature engineering costs during model development. To construct the training data of the neural ranker for response selection, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. We evaluate our proposed approach with experiments on Twitter and Foursquare data from a previous work by Ghazvininejad et al. [8]. Experimental results show that the proposed model can outperform both retrieval-based models and generation-based models (including a recently proposed knowledge-grounded neural conversation model [8]) on both automatic evaluation and human evaluation.[2]

In all, our contributions can be summarized as follows:

- We perform a comparative study of retrieval-based models and generation-based models for the conversational response generation task.
- We propose a hybrid neural conversational model to combine response generation and response retrieval with a neural ranking model to reduce feature engineering costs.
- For model training, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. We evaluate the effectiveness of different kinds of distant supervision signals and settings for the hybrid ranking of response candidates.
- We run extensive experimental evaluation on retrieval-based, generation-based and hybrid models using the Twitter and Foursquare data. Experimental results show that the proposed hybrid neural conversation model can outperform both retrieval-based and generation-based models on both automatic evaluation and human evaluation. We also perform a qualitative analysis on top responses selected by the neural re-ranker and response generation examples to provide insights.

2 RELATED WORK

Retrieval-based Conversation Models. There have been several recent studies on retrieval based-conversation models [15, 21, 30, 39, 41–43, 45, 46, 51]. Yan et al. [41] proposed a retrieval-based conversation system with the deep learning-to-respond schema by concatenating context utterances with the input message as reformulated queries. Wu et al. [39] proposed a sequential matching network that matches a response with each utterance in the context on multiple levels of granularity to distill important matching information. Yang et al. [45] considered external knowledge beyond dialog context through pseudo-relevance feedback and

[2] Code on Github: https://github.com/yangliuy/HybridNCM

QA correspondence knowledge distillation for multi-turn response ranking. Although retrieval-based methods can return fluent responses with great diversity, these approaches lack the flexibility of generation based methods since the set of responses of a retrieval system is fixed once the historical context/ response repository is constructed in advance. Thus retrieval systems may fail to return any appropriate responses for those unseen conversation context inputs [7]. In our work, we study the integration of retrieval-based and generation-based methods for response generation to combine the merits of these two types of methods.

Generation-based Conversation Models. There have also been a number of recent studies on conversation response generation with deep learning and reinforcement learning [2, 4, 18, 19, 25, 29, 31, 33, 35–38, 48, 49]. Gao et al. [7] performed a comprehensive survey of neural conversation models in this area. Shang et al. [33] proposed the Neural Responding Machine (NRM), which is an RNN encoder-decoder framework for short text conversations. In order to mitigate the blandness problem of universal responses generated by Seq2Seq models, Li et al. [17] proposed the Maximum Mutual Information (MMI) objective function for conversation response generation. Some previous work augments the context encoder to not only represent the conversation history, but also some additional input from external knowledge. Ghazvininejad et al. [8] proposed a knowledge-grounded neural conversation model which infuses factual content that is relevant to the conversation context. Qin et al. [28] extended the knowledge-grounded neural conversation model and jointly models response generation and on-demand machine reading, which takes advantage of machine reading models, such as [20]. Our research shares a similar motivation with the work by Ghazvininejad et al. [8], but we do not adopt a pure generation-based approach. Instead, we explore a hybrid approach that combines retrieval-based models and generation-based models. Similar hybrid approaches are also used in some popular personal intelligent assistant systems including the “Core Chat” component of Microsoft XiaoIce [50]. Our proposed model distinguishes from prior work using the boosted tree ranker [34, 50] by using a neural ranking model which holds the advantage of reducing feature engineering efforts for the conversation context/ response candidates pairs during the hybrid re-ranking process.

Neural Ranking Models. A number of neural ranking models have been proposed for information retrieval, question answering and conversation response ranking [9, 13, 14, 24, 26, 39, 40, 44, 47]. These models could be classified into three categories [9, 10]. The first category is the *representation-focused* models. These models learn the representations of queries and documents separately and then calculate the similarity score of the learned representations with functions such as cosine, dot, bilinear or tensor layers. A typical example is the DSSM [14] model, which is a feed forward neural network with a word hashing phase as the first layer to predict the click probability given a query string and a document title. The second category is the *interaction-focused* models, which build a query-document pairwise interaction matrix to capture the exact matching and semantic matching information between the query-document pairs. The interaction matrix is further fed into deep neural networks which could be a CNN [13, 26, 47], term gating network with histogram or value shared weighting mechanism [9, 44] to generate the final ranking score. The neural ranking models in the third category combine the ideas of the *representation-focused* models and *interaction-focused* models to jointly learn the lexical matching and semantic matching between queries and documents [24, 47]. The neural ranking model used in our research belongs to the interaction-focused models due to their better performance on a variety of text matching and ranking tasks compared with representation-focused models [9, 13, 26, 39, 40, 44].

Table 2: A summary of key notations in this work.

u_i, $\mathcal{U}$	The context of the i-th conversation and the set of all conversation contexts
f_i^k, $\mathcal{F}_i$, $\mathcal{F}$	The k-th factual text relevant to context u_i, the factual texts relevant to context u_i and the set of all factual texts
r_i^k, $\mathcal{R}_i$, $\mathcal{R}$	the k-th retrieved response candidate to context u_i, the set of all retrieved response candidates for context u_i and the set of all retrieved response candidates
g_i^k, $\mathcal{G}_i$, $\mathcal{G}$	the k-th generated response candidate to context u_i, the set of all generated response candidates for context u_i and the set of all generated response candidates
y_i^k, $\mathcal{Y}_i$	the k-th response candidate and the union set of all the candidates for the i-th context, i.e., $y_i^k \in \mathcal{Y}$, $\mathcal{Y}_i = \mathcal{R}_i \cup \mathcal{G}_i$
y_i^*, $\mathcal{Y}^*$	The ground truth response candidate for the i-th context and the set of all ground truth response candidates
$f(\cdot)$	The neural ranking model learned in the hybrid ranking module
$f(u_i, y_i^k)$	The predicted matching score between u_i and y_i^k

3 OUR APPROACH

3.1 Problem Formulation

We define the task of conversational response generation following the previous literature [8]. We are given a conversation context $u_i \in \mathcal{U}$, where u_i is the i-th context sequence which contains one or multiple utterances. There are also F factual snippets of text $\mathcal{F}_i = \{f_i^1, f_i^2, ..., f_i^F\}$ that are relevant to the i-th conversation context u_i. Based on the conversation context u_i and the set of external facts $\mathcal{F}_i$, the system outputs an appropriate response which provides useful information to users. Figure 1 shows an example of the conversational response generation task. Given an conversation context “*Going to Din Tai Fung Dumpling House tonight!*”, we can associate it with several contextually relevant facts from a much larger collection of external knowledge text (e.g., the Wikipedia dump, tips on Foursquare, product customer reviews on Amazon, etc.). A response that is both appropriate and informative in the given example could be “*The shrimp and pork wontons with spicy sauce are amazing!*”.

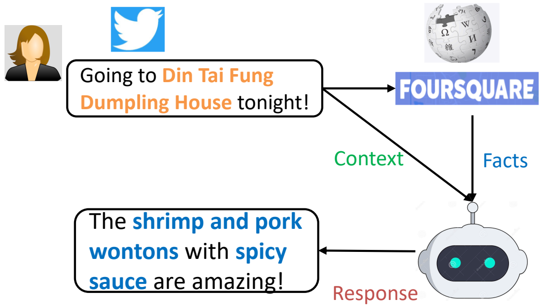
Going to Din Tai Fung Dumpling House tonight!
FOURSQUARE
Context
Facts
The shrimp and pork wontons with spicy sauce are amazing!
Response

Figure 1: An example of the conversational response generation task. The factual information from external knowledge is denoted in blue color.

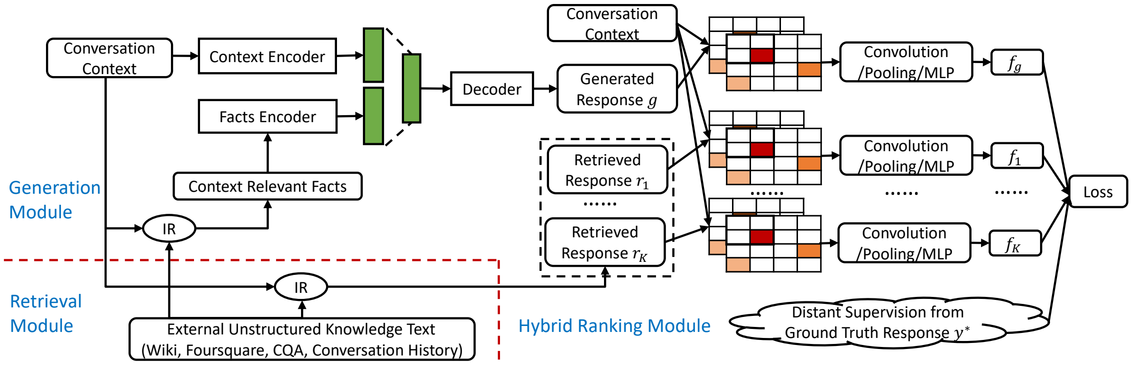

Figure 2: The architecture of the Hybrid Neural Conversation Model (HybridNCM).

3.2 Method Overview

In the following sections, we describe the proposed Hybrid Neural Conversation Model (HybridNCM) for response generation. Figure 2 shows the architecture of the hybrid neural conversation model, which consists of three modules:

(1) **Generation Module**: Given the conversation context u_i and the relevant facts $\mathcal{F}_i$, this module is to generate a set of response candidates $\mathcal{G}_i$ using a Seq2Seq model which consists of a conversation context encoder, a facts encoder and a response decoder.

(2) **Retrieval Module**: This module adopts a "context-context match" approach to retrieving a few response candidates $\mathcal{R}$. The "context-context matching" approach matches the conversation context u_i with all historical conversation context. It then returns the corresponding responses of the top ranked historical conversation context as a set of the retrieved response candidates $\mathcal{R}_i$.

(3) **Hybrid Ranking Module**: Given the generated and retrieved response candidates, i.e., $\mathcal{Y}_i = \mathcal{G}_i \cup \mathcal{R}_i$, this module re-ranks all the response candidates with a hybrid neural ranker trained with labels from distant supervision to find the best response as the final system output.

We will present the details of generating the responses for the i-th context u_i by these modules from Section 3.3 to Section 3.5. A summary of key notations in this work is presented in Table 2. We use a bold letter for a vector or a matrix, and an unbold letter for a word sequence or a set.

3.3 Generation Module

We map a sequence of words to a sequence of embeddings by looking up the indices in an embedding matrix, e.g., $\mathbf{u} = \mathbf{E}(u_i) = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{L_u}]$ where L_u is the length of a word sequence u_i.

3.3.1 *Context Encoder*.

Inspired by previous works on response generation with Seq2Seq models [8, 33, 37], we adopt a Seq2Seq architecture with attention mechanism [1, 22] in the hybrid neural conversation model. In the Seq2Seq architecture, a context encoder is used to transform a sequence of context vectors $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{L_u}]$ into contextual hidden vectors $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{L_u}]$ in Eq. (1).

$$\mathbf{h}_t = \text{RNN}(\mathbf{u}_t, \mathbf{h}_{t-1}), \tag{1}$$

where $\mathbf{h}_t \in \mathbb{R}^H$ is the hidden state at time step t. In our implementation, we stack two layers of LSTM networks as the recurrent neural network. With the context encoder, we can summarize the conversation context by the last hidden vector $\mathbf{h}_{L_u}$ and maintain the detailed information at each time step by each hidden state $\mathbf{h}_t$.

3.3.2 *Facts Encoder*.

For the facts encoder, we use the same architecture of the stacked LSTM as the context encoder in Section 3.3.1 to generate the hidden representations of relevant facts. Note that for each conversation context u_i, there are F sequences of facts $\mathcal{F} = \{f^1, f^2, \cdots, f^F\}$. We encode these facts into F sequences of hidden vectors $\{\mathbf{f}^1, \mathbf{f}^2, \cdots, \mathbf{f}^F\}$ by the stacked LSTM, where $\mathbf{f}^j = [\mathbf{f}_1^j, \mathbf{f}_2^j, \cdots, \mathbf{f}_L^j]$ and $L = |\mathbf{f}^j|$. We summarize a fact into a fixed-size vector by averaging its hidden vectors, i.e., $\bar{\mathbf{f}}^j = \text{mean}(\mathbf{f}^j)$.

3.3.3 *Response Decoder*.

The response decoder is trained to predict the next word g_t given the representations of conversation context $\mathbf{h}_{L_u}$, facts $\bar{\mathbf{f}}$, and all the previously generated words $g_{1:t-1}$ as follows:

$$p(g|u_i, \mathcal{F}) = \prod_{t=1}^{L_g} p(g_t|g_{1:t-1}, u_i, \mathcal{F}) \tag{2}$$

$$\mathbf{E} = [\mathbf{h}_1, \cdots, \mathbf{h}_{L_u}, \bar{\mathbf{f}}^1, \cdots, \bar{\mathbf{f}}^F] \in \mathbb{R}^{H\times(L_u+F)} \tag{3}$$

$$\mathbf{a}_t = \text{softmax}\left(\mathbf{E}^T \mathbf{s}_{t-1}\right) \tag{4}$$

$$\mathbf{c}_t = \mathbf{E}\mathbf{a}_t \tag{5}$$

$$\mathbf{v}_t = \tanh\left([\mathbf{s}_{t-1}, \mathbf{c}_t]\right) \tag{6}$$

$$\mathbf{s}_t = \text{RNN}(\mathbf{v}_t, \mathbf{s}_{t-1}) \tag{7}$$

$$\mathbf{s}_0 = \varphi\left(\tanh\left(\mathbf{h}_{L_u} + \frac{1}{F}\sum_{j=1}^{F} \bar{\mathbf{f}}^j\right)\right) \tag{8}$$

For the decoder, we stack two layers of LSTM networks with the attention mechanism proposed in [22]. More specifically, we concatenate the hidden vectors of a context u_i and all factual vectors into a matrix $\mathbf{E}$ in Eq. (3). We then compute the attention weight $\mathbf{a}_t$ by the dot product between the decoder's previous hidden state $\mathbf{s}_{t-1}$ and all vectors in $\mathbf{E}$, followed by a softmax function in Eq. (4). The attention context summarizes the conversation context u_i and facts $\mathcal{F}$ by the weighted sum of $\mathbf{E}$ in Eq. (5). For the input to the decoder's RNN network, we concatenate the attention context $\mathbf{c}_t$ and the previous hidden state $\mathbf{s}_{t-1}$ that summarizes the partial generated response $g_{1:t-1}$, and apply a tanh function afterwards in Eq. (6). The initial hidden vector of the decoder is initialized by the last hidden state of the context encoder and the average factual vectors in Eq. (8). $\varphi(\cdot)$ is a linear function that maps a vector from the encoder's hidden space to the decoder's hidden space. The conditional probability at the t-th time step can be computed by a linear function $\phi(\cdot)$, which is a fully connected layer, that maps the decoder's hidden state $\mathbf{s}_{t-1}$ to a distributional vector over the vocabulary, and a softmax function in Eq. (9).

$$p(g_t|g_{1:t-1}, u_i, \mathcal{F}) = \text{softmax}(\phi([\mathbf{s}_{t-1}, \mathbf{c}_t])) \tag{9}$$

where $\mathbf{s}_t$ is the hidden state of the decoder RNN at time step t.

3.3.4 ***Train and Decode.*** Given the ground-truth response y^* to a conversation context u_i with facts $\mathcal{F}$, the training objective is to minimize the negative log-likelihood over all the training data $\mathcal{L}_g$ in Eq. (10).

$$\mathcal{L}_g = -\frac{1}{|\mathcal{U}|} \sum_{y^*, u_i, \mathcal{F}} \log p(y^*|u_i, \mathcal{F}) \tag{10}$$

During prediction, we use beam search to generate response candidates and perform length normalization by dividing the output log-likelihood score with the length of generated sequences to add penalty on short generated sequences.

3.4 Retrieval Module

The retrieval module retrieves a set of response candidates from the historical conversation context-response repository constructed from the training data. It adopts a "context-context match" approach to retrieve a few response candidates. We first index all context/ response pairs in the training data with Lucene.[3] Then for each conversation context u_i, we match it with the "conversation context" text field in the index with BM25. We return the "response" text field of top K ranked context/ response pairs as the retrieved response candidates.[4] Note here we only used the context/ response pairs without the facts in the training data. We would like to keep the retrieval module simple and efficient. The re-ranking process of response candidates will be performed in the hybrid ranking module as presented in Section 3.5.

3.5 Hybrid Ranking Module

3.5.1 ***Interaction Matching Matrix.*** We combine a set of generated response candidates $\mathcal{G}_i$ and a set of retrieved response candidates $\mathcal{R}_i$ as the set of all response candidates $\mathcal{Y}_i = \mathcal{G}_i \cup \mathcal{R}_i$. The hybrid ranking module re-ranks all candidates in $\mathcal{Y}_i$ to find the best one as the final system output. In our implementation, $\mathcal{G}_i$ contains one generated response and $\mathcal{R}_i$ contains K retrieved responses.[5] Note that facts are not used in this re-ranking process. They are only modeled by the facts encoder in the generation module. We adopt a neural ranking model following the previous work [26, 45]. Specifically, for each conversation context u_i and response candidate $y_i^k \in \mathcal{Y}_i$, we first build an interaction matching matrix. Given y_i^k and u_i, the model looks up a global embedding dictionary to represent y_i^k and u_i as two sequences of embedding vectors $\mathbf{E}(y_i^k) = [\mathbf{y}_{i,1}^k, \mathbf{y}_{i,2}^k, \cdots, \mathbf{y}_{i,L_y}^k]$ and $\mathbf{E}(u_i) = [\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \cdots, \mathbf{u}_{i,L_u}]$, where $\mathbf{y}_{i,j}^k \in \mathbb{R}^d$, $\mathbf{u}_{i,j} \in \mathbb{R}^d$ are the embedding vectors of the j-th word in the word sequences y_i^k and u_i respectively. The model then builds an interaction matrix $\mathbf{M}$, which computes the pairwise similarity between words in y_i^k and u_i via the dot product similarity between the embedding representations. The interaction matching matrix is used as the input of a convolutional neural network (CNN) to learn important matching features, which are aggregated by the final multi-layer perceptron (MLP) to generate a matching score.

3.5.2 ***CNN Layers and MLP.*** The interaction matrices are fed into a CNN to learn high level matching patterns as features. CNN alternates convolution and max-pooling operations over these inputs. Let $\mathbf{z}^{(l,k)}$ denote the output feature map of the l-th layer and k-th kernel, the model performs convolution operations and max-pooling operations respectively in Eq. (11) and (12).

Convolution. Let $r_w^{(l,k)} \times r_h^{(l,k)}$ denote the shape of the k-th convolution kernel in the l-th layer, the convolution operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \sigma\left(\sum_{k'=0}^{K_l-1} \sum_{s=0}^{r_w^{(l,k)}-1} \sum_{t=0}^{r_h^{(l,k)}-1} \mathbf{w}_{s,t}^{(l+1,k)} \cdot \mathbf{z}_{i+s,j+t}^{(l,k')} + b^{(l+1,k)}\right) \tag{11}$$
$$\forall l = 0, 2, 4, 6, \cdots,$$

where σ is the activation function ReLU, and $\mathbf{w}_{s,t}^{(l+1,k)}$ and $b^{(l+1,k)}$ are the parameters of the k-th kernel on the $(l+1)$-th layer to be learned. K_l is the number of kernels on the l-th layer.

Max Pooling. Let $p_w^{(l,k)} \times p_h^{(l,k)}$ denote the shape of the k-th pooling kernel in the l-th layer, the max pooling operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \max_{0 \le s < p_w^{l+1,k}} \max_{0 \le t < p_h^{l+1,k}} \mathbf{z}_{i+s,j+t}^{(l,k)} \quad \forall l = 1, 3, 5, 7, \cdots, \tag{12}$$

Finally we feed the output feature representation vectors learned by CNN into a multi-layer perceptron (MLP) to calculate the final matching score $f(u_i, y_i^k)$.

3.5.3 ***Distant Supervision for Model Training.*** For model training, we consider a pairwise ranking learning setting. The training data consists of triples $(u_i, y_i^{k+}, y_i^{k-})$, where y_i^{k+} and y_i^{k-} denote the positive and the negative response candidate for dialog context u_i. A challenging problem here is that there is no ground truth ranking labels for all the candidate responses (either the generated response or the retrieved responses) in $\mathcal{Y}_i$ given a conversation context u_i. The costs for annotating all context-response candidates pairs for model training would be very high. Thus, we generate training data to train the hybrid ranking module with distant supervision inspired by previous work on relation extraction [23]. Specifically we construct $\mathcal{Y}_i$ by mixing K retrieved response candidates $\{r_i^1, r_i^2, ..., r_i^K\}$ and one generated response candidate $\{g_i^1\}$. We then score these $K+1$ response candidates with metrics like BLEU/ ROUGE-L by comparing them with the ground truth responses in the training data of the generation module. Note that in our setting there can be two different types of ground truth: the one for the generation module to train the Seq2Seq models which we have, and the one for the generated/ retrieved response candidates to train the hybrid ranking module which does not exist in the data. Inspired by the way on deriving the supervision signals for relation extraction from Freebase by Mintz et al. [23], here we derive the

[3] http://lucene.apache.org/

[4] We set $K = 9$ in our experiments.

[5] We adopt this setting as we find that the generated top responses by Seq2Seq based models are very similar with each other.

negative and positive candidates

supervision signals for the hybrid neural ranker from the observed context/ response pairs in the training data of the generation module. Finally we treat the top k' response candidates[6] ranked by BLEU/ ROUGE-L as positive candidates and other responses as negative candidates. In this way, the training labels of response candidates can be inferred by distant supervision.[7] We perform experiments to evaluate the effectiveness of different kinds of distant supervision signals. In practice, there could be multiple appropriate and diverse responses for a given conversation context. Ideally, we need multiple reference responses for each conversation context, each for a different and relevant response. We leave generating multiple references for a conversation context for distant supervision to future work. We have to point out that it is difficult to collect the data where each context is paired with comprehensive reference responses. Our proposed method can also be easily adapted to the scenario where we have multiple reference responses for a conversation context.

Given inferred training labels, we can compute the pairwise ranking-based hinge loss, which is defined as:

$$\mathcal{L}_h = \sum_{i=1}^{I} \max(0, \epsilon - f(u_i, y_i^{k+}) + f(u_i, y_i^{k-})) + \lambda||\Theta||_2^2 \qquad (13)$$

where I is the total number of triples in the training data. $\lambda||\Theta||_2^2$ is the regularization term where λ denotes the regularization coefficient. ϵ denotes the margin in the hinge loss.

Table 3: Statistics of experimental data used in this paper.

Items	Train	Valid	Test
# Context-response pairs	1,059,370	2,067	2,066
# Facts	43,111,643	79,950	79,915
Avg # facts per context	40.70	38.68	38.68
Avg # words per facts	17.58	17.42	17.47
Avg # words per context	16.66	17.85	17.66
Avg # words per response	11.65	15.58	15.89

4 EXPERIMENTS

4.1 Data Set Description

We used the same grounded Twitter conversation data set from the study by Ghazvininejad et. al. [8]. The data contains 1 million two-turn Twitter conversations. Foursquare tips[8] are used as the fact data, which is relevant to the conversation context in the Twitter data. The Twitter conversations contain entities that tie to Foursquare. Then the conversation data is associated with the fact data by identifying Twitter conversation pairs in which the first turn contained either a handle of the entity name or a hashtag that matched a handle appears in the Foursquare tip data. The validation and test sets (around 4K conversations) are created to contain responses that are informative and useful, in order to evaluate conversation systems on their ability to produce contentful responses. The statistics of data are shown in Table 3.

4.2 Experimental Setup

4.2.1 ***Competing Methods.*** We consider different types of methods for comparison including retrieval-based, generation-based and hybrid retrieval-generation methods as follows:[9]

Seq2Seq. This is the standard Seq2Seq model with a conversation context encoder and a response decoder, which is the method proposed in [37].

Seq2Seq-Facts. This is the Seq2Seq model with an additional facts encoder, which is the generation module in the proposed hybrid neural conversational model.

KNCM-MTask-R. KNCM-MTask-R is the best setting of the knowledge-grounded neural conversation model proposed in the research by Ghazvininejad et al. [8] with multi-task learning. This system is trained with 23 million general Twitter conversation data to learn the conversation structure or backbone and 1 million grounded conversation data with associated facts from Foursquare tips. Since we used the same 1 million grounded Twitter conversation data set from this work, our experimental results are directly comparable with response generation results reported by Ghazvininejad et al. [8].

Retrieval. This method uses BM25 model [32] to match the conversation context with conversation context/ response pairs in the historical conversation repository to find the best pair, which is the retrieval module in the proposed model.

HybridNCM. This is the method proposed in this paper. It contains two different variations: 1) **HybridNCM-RS** is a hybrid method by mixing generated response candidates from Seq2Seq and retrieved response candidates from the retrieval module in HybridNCM; 2) **HybridNCM-RSF** is a hybrid method by mixing generated response candidates from Seq2Seq-Facts and retrieved response candidates from the retrieval module in HybridNCM.

4.2.2 ***Evaluation Methodology.*** Following previous related work [8, 18, 35], we use BLEU and ROUGE-L for the automatic evaluation of the generated responses. The corpus-level BLEU is known to better correlate with human judgments including conversation response generation [6] comparing with sentence-level BLEU. We also report lexical diversity as an automatic measure of informativeness and diversity. The lexical diversity metrics include Distinct-1 and Distinct-2, which are respectively the number of distinct unigrams and bigrams divided by the total number of generated words in the responses. In additional to automatic evaluation, we also perform human evaluation (Section 4.3.2) of the generated responses of different systems on the *appropriateness* and *informativeness* following previous work [8].

4.2.3 ***Parameter Settings.*** All models are implemented with PyTorch[10] and MatchZoo[11] toolkit. Hyper-parameters are tuned with the validation data. The hyper-parameters in the generation-based baselines and the generation module in the proposed hybrid neural conversation model are shown in Table 4. For the hyper-parameter settings in the hybrid ranking module, we set the window size of the convolution and pooling kernels as (6, 6). The number of convolution kernels is 64. The dropout rate is set to 0.5. The

[6] We set $k' = 3$ in our experiments.

[7] Note that we do not have to do such inference during model testing, since we just need to use the trained ranking model to score response candidates instead of computing training loss during model testing.

[8] https://foursquare.com/

[9] We did not compare with [34] since the code of both the state-of-the-practice IR system [42] and the multi-seq2seq model, which are the two main components of the proposed ensemble model in [34], is not available.

[10] https://pytorch.org/

[11] https://github.com/NTMC-Community/MatchZoo

Table 4: The hyper-parameter settings in the generation-based baselines and the generation module in the proposed hybrid neural conversation model. These settings are the optimized settings tuned with the validation data.

Models	Seq2Seq	Seq2Seq-Facts
Embedding size	512	256
# LSTM layers in encoder/decoder	2	2
LSTM hidden state size	512	256
Learning rate	0.0001	0.001
Learning rate decay	0.5	0.5
# Steps between validation	10000	5000
Patience of early stopping	10	10
Dropout	0.3	0.3

margin in the pairwise-ranking hinge loss is 1.0. The distant supervision signals and the number of positive samples per context in the hybrid ranking module are tuned with validation data. The used distant supervision signal is BLEU-1 and we treat top 3 response candidates ranked by BLEU-1 as positive samples. All models are trained on a single Nvidia Titan X GPU by stochastic gradient descent with Adam [16] algorithm. The initial learning rate is 0.0001. The parameters of Adam, β_1 and β_2, are 0.9 and 0.999 respectively. The batch size is 500. The maximum conversation context/ response length is 30. Word embeddings in the neural ranking model will be initialized by the pre-trained GloVe[12] word vectors and updated during the training process.

4.3 Evaluation Results

4.3.1 ***Automatic Evaluation.*** We present evaluation results over different methods on Twitter/ Foursquare data in Table 5. We summarize our observations as follows: (1) If we compare retrieval-based methods and HybridNCM with pure generation based methods such as Seq2Seq, Seq2Seq-Facts and KNCM-MTask-R, we find that retrieval-based methods and HybridNCM with a retrieval module achieve better performance in terms of BLEU and ROUGE-L. This verifies the competitive performance of retrieval-based methods for conversation response generation reported in previous related works [34]. (2) Both HybridNCM-RS and HybridHCM-RSF outperform all the baselines including KNCM-MTask-R with multi-task learning proposed recently by Ghazvininejad et al. [8] under BLEU and ROUGE-L. The results demonstrate that combining both retrieved and generated response candidates does help produce better responses in conversation systems. For the two variations of HybridNCM, HybridNCM-RSF achieves better BLEU and worse ROUGE-L. Overall the performances of these two variations of HybridNCM are similar to each other. One possible reason is that, the main gain over baselines comes from the retrieval module and the re-ranking process in hybrid ranking module. So the differences in the generation module do not change the results too much. (3) For lexical diversity metrics like 1-gram/ 2-gram diversity, generation-based methods are far behind retrieval-based methods and HybridNCM, even for KNCM-MTask-R with external grounded knowledge and multi-task learning. This result shows that the retrieved response candidates are more diverse than the response candidates generated by Seq2Seq models. Researchers have studied Maximum Mutual Information (MMI) object functions [17] in neural models in order to generate more diverse responses. It would be interesting to compare MMI models with IR models for conversation response generation. We leave this study to future work.

Table 5: Comparison of different models over the Twitter/ Foursquare data. Numbers in bold font mean the result is the best under the metric corresponding to the column. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. Note that we can only do significance test for ROUGE-L since the other metrics are corpus-level metrics. The results of KNCM-MTask-R are directly cited from Ghazvininejad et al. [8] since we used the same 1 million grounded Twitter conversation data set from this work. Thus we don't have the ROUGE-L result for this baseline method.

Method	BLEU	ROUGE-L	Distinct-1	Distinct-2
Seq2Seq	0.5032	8.4432	2.36%	11.18%
Seq2Seq-Facts	0.5904	8.8291	1.91%	7.85%
KNCM-MTask-R	1.0800	\	7.08%	21.90%
Retrieval	1.2491	8.6302	**14.68%**	**58.71%**
HybridNCM-RS	1.3450	**10.4078**‡	11.30%	47.35%
HybridNCM-RSF	**1.3695**	10.3445‡	11.10%	46.01%

4.3.2 ***Human Evaluation.*** Automatic evaluation of response generation is still a challenging problem. To complement the automatic evaluation results, we also perform human evaluation to compare the performance of different methods following previous related works [8, 33, 34]. We ask three educated annotators to do the human evaluation. We randomly sample 400 conversation contexts from the test data, and instruct the annotators to rate the output responses of different systems.[13] We hide the system ids and randomly permute the output responses to rule out human bias. In the annotation guidelines, we ask the annotators to evaluate the quality of output responses by different systems from the following 2 dimensions:

- *Appropriateness*: evaluate whether the output response is appropriate and relevant to the given conversation context.
- *Informativeness*: evaluate whether the output response can provide useful and factual information for the users.

Three different labels "0" (bad), "+1" (neutral), "+2" (good) are used to evaluate the quality of system output responses. Table 6 shows the comparison of different models with human evaluation. The table contains the mean score, ratio of three different categories of labels and the agreement scores among three annotators. The agreement score is evaluated by Fleiss' kappa [5] which is a statistical measure of inter-rater consistency. Most agreement scores are in the range from 0.2 to 0.5, which can be interpreted as "fair agreement" or "moderate agreement". [14] The annotators have relative higher agreement scores for the informativeness of

[12] https://nlp.stanford.edu/projects/glove/

[13] We mainly performed human evaluation on our methods and three baselines Seq2Seq, Seq2Seq-Facts and Retrieval. We didn't include KNCM-MTask-R into human evaluation since there is no open source code or official implementation from [8]. The results of KNCM-MTask-R in Table 5 are cited numbers from [8] since we used the same experimental data sets.

[14] https://en.wikipedia.org/wiki/Fleiss%27_kappa

Table 6: Comparison of different models with human evaluation. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. The agreement score is evaluated by Fleiss' kappa [5] which is a statistical measure of inter-rater consistency. Agreement scores are comparable to previous results (0.2-0.5) as reported in [33, 34]. Higher scores indicate higher agreement degree. The results of KNCM-MTask-R are not included in this table since the generated responses by KNCM-MTask-R are not available and the code of the KNCM-MTask-R is also not available.

Comparision	Appropriateness					Informativeness				
Method	Mean	Bad(0)	Neutral(+1)	Good(+2)	Agreement	Mean	Bad(0)	Neutral(+1)	Good(+2)	Agreement
Seq2Seq	0.4733	61.67%	29.33%	9.00%	0.2852	0.2417	77.58%	20.67%	1.75%	0.4731
Seq2Seq-Facts	0.4758	62.50%	27.42%	10.08%	0.3057	0.3142	70.75%	27.08%	2.17%	0.4946
Retrieval	0.9425	34.42%	36.92%	28.67%	0.2664	0.8008	35.50%	48.92%	15.58%	0.3196
HybridNCM-RS	**1.1175**‡	27.83%	32.58%	39.58%	0.3010	**1.0650**‡	18.42%	56.67%	24.92%	0.1911
HybridNCM-RSF	1.0358‡	31.67%	33.08%	35.25%	0.2909	1.0292‡	20.42%	56.25%	23.33%	0.2248

Table 7: Side-by-side human evaluation results. Win/Tie/Loss are the percentages of conversation contexts a method improves, does not change, or hurts, compared with the method after "v.s." on human evaluation scores. HNCM denotes HybridNCM. Seq2Seq-F denotes Seq2Seq-Facts.

Type	Appropriateness	Informativeness
Comparision	Win/Tie/Loss	Win/Tie/Loss
HNCM-RS v.s. Seq2Seq	0.71/0.15/0.14	0.84/0.10/0.06
HNCM-RSF v.s. Seq2Seq	0.68/0.16/0.16	0.82/0.11/0.07
HNCM-RS v.s. Seq2Seq-F	0.70/0.15/0.15	0.80/0.12/0.08
HNCM-RSF v.s. Seq2Seq-F	0.65/0.19/0.17	0.77/0.15/0.09
HNCM-RS v.s. Retrieval	0.43/0.31/0.26	0.50/0.31/0.18
HNCM-RSF v.s. Retrieval	0.41/0.30/0.29	0.50/0.28/0.22

Table 8: The number and percentage of top responses selected by the hybrid ranking module from retrieved/ generated response candidates. #PickedGenRes is the number of selected responses from generated response candidates. #PickedRetRes is the number of selected responses from retrieved response candidates. #PickedTop1BM25 is the number of selected responses which is also ranked as top 1 responses by BM25.

Item	HybridNCM-RS		HybridNCM-RSF	
#TestQNum	2066	100.00%	2066	100.00%
#PickedGenRes	179	8.66%	275	13.31%
#PickedRetRes	1887	91.34%	1791	86.69%
#PickedTop1BM25	279	13.50%	253	12.25%

generation-based methods like Seq2Seq and Seq2Seq-Facts, since these methods are likely to generate short responses or even responses containing fluency and grammatical problems.

We summarize our observations on the human evaluation results in Table 6 as follows: (1) For the mean scores, we can see both HybridNCM-RS and HybridNCM-RSF achieve higher average rating scores compared with all baselines, in terms of both appropriateness and informativeness. Human evaluation results verify that hybrid models indeed help improve the response generation performances of conversation systems. For baselines, the retrieval-based baseline is stronger than generation-based baselines. For HybridNCM-RS and HybridNCM-RSF, HybridNCM-RS achieves relatively higher average human rating scores with a small gap. (2) For the ratios of different categories of labels, we can see that more than 72% of output responses by HybridNCM-RS (68% for HybridNCM-RSF) are labeled as "good (+2)" or "neutral (+1)" for appropriateness, which means that most output responses of hybrid models are semantically relevant to the conversation contexts. Generation-based methods like Seq2Seq and Seq2Seq-Facts perform worse than both the retrieval-based method and hybrid models. The retrieval-based method, although quite simple, achieves much higher ratios for the categories "good (+2)" and "neutral (+1)" compared with generation-based methods. For informativeness, the hybrid models HybridNCM-RS and HybridNCM-RSF are still the best, beating both generation-based baselines and retrieval-based baselines. These results show that the re-ranking process in the hybrid ranking module trained with distant supervision in hybrid conversation models can further increase the informativeness of results by promoting response candidates with more factual content. (3) For the statistical significance test, both HybridNCM-RS and HybridNCM-RSF outperform all baseline methods with $p < 0.05$ measured by the Student's paired t-test in terms of human evaluation scores. We also show the side-by-side human evaluation results in Table 7. The results clearly confirm that performances of hybrid models are better than or comparable to the performances of all baselines for most test conversation contexts.

4.4 Analysis of Top Responses Selected by Re-ranker

The number and percentage of top responses selected from retrieved/ generated response candidates by the neural ranking model are shown in Table 8. We summarize our observation as follows: (1) most picked results (91.34% for HybridNCM-RS and 86.69% for HybridNCM-RSF) are from the retrieved response candidates. This is reasonable because we have multiple retrieved response candidates but only one generated response candidate. In some cases, generated responses are preferred to retrieved responses. (2) Although the percentage of generated responses is not high, this does not mean we can just directly use the results returned by the retrieval method. If we look at the row "PickedTop1BM25", we can find that only very few responses ranked as the 1st by BM25 are ranked as the 1st again by HybridNCM. Thus, HybridNCM changed the order of these responses candidates significantly. In particular, the hybrid ranking module in HybridNCM did the following two

Table 9: The response generation performance when we vary the ratios of positive samples in distant supervision.

	Supervision	BLEU-1		BLEU-2		ROUGE-L	
Model	# Positive	BLEU	ROUGE-L	BLEU	ROUGE-L	BLEU	ROUGE-L
HybridNCM-RS	k'=1	0.9022	8.9596	0.7547	8.8351	1.0964	8.9234
	k'=2	1.0649	9.7241	1.1099	9.9168	1.1019	9.6216
	k'=3	**1.3450**	**10.4078**	**1.1165**	**10.1584**	**1.1435**	**10.0928**
HybridNCM-RSF	k'=1	1.0223	9.2996	**1.1027**	9.2453	1.0035	9.2812
	k'=2	1.3284	9.8637	1.0175	9.8562	**1.0999**	**9.8061**
	k'=3	**1.3695**	**10.3445**	0.8239	**9.8575**	0.9838	9.7961

Table 10: The response generation performance when we vary different distant supervision signals. This table shows the results for the setting "k'=3", where there are 3 positive response candidates for each conversation context. "SentBLEU" denotes using sentence-level BLEU scores as distant supervision signals.

Model	HybridNCM-RS		HybridNCM-RSF	
Supervision	BLEU	ROUGE-L	BLEU	ROUGE-L
BLEU-1	**1.3450**	**10.4078**	**1.3695**	**10.3445**
BLEU-2	1.1165	10.1584	0.8239	9.8575
ROUGE-L	1.1435	10.0928	0.9838	9.7961
SentBLEU	0.8326	9.2887	1.0631	9.6338

tasks: a) re-evaluate and re-rank the previous generated/ retrieved responses to promote the good response; b) try to inject some generated responses by Seq2Seq models into retrieved results if possible. (3) We notice that response candidates generated by Seq2Seq-Facts model are more likely to be picked compared to those generated by Seq2Seq. When a generated response contains rich factual content, the hybrid ranking module is more likely to pick it, which also helps boost the BLEU metrics.

4.5 Impact of Distant Supervision Signals

We investigate the impact of different distant supervision signals on the response generation performance in Table 10. We find that distant supervision signals like BLEU-1 are quite effective for training the hybrid ranking module. The sentence-level BLEU is not a good choice for the distant supervision signal. The reason is that the sentence-level BLEU is computed only based on the n-gram precision statistics for a given sentence pair. This score has a larger variance compared with the corpus-level BLEU. Since sentence-level BLEU scores would become very small smoothed values if there are no 4-gram or trigram matches between two sentences, which may happen frequently in short text pairs.

4.6 Impact of Ratios of Positive Samples

We further analyze the impact of the ratios of positive/ negative training samples on the response generation performance. Table 9 shows the results. The value of k' is the number of positive response candidates for each conversation context when we train the hybrid ranking module. When $k' = 1$, we select one positive candidate from the ground truth responses in the training data, which is equivalent to the negative sampling technique. As k' increases, we construct the positive candidates by selecting one positive sample from the ground truth responses and $k' - 1$ positive samples from the top ranked candidates by distant supervision. We find that larger k' can improve the response generation performance. This is reasonable since larger k' means the model can observe more positive training samples and positive/ negative response pairs in the pairwise ranking loss minimization process. However, increasing the value of k' also adds risks of introducing noisy positive training data. Thus, the value of k' is a hyper-parameter, and needs to be tweaked via trial and error.

Table 11: Examples of output responses by different methods. r means the response is retrieved. g means the response is generated. Entities marked with [ENTITY] have been anonymized to avoid potentially negative publicity. "HNCM" denotes "HybridNCM".

Context	Donated to the [ENTITY] last night and now I have to listen to automated phone calls. It's enough to make me want to cancel.	
Method	r/g	System Output Response
Ground Truth	-	Ask them to put you on their internal dnc list. They will likely respect this, because future calls can get them charged.
Seq2Seq	g	I didn't get it. I didn't.
Seq2Seq-Facts	g	I'm sorry to hear that. Please dm us your email address so we can look into this. Thanks!
Retrieval	r	It's a known issue in LA county. I just got an email from my dm and tech and it should be good in 30 minutes or so.
HNCM-RS	r	We're listening and would like to know more and help with your experience. Please follow us so i can dm you our contact info. [ENTITY]
HNCM-RSF	g	We're sorry to hear this. Please dm us if you need assistance. Please dm us your contact info so we can look into this.

4.7 Examples and Case Study

We perform a case study in Table 11 on the outputs by different methods. In this example, we can find that the response produced by Seq2Seq is very general and it does not provide any useful information for the user. Seq2Seq-Facts generates a much better response by injecting more factual content into response generation process. The response returned by the Retrieval method is also relevant to the context. However, it provides very specific information like "LA county", "30 minutes", which may have negative impact on the appropriateness of this response for some users. The responses produced by hybrid models achieve a good balance between specificity and generalization. The response by HybridNCM-RS is from retrieved results and the response by HybridNCM-RSF is from generated results, which shows that both retrieval-based methods and generation-based methods have the capacity to produce good responses for certain contexts. Thus it is a natural to combine these two different types of methods for response generation.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we perform a comparative study of retrieval-based and generation-based methods for building conversation systems. We propose a hybrid neural conversation model with the capability of both response retrieval and generation in order to combine the merits of these two types of methods. For the training of the hybrid ranking module, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. Experimental results with Twitter/ Foursquare data show that the proposed model outperforms both retrieval-based and generation-based methods including a recently proposed knowledge-grounded neural conversation model under both automatic evaluation and human evaluation. The findings in this study provide insights on how to integrate text retrieval and text generation models for building conversation systems. For future work, we would like to study reinforcement learning methods for response selection in order to directly optimize metrics like BLEU/ ROUGE.

6 ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor. The authors would like to thank Kaixi Zhang for the contribution on data annotation/ proofreading on this work.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014).

[2] A. Bordes, Y. Boureau, and J. Weston. 2017. Learning end-to-end goal-oriented Dialog. *ICLR '17*.

[3] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* (2014).

[4] B. Dhingra, L. Li, X. Li, J. Gao, Y. Chen, F. Ahmed, and L. Deng. 2017. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. In *ACL'17*.

[5] J.L. Fleiss et al. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin* 76, 5 (1971), 378–382.

[6] M. Galley, C. Brockett, A. Sordoni, Y. Ji, M. Auli, C. Quirk, M. Mitchell, J. Gao, and B. Dolan. 2015. deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets. *CoRR* abs/1506.06863 (2015).

[7] J. Gao, M. Galley, and L. Li. 2018. Neural Approaches to Conversational AI. *CoRR* abs/1809.08267 (2018).

[8] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih, and M. Galley. 2018. A Knowledge-Grounded Neural Conversation Model. In *AAAI '18*.

[9] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM '16*.

[10] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng. 2019. A Deep Look into Neural Ranking Models for Information Retrieval. *CoRR* (2019).

[11] M. Henderson. 2015. Machine Learning for Dialog State Tracking : a Review.

[12] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997).

[13] B. Hu, Z. Lu, H. Li, and Q. Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NIPS '14*.

[14] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. P. Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM '13*.

[15] Z. Ji, Z. Lu, and H. Li. 2014. An Information Retrieval Approach to Short Text Conversation. *CoRR* abs/1408.6988 (2014).

[16] D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* (2014).

[17] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 2015. A Diversity-Promoting Objective Function for Neural Conversation Models. *CoRR* abs/1510.03055 (2015).

[18] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and W. B. Dolan. 2016. A Persona-Based Neural Conversation Model. In *ACL'16*.

[19] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *EMNLP'16*.

[20] X. Liu, Y. Shen, K. Duh, and J. Gao. 2018. Stochastic Answer Networks for Machine Reading Comprehension. In *ACL '18*.

[21] R. Lowe, N. Pow, I. Serban, and J. Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *CoRR* abs/1506.08909 (2015).

[22] T. Luong, H. Pham, and C. D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP '15*.

[23] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant Supervision for Relation Extraction without Labeled Data. In *ACL '09*.

[24] B. Mitra, F. Diaz, and N. Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW '17*.

[25] G. Pandey, D. Contractor, V. Kumar, and S. Joshi. 2018. Exemplar Encoder-Decoder for Neural Conversation Generation. In *ACL '18*.

[26] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. 2016. Text Matching as Image Recognition. In *AAAI '16*.

[27] Jay M. Ponte and W. B. Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR '98*.

[28] L. Qin, M. Galley, C. Brockett, X. Liu, X. Gao, B. Dolan, Y. Choi, and J. Gao. 2019. Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading. In *ACL '19*.

[29] M. Qiu, F. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, and W. Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *ACL '17*.

[30] M. Qiu, L. Yang, F. Ji, W. Zhou, J. Huang, H. Chen, W. B. Croft, and W. Lin. 2018. Transfer Learning for Context-Aware Question Matching in Information-seeking Conversations in E-commerce. In *ACL '18*.

[31] A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *ACL '11*.

[32] S. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR '94*.

[33] L. Shang, Z. Lu, and H. Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL '15*.

[34] Y. Song, C. Li, J. Nie, M. Zhang, D. Zhao, and R. Yan. 2018. An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems. In *IJCAI '18*.

[35] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *NAACL '15*.

[36] Z. Tian, R. Yan, L. Mou, Y. Song, Y. Feng, and D. Zhao. 2017. How to Make Context More Useful? An Empirical Study on Context-Aware Neural Conversational Models. In *ACL '17*.

[37] O. Vinyals and Q. V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015).

[38] Y. Wu, F. Wei, S. Huang, Z. Li, and M. Zhou. 2018. Response Generation by Context-aware Prototype Editing. *CoRR* (2018).

[39] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL '17*.

[40] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR '17*.

[41] R. Yan, Y. Song, and H. Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*.

[42] R. Yan, Y. Song, X. Zhou, and H. Wu. 2016. "Shall I Be Your Chat Companion?": Towards an Online Human-Computer Conversation System. In *CIKM '16*.

[43] R. Yan, D. Zhao, and W. E. 2017. Joint Learning of Response Ranking and Next Utterance Suggestion in Human-Computer Conversation System. In *SIGIR '17*.

[44] L. Yang, Q. Ai, J. Guo, and W. B. Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM '16*.

[45] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *SIGIR '18*.

[46] L. Yang, H. Zamani, Y. Zhang, J. Guo, and W. B. Croft. 2017. Neural Matching Models for Question Retrieval and Next Question Prediction in Conversation. *CoRR* (2017).

[47] J. Yu, M. Qiu, J. Jiang, J. Huang, S. Song, W. Chu, and H. Chen. 2018. Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce. *WSDM '18*.

[48] R. Zhang, J. Guo, Y. Fan, Y. Lan, J. Xu, and X. Cheng. 2018. Learning to Control the Specificity in Neural Response Generation. In *ACL '18*.

[49] Y. Zhang, M. Galley, J. Gao, Z. Gan, X. Li, C. Brockett, and B. Dolan. 2018. Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization. *CoRR* (2018).

[50] L. Zhou, J. Gao, D. Li, and H. Shum. 2018. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *CoRR* (2018).

[51] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, and R. Yan. 2016. Multi-view Response Selection for Human-Computer Conversation. In *EMNLP*.