RefNet: A Reference-Aware Network for Background Based Conversation

Meng, C.; Ren, P.; Chen, Z.; Monz, C.; Ma, J.; de Rijke, M.

DOI
10.1609/aaai.v34i05.6370

Publication date
2020

Document Version
Author accepted manuscript

Published in
AAAI-20, IAAI-20, EAAI-20 proceedings

RefNet: A Reference-aware Network for Background Based Conversation

Chuan Meng,[1] Pengjie Ren,[2*] Zhumin Chen,[1*] Christof Monz,[2] Jun Ma,[1] Maarten de Rijke[2]
[1]Shandong University, Qingdao, China,
[2]University of Amsterdam, Amsterdam, The Netherlands
mengchuan@mail.sdu.edu.cn
{p.ren, c.monz, derijke}@uva.nl, {chenzhumin, majun}@sdu.edu.cn

Abstract

Existing conversational systems tend to generate generic responses. Recently, Background Based Conversations (BBCs) have been introduced to address this issue. Here, the generated responses are grounded in some background information. The proposed methods for BBCs are able to generate more informative responses, however, they either cannot generate natural responses or have difficulties in locating the right background information. In this paper, we propose a Reference-aware Network (RefNet) to address both issues. Unlike existing methods that generate responses token by token, RefNet incorporates a novel *reference decoder* that provides an alternative way to learn to directly select a *semantic unit* (e.g., a span containing complete semantic information) from the background. Experimental results show that RefNet significantly outperforms state-of-the-art methods in terms of both automatic and human evaluations, indicating that RefNet can generate more appropriate and human-like responses.

1 Introduction

Dialogue systems have attracted a lot of attention recently (Huang, Zhu, and Gao 2019). Sequence-to-sequence models (Sutskever, Vinyals, and Le 2014; Lei et al. 2018) are an effective framework that is commonly adopted in existing studies. However, a problem of sequence-to-sequence based methods is that they tend to generate generic and non-informative responses which provide deficient information (Gao et al. 2019).

Previous research has proposed various methods to alleviate the issue, such as adjusting objective functions (Li et al. 2016; Jiang et al. 2019), incorporating external knowledge (Ghazvininejad et al. 2018; Parthasarathi and Pineau 2018; Dinan et al. 2019), etc. Recently, Background Based Conversations (BBCs) have been proposed for generating more informative responses that are grounded in some background information (Zhou, Prabhumoye, and Black 2018; Moghe et al. 2018). As shown in Fig. 1, unlike previous conversational settings (Serban et al. 2016), in a BBC background material (e.g., a plot or review about a movie) is supplied to promote topic-specific conversations.

*Corresponding author

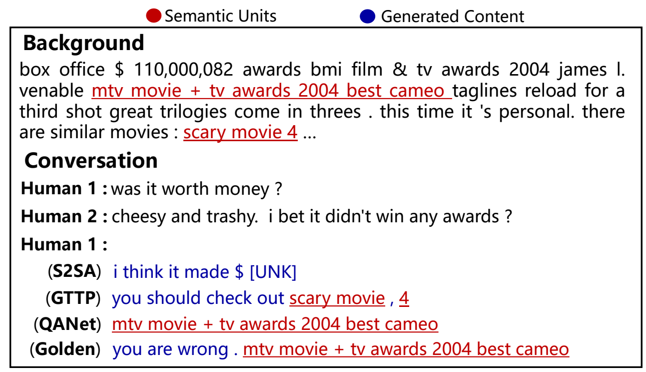

Figure 1: Background Based Conversation (BBC).

Existing methods for BBCs can be grouped into two categories, *generation-based* methods (e.g., GTTP (See, Liu, and Manning 2017)) and *extraction-based* methods (e.g., QANet (Yu et al. 2018)). Generation-based methods generate the response token by token, so they can generate natural and fluent responses, generally. However, generation-based methods suffer from two issues. First, they are relatively ineffective in leveraging background information. For example, for the case in Fig. 1, S2SA does not leverage background information at all. Second, they have difficulties locating the right semantic units in the background information. Here, a *semantic unit* is a span from the background information that expresses complete semantic meaning. For example, in Fig. 1, the background contains many semantic units, e.g., "*mtv movie + tvawards 2004 best cameo*" and "*scary movie 4*." GTTP uses the wrong semantic unit "*scary movie 4*" to answer the question by "human 2." Moreover, because generation-based methods generate the response one token at a time, they risk breaking a complete semantic unit, e.g., "*scary movie 4*" is split by a comma in the response of GTTP in Fig. 1. The reason is that generation-based methods lack a global perspective, i.e., each decoding step only focuses on a single (current) token and does not consider the tokens to be generated in the following steps. Extraction-based methods extract a span from the background as their response and are relatively good at locating the right semantic unit. But because of their extractive nature, they cannot generate natural conversational responses, see, e.g., the response of QANet in Fig. 1.

We propose a **Ref**erence-aware **Net**work (RefNet) to address above issues. RefNet consists of four modules: a *background encoder*, a *context encoder*, a *decoding switcher*, and a *hybrid decoder*. The background encoder and context encoder encode the background and conversational context into representations, respectively. Then, at each decoding step, the decoding switcher decides between *reference decoding* and *generation decoding*. Based on the decision made by the decoding switcher, the hybrid decoder either selects a semantic unit from the background (*reference decoding*) or generates a token otherwise (*generation decoding*). In the latter case, the decoding switcher further determines whether the hybrid decoder should predict a token from the vocabulary or copy one from the background. Besides generating the response token by token, RefNet also provides an alternative way to learn to select a semantic unit from the background directly. Experiments on a BBC dataset show that RefNet significantly outperforms state-of-the-art methods in terms of both automatic and, especially, human evaluations.

Our contributions are as follows:

- We propose a novel architecture, RefNet, for BBCs by combing the advantages of extraction-based and generation-based methods. RefNet can generate more informative and appropriate responses while retaining fluency.
- We devise a decoding switcher and a hybrid decoder to adaptively coordinate between *reference decoding* and *generation decoding*.
- Experiments show that RefNet outperforms state-of-the-art models by a large margin in terms of both automatic and human evaluations.

2 Related work

We survey two types of related work on BBCs: generation-based and extraction-based methods.

2.1 Generation-based methods

Most effective generation-based models are based on sequence-to-sequence modeling (Sutskever, Vinyals, and Le 2014) and an attention mechanism (Bahdanau, Cho, and Bengio 2015). The proposed methods have achieved promising results on different conversational tasks (Serban et al. 2016). However, response informativeness is still a urgently need to be addressed challenge; these approaches prefer generating generic responses such as "I don't know" and "thank you", which make conversations dull (Gao et al. 2019). Various methods have been proposed to improve response informativeness, such as adjusting objective functions (Li et al. 2016; Jiang et al. 2019), incorporating latent topic information (Xing et al. 2017), leveraging outside knowledge bases (Liu et al. 2018; Zhou et al. 2018) and knowledge representation (Ghazvininejad et al. 2018; Parthasarathi and Pineau 2018; Lian et al. 2019), etc.

Recently, Background Based Conversations (BBCs) have been proposed for generating more informative responses by exploring related background information (Zhou, Prabhumoye, and Black 2018; Dinan et al. 2019). Moghe et al. (2018) build a dataset for BBC and conduct experiments with state-of-the-art generation-based methods. They show that generation-based methods can generate fluent, natural responses, but have difficulty in locating the right background information. Therefore, most recent studies try to address this issue (Li et al. 2019; Qin et al. 2019). Zhang, Ren, and de Rijke (2019) introduce a pre-selection process that uses dynamic bi-directional attention to improve background information selection. Liu et al. (2019) propose an augmented knowledge graph based chatting model via transforming background information into knowledge graph. However, generation-based models still cannot solve inherent problems effectively, such as tending to break a complete semantic unit and generate shorter responses.

2.2 Extraction-based methods

Extraction-based methods have originally been proposed for Reading Comprehension (RC) tasks (Rajpurkar et al. 2016), where each question can be answered by a right span in a given passage. Wang and Jiang (2017) combine match-LSTM and a pointer network (Vinyals, Fortunato, and Jaitly 2015) to predict the boundary of the answer. Seo et al. (2016) propose BiDAF, which uses a variant co-attention architecture (Xiong, Zhong, and Socher 2017) to enhance the extraction result. Wang et al. (2017) propose R-Net, which introduces a self-matching mechanism. Yu et al. (2018) propose QANet, which devises an encoder consisting exclusively of convolution and self-attention. For BBCs, Moghe et al. (2018) show that extraction-based methods are better at locating the right background information than generation-based methods. However, current extraction-based methods are specifically designed for RC tasks. They are not suitable for BBCs for two reasons: First, BBCs usually do not have standard factoid questions like those in RC tasks. Second, BBCs require that the responses are fluent and conversational, which cannot be met by rigid extraction; see Fig. 1.

Unlike the work summarized above, we propose RefNet to combine the advantages of generation-based and extraction-based methods while avoiding their shortcomings. The main challenge that RefNet addresses is how to design an effective neural architecture that is able to refer to the right background information at the right time in the right place of a conversation while minimizing the influence on response fluency.

3 Reference-aware Network

Given a background in the form of free text $K = (k_1, k_2, \ldots, k_t, \ldots, k_{L_K})$ with L_K tokens and a current conversational context $C_\tau = (\ldots, X_{\tau-3}, X_{\tau-2}, X_{\tau-1})$, the task of BBC is to generate a response X_τ at τ. Each X_τ contains a sequence of L_{X_τ} units, i.e., $X_\tau = (x_1^\tau, x_2^\tau, \ldots, x_t^\tau, \ldots, x_{L_{X_\tau}}^\tau)$, where x_t^τ, the unit at timestamp t, could be a token $\{x_{t,i}^\tau\}_{i=1}^1$ or a semantic unit $\{x_{t,i}^\tau\}_{i=1}^n$ containing n tokens.

RefNet consists of four modules: background encoder, context encoder, decoding switcher, and hybrid decoder; see

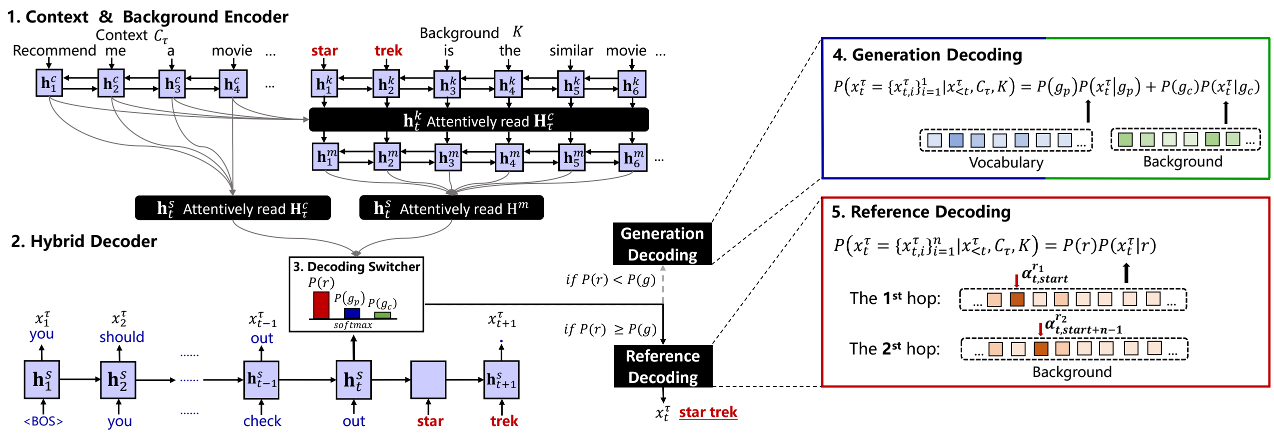

Figure 2: Overview of RefNet.

Fig. 2. Background and context encoders encode the given background K and context C_τ into latent representations $\mathbf{H}^k$ and $\mathbf{H}_\tau^c$, respectively. $\mathbf{H}^k$ and $\mathbf{H}_\tau^c$ go through a matching layer to get a context-aware background representation $\mathbf{H}^m$. At each decoding step, the decoding switcher predicts the probabilities of executing the *reference decoding* or *generation decoding*. The hybrid decoder takes $\mathbf{H}_\tau^c$, $\mathbf{H}^m$ and the embedding of the previous token as input and computes the probability of selecting a semantic unit from the background (*reference decoding*) or generating a token (*generation decoding*) based on the decision made by the decoding switcher. Next, we introduce the separate modules.

3.1 Background and context encoders

We use a bi-directional RNN (Schuster and Paliwal 1997) with GRU (Cho et al. 2014) to convert the context and background sequences into two hidden state sequences $\mathbf{H}_\tau^c = (\mathbf{h}_1^c, \mathbf{h}_2^c, \ldots, \mathbf{h}_{L_{C_\tau}}^c)$ and $\mathbf{H}^k = (\mathbf{h}_1^k, \mathbf{h}_2^k, \ldots, \mathbf{h}_{L_K}^k)$:

$$
\begin{aligned}
\mathbf{h}_t^c &= \mathbf{BiGRU}_c(\mathbf{h}_{t-1}^c, \mathbf{e}(x_t)),\\
\mathbf{h}_t^k &= \mathbf{BiGRU}_k(\mathbf{h}_{t-1}^k, \mathbf{e}(k_t)),
\end{aligned} \tag{1}
$$

where $\mathbf{h}_t^c$ or $\mathbf{h}_t^k$ correspond to a token in the context or background, respectively, and $\mathbf{e}(x_t)$ and $\mathbf{e}(k_t)$ are the embedding vectors, respectively. We concatenate the responses in the context, L_{C_τ} is the number of all tokens in the context, and we do not consider the segmentation of semantic units during encoding, i.e., each x_t^τ is a token $\{x_{t,i}^\tau\}_{i=1}^1$.

Further, we use a matching layer (Wang and Jiang 2017; Wang et al. 2017) to get the context-aware background representation $\mathbf{H}^m = (\mathbf{h}_1^m, \mathbf{h}_2^m, \ldots, \mathbf{h}_{L_K}^m)$:

$$
\mathbf{h}_t^m = \mathbf{BiGRU}_m(\mathbf{h}_{t-1}^m, [\mathbf{h}_t^k; \mathbf{c}_t^{kc}]), \tag{2}
$$

where $\mathbf{c}_t^{kc}$ is calculated using an attention mechanism (Bahdanau, Cho, and Bengio 2015) with $\mathbf{h}_t^k$ attentively reading $\mathbf{H}_\tau^c$:

$$
\begin{aligned}
s_{t,j}^{kc} &= \mathbf{v}_{kc}^{\mathrm{T}} \tanh(\mathbf{W}_{kc}\mathbf{h}_j^c + \mathbf{U}_{kc}\mathbf{h}_t^k + \mathbf{b}_{kc}),\\
\alpha_{t,i}^{kc} &= \frac{\exp(s_{t,i}^{kc})}{\sum_{j=1}^{L_{C_\tau}} \exp(s_{t,j}^{kc})},\ \mathbf{c}_t^{kc} = \sum_{i=1}^{L_{C_\tau}} \alpha_{t,i}^{kc}\mathbf{h}_i^c,
\end{aligned} \tag{3}
$$

where $\mathbf{W}_{kc}$, $\mathbf{U}_{kc}$, $\mathbf{v}_{kc}$ and $\mathbf{b}_{kc}$ are parameters.

3.2 Hybrid decoder

During training, we know that the next x_t^τ to be generated is a token $\{x_{t,i}^\tau\}_{i=1}^1$ or a semantic unit $\{x_{t,i}^\tau\}_{i=1}^n$. If $x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^n$, then x_t^τ is generated in *reference decoding* mode with the probability modeled as follows:

$$
P(x_t^\tau | x_{<t}^\tau, C_\tau, K) = P(r)P(x_t^\tau \mid r), \tag{4}
$$

where $P(r)$ is the *reference decoding* probability (see §3.3); $P(x_t^\tau \mid r)$ is the probability of generating x_t^τ under the *reference decoding* r. If $x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^1$, then x_t^τ is generated in *generation decoding* mode with the probability modeled as:

$$
\begin{aligned}
&P(x_t^\tau | x_{<t}^\tau, C_\tau, K) =\\
&\quad P(g_p)P(x_t^\tau \mid g_p) + P(g_c)P(x_t^\tau \mid g_c),
\end{aligned} \tag{5}
$$

where $P(g) = P(g_p) + P(g_c)$ is the *generation decoding* probability; $P(g_p)$ is the *predicting generation decoding* probability (see §3.3) and $P(g_c)$ is the *copying generation decoding* probability (see §3.3). $P(x_t^\tau \mid g_p)$ and $P(x_t^\tau \mid g_c)$ are the probabilities of generating x_t^τ under g_p and g_c, respectively.

Reference decoding. Within *reference decoding*, the probability of generating the semantic unit $\{x_{t,i}^\tau\}_{i=1}^n$ is evaluated as follows:

$$
P(x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^n | r) = \alpha_{t,start}^{r_1}\alpha_{t,start+n-1}^{r_2}, \tag{6}
$$

where $\alpha_{t,start}^{r_1}$ and $\alpha_{start+n-1}^{r_2}$ are the probabilities of the start and end tokens of $\{x_{t,i}^\tau\}_{i=1}^n$ (from the background), respectively, which are estimated by two-hop pointers with respect to the context-aware background hidden state sequence $\mathbf{H}^m$. The $\alpha_{t,start}^{r_1}$ is calculated by the first hop pointer, as shown in Eq. 7:

$$
\begin{aligned}
\mathbf{o}_t^1 &= \mathbf{W}_{o_1}[\mathbf{h}_t^s; \mathbf{c}_t^{sc}; \mathbf{c}_t^{sm}] + \mathbf{b}_{o_1},\\
s_{t,j}^{r_1} &= \mathbf{v}_r^{\mathrm{T}} \tanh(\mathbf{W}_r\mathbf{h}_j^m + \mathbf{U}_r\mathbf{o}_t^1 + \mathbf{b}_r),\\
\alpha_{t,start}^{r_1} &= \frac{\exp(s_{t,start}^{r_1})}{\sum_{j=1}^{L_K} \exp(s_{t,j}^{r_1})},
\end{aligned} \tag{7}
$$

where $\mathbf{W}_{o_1}$, $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{v}_r$, $\mathbf{b}_{o_1}$ and $\mathbf{b}_r$ are parameters. $\mathbf{h}_t^s$ is the decoding hidden state vector, the updating scheme of which will be detailed in §3.4. $\mathbf{c}_t^{sc}$ and $\mathbf{c}_t^{sm}$ are calculated in a similar way like Eq. 3 with $\mathbf{h}_t^s$ attentively reading $\mathbf{H}_\tau^c$ and $\mathbf{H}^m$, respectively. The $\alpha_{t,start+n-1}^{r_2}$ is calculated by the second hop pointer, as shown in Eq. 8:

$$\begin{aligned}
\mathbf{c}_t^r &= \sum_{i=1}^{L_K} \alpha_{t,i}^{r_1} \mathbf{h}_i^m,\ \mathbf{o}_t^2 = \mathbf{W}_{o_2}[\mathbf{o}_t^1; \mathbf{c}_t^r] + \mathbf{b}_{o_2}, \\
s_{t,j}^{r_2} &= \mathbf{v}_r^{\mathrm{T}} \tanh(\mathbf{W}_r \mathbf{h}_j^m + \mathbf{U}_r \mathbf{o}_t^2 + \mathbf{b}_r), \\
\alpha_{t,start+n-1}^{r_2} &= \frac{\exp(s_{t,start+n-1}^{r_2})}{\sum_{j=1}^{L_K} \exp(s_{t,j}^{r_2})},
\end{aligned} \tag{8}$$

where $\mathbf{W}_{o_2}$ and $\mathbf{b}_{o_2}$ are parameters. *Reference decoding* adopts soft pointers $\alpha_{t,start}^{r_1}$ and $\alpha_{start+n-1}^{r_2}$ to select semantic units, so it will not influence the automatic differentiation during training.

Generation decoding. Within *predicting generation decoding*, the probability of predicting the token x_t^τ from the vocabulary is estimated as follows:

$$P(x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^1 | g_p) = \mathrm{softmax}(\mathbf{W}_{g_p} \mathbf{o}_t^1 + \mathbf{b}_{g_p}), \tag{9}$$

where $\mathbf{W}_{g_p}$ and $\mathbf{b}_{g_p}$ are parameters and the vector $\mathbf{o}_t^1$ is the same one as in Eq. 7.

Within *copying generation decoding*, the probability of copying the token x_t^τ from the background is estimated as follows:

$$P(x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^1 \mid g_c) = \sum_{i: k_i = x_t^\tau} \alpha_{t,i}^{sm}, \tag{10}$$

where $\alpha_{t,i}^{sm}$ is the attention probability distribution on $\mathbf{H}^m$ produced by the same attention process with $\mathbf{c}_t^{sm}$ in Eq. 7.

3.3 Decoding switcher

The decoding switching probabilities $P(r)$, $P(g_p)$ and $P(g_c)$ are estimated as follows:

$$[P(r), P(g_p), P(g_c)] = \mathrm{softmax}(\mathbf{f}_t), \tag{11}$$

where $\mathbf{f}_t$ is a fusion vector, which is computed through a linear transformation in Eq. 12:

$$\mathbf{f}_t = \mathbf{W}_f[\mathbf{h}_t^s; \mathbf{c}_t^{sc}; \mathbf{c}_t^{sm}] + \mathbf{b}_f, \tag{12}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are parameters. $\mathbf{h}_t^s$ is decoding states (see §3.4).

During testing, at each decoding step, we first compute $P(r)$ and $P(g) = P(g_p) + P(g_c)$. If $P(r) \geq P(g)$, we use Eq. 4 to generate a semantic unit, otherwise we use Eq. 5 to generate a token.

3.4 State updating

The decoding state updating depends on whether the generated unit is a token or semantic unit. If x_{t-1}^τ is a token, then $\mathbf{h}_t^s =$

$$\mathbf{GRU}(\mathbf{h}_{t-1}^s, [\mathbf{e}(x_{t-1}^\tau); \mathbf{c}_{t-1}^{sc}; \mathbf{c}_{t-1}^{sm}]). \tag{13}$$

If x_{t-1}^τ is a span containing n tokens, Eq. 13 will update n times with one token as the input, and the last state will encode the full semantics of a span; see $\mathbf{h}_t^s$ to $\mathbf{h}_{t+1}^s$ in Fig. 2.

The decoding states are initialized using a linear layer with the last state of $\mathbf{H}^m$ and $\mathbf{H}_\tau^c$ as input:

$$\mathbf{h}_0^s = \mathrm{ReLU}(\mathbf{W}_{hs}[\mathbf{h}_{L_K}^m; \mathbf{h}_{L_{C_\tau}}^c] + \mathbf{b}_{hs}), \tag{14}$$

where $\mathbf{W}_{hs}$ and $\mathbf{b}_{hs}$ are parameters. ReLU is the ReLU activation function.

3.5 Training

Our goal is to maximize the prediction probability of the target response given the context and background. We have three objectives, namely generation loss, reference loss and switcher loss.

The *generation loss* is defined as $\mathcal{L}_g(\theta) =$

$$-\frac{1}{M} \sum_{\tau=1}^{M} \sum_{t=1}^{L_{X_\tau}} \log[P(x_t^\tau \mid x_{<t}^\tau, C_\tau, K)], \tag{15}$$

where θ are all the parameters of RefNet. M is the number of all training samples given a background K. In $\mathcal{L}_g(\theta)$, each x_t^τ is a token $\{x_{t,i}^\tau\}_{i=1}^1$.

The *reference loss* is defined as $\mathcal{L}_r(\theta) =$

$$-\frac{1}{M} \sum_{\tau=1}^{M} \sum_{t=1}^{L_{X_\tau}} I(x_t^\tau) \cdot \log[P(x_t^\tau \mid x_{<t}^\tau, C_\tau, K))], \tag{16}$$

where $I(x_t^\tau)$ is an indicator function that equals 1 if $x_t^\tau = \{x_{t,i}^\tau\}_{i=1}^n$ and 0 otherwise.

RefNet introduces a decoding switcher to decide between *reference decoding* and *generation decoding*. To better supervise this process we define *switcher loss* $\mathcal{L}_s(\theta) =$

$$-\frac{1}{M} \sum_{\tau=1}^{M} \sum_{t=1}^{L_{X_\tau}} I(x_t^\tau) \log[P(r)] + (1 - I(x_t^\tau)) \log[P(g)], \tag{17}$$

where $I(x_t^\tau)$ is also an indicator function, which is the same as in $\mathcal{L}_r(\theta)$.

The *final loss* is a linear combination of the three loss functions just defined:

$$\mathcal{L}(\theta) = \mathcal{L}_g(\theta) + \mathcal{L}_r(\theta) + \mathcal{L}_s(\theta). \tag{18}$$

All parameters of RefNet as well as word embeddings are learned in an end-to-end back-propagation training paradigm.

4 Experimental Setup

4.1 Implementation details

We set the word embedding size and GRU hidden state size to 128 and 256, respectively. The vocabulary size is limited to 25,000. For fair comparison, all models use the same embedding size, hidden state size and vocabulary size. Following Moghe et al. (2018), we limit the context length of all models to 65. We train all models for 30 epochs and test on a validation set after each epoch, and select the best model based on the validation results according to BLEU metric.

We use gradient clipping with a maximum gradient norm of 2. We use the Adam optimizer with a mini-batch size of 32. The learning rate is 0.001. The code is available online.[1]

4.2 Dataset

Recently, some datasets for BBCs have been released (Zhou, Prabhumoye, and Black 2018; Dinan et al. 2019). We choose the Holl-E dataset released by Moghe et al. (2018) because it contains boundary annotations of the background information used for each response. We did not use the other datasets because they do not have such annotations for training RefNet. Holl-E is built for movie chats in which each response is explicitly generated by copying and/or modifying sentences from the background. The background consists of plots, comments and reviews about movies collected from different websites. We use the mixed-short background which is truncated to 256 words, because it is more challenging according to Moghe et al. (2018). We follow the original data split for training, validation and test. There are also two versions of the test set: one with single golden reference (SR) and the other with multiple golden references (MR); see (Moghe et al. 2018).

4.3 Baselines

We compare with all methods we can get on this task.

- Extraction-based methods[2]: (i) **BiDAF** extracts a span from background as response and uses a co-attention architecture to improve the span finding accuracy (Seo et al. 2016). (ii) **R-Net** proposes gated attention-based recurrent networks and a self-matching attention mechanism to encode background (Wang et al. 2017). (iii) **QANet** uses an encoder consisting exclusively of convolution and self-attention to capture local and global interactions in background (Yu et al. 2018).
- Generation-based methods: (i) **S2S** maps the context to the response with an encoder-decoder framework (Sutskever, Vinyals, and Le 2014). (ii) **HRED** encodes the context of the conversation with two hierarchical levels (Serban et al. 2016). S2S and HRED do not use any background information. (iii) **S2SA** adds an attention mechanism to the original S2S model to attend to the relevant background information (Bahdanau, Cho, and Bengio 2015). (iv) **GTTP** leverages background information with a copying mechanism to copy a token from the background at the appropriate decoding step (See, Liu, and Manning 2017). (v) **CaKe** is a improved version of GTTP, which introduces a pre-selection process that uses dynamic bi-directional attention to improve knowledge selection from background (Zhang, Ren, and de Rijke 2019). (vi) **AKGCM** first transforms background information into knowledge graph, and uses a policy network to select knowledge with an additional GTTP to generate responses (Liu et al. 2019).

[1]https://github.com/ChuanMeng/RefNet

[2]For fair comparison, different from Moghe et al. (2018), we do not use pre-trained GloVe (Pennington, Socher, and Manning 2014) such that all models randomly initialize the word embedding with the same vocabulary size.

4.4 Evaluation metrics

Following the work of Moghe et al. (2018), we use BLEU-4, ROUGE-1, ROUGE-2 and ROUGE-L as automatic evaluation metrics. We also report the average length of responses outputted by each model. For extraction-based methods and RefNet, we further report F1 (Seo et al. 2016), which only evaluates the extracted spans not the whole responses. We also randomly sample 500 test samples to conduct human evaluations using Amazon Mechanical Turk. For each sample, we ask 3 workers to annotate whether the response is good in terms of four aspects: (1) *Naturalness* (**N**), i.e., whether the responses are conversational, natural and fluent; (2) *Informativeness* (**I**), i.e., whether the responses use some background information; (3) *Appropriateness* (**A**), i.e., whether the responses are appropriate/relevant to the given context; and (4) *Humanness* (**H**), i.e., whether the responses look like they are written by a human.

5 Results

5.1 Automatic evaluation

We list the results of all methods on mixed-short background setting in Table 1.

First, RefNet significantly outperforms all generation-based methods on all metrics, except in the BLEU score compared to AKGCM. Especially, RefNet outperforms the recent and strong baseline CaKe by around 2%-4% (significantly). The improvements show that RefNet is much better at leveraging and locating the right background information to improve responses than these generation-based methods. We believe RefNet benefits from *reference decoding* to tend to produce more complete semantic units, alleviating the inherent problems that pure generation-based method faced.

Second, RefNet outperforms extraction-based methods in most cases, including the strong baseline QANet. We think the reason is that extraction-based methods can only rigidly extracts the relevant spans from the background, which does not consider the conversational characteristics of responses. Differently, RefNet also benefits from the *generation decoding* to generate natural conversational words in responses, which makes up the shortcoming of only extraction. RefNet is comparable in average length with extraction-based methods, which demonstrates that RefNet retains the advantages of extraction-based methods.

Third, the performance of these three extraction-based methods are comparable. However, their performances differ greatly between each other on the RC task dataset SQuAD (Rajpurkar et al. 2016), e.g. QANet outperforms BiDAF by around 7% on F1 score. Even with a stronger extraction-based model, we will arrive at a similar conclusion that they cannot generate natural and fluent responses due to the extraction nature. This confirms that extraction-based methods are not suitable for this task. Besides, we can further enhance the *reference decoding* of RefNet by incorporating various mechanisms used by extraction-based models. But that's beyond the scope of this paper.

Table 1: Automatic evaluation results. **Bold face** indicates leading results. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.05$). SR and MR refer to test sets with single and multiple references. The results of AKGCM are taken from the paper because the authors have not released their code and processed knowledge graph. Note that AKGCM uses GloVe and BERT (Devlin et al. 2019) to improve performance but none of other models do.

Methods	**F1**		**BLEU**		**ROUGE-1**		**ROUGE-2**		**ROUGE-L**		**Average length**
	SR	**MR**	**SR**	**MR**	**SR**	**MR**	**SR**	**MR**	**SR**	**MR**	
no background											
S2S	-	-	5.26	7.11	27.15	30.91	9.56	11.85	21.48	24.81	16.08
HRED	-	-	5.23	5.38	24.55	25.38	7.61	8.35	18.87	19.67	16.22
mixed-short background (256 words)											
BiDAF	40.38	45.86	27.44	33.40	38.79	43.93	32.91	**39.50**	35.09	40.12	25.40
R-Net	40.92	46.84	27.54	33.18	39.78	44.30	32.34	37.65	35.63	40.49	23.08
QANet	41.65	47.32	28.21	33.91	40.66	44.82	**33.62**	39.04	35.29	41.02	23.21
S2SA	-	-	11.71	12.76	26.36	30.76	13.36	16.69	21.96	25.99	16.94
GTTP	-	-	13.65	19.49	30.77	36.06	18.72	23.70	25.67	30.69	14.31
CaKe	-	-	26.03	29.18	40.21	44.12	29.03	34.00	35.01	39.03	20.06
AKGCM	-	-	**30.84**	-	-	-	29.29	-	34.72	-	-
RefNet	**41.86***	**48.46***	30.33	**33.97**	**42.11***	**47.35***	31.35	36.53	**36.70***	**41.88***	23.51

Table 2: Human evaluation results on mixed-short background version. $\geq n$ means that at least n MTurk workers think it is a good response w.r.t. *Naturalness* (**N**), *Informativeness* (**I**), *Appropriateness* (**A**) and *Humanness* (**H**).

	CaKe		QANet		RefNet	
	≥ 1	≥ 2	≥ 1	≥ 2	≥ 1	≥ 2
(**N**)	449	264	288	63	**457**	**299**
(**I**)	359	115	414	225	**434**	**247**
(**A**)	390	153	406	213	**435**	**240**
(**H**)	438	231	355	128	**444**	**242**

Table 3: Analysis of reference and generation decoding on mixed-short background version. **Bold face** indicates leading results. Significant improvements over the best competitor are marked with * (t-test, $p < 0.05$).

	Force reference		Force generation		Combination	
	SR	MR	SR	MR	SR	MR
BLEU	26.73	30.84	26.01	29.19	**30.33***	**33.97***
ROUGE-1	38.03	43.76	39.86	45.53	**42.11***	**47.35***
ROUGE-2	29.06	34.70	28.34	34.07	**31.35***	**36.53***
ROUGE-L	34.11	39.67	35.03	40.63	**36.70***	**41.88***

5.2 Human evaluation

We also conduct a human evaluation for RefNet, CaKe (the best generation-based baseline), and QANet (the best extraction-based baseline). The results are shown in Table 2. Generally, RefNet achieves the best performance in terms of all metrics. In particular, we find that RefNet is even better than CaKe in terms of *Naturalness* and *Humanness*. We believe this is because RefNet has a good trade-off between *reference decoding* and *generation decoding*, where the generated conversational words and the selected semantic units are synthesized in a natural and appropriate way. RefNet is also much better than CaKe in terms of *Appropriateness* and *Informativeness*, which shows that RefNet is better at locating the appropriate semantic units. The reason is that with the ability to generate a full semantic unit at once, RefNet has a global perspective to locate the appropriate semantic units, reducing the risk of breaking a complete semantic unit. QANet achieves good evaluation scores on *Informativeness* and *Appropriateness* than CaKe, but gets the worst scores on *Naturalness* and *Humanness*. Although QANet is relatively good at locating the relevant semantic unit, its responses lack contextual explanations, which makes workers hard to understand. This further shows that only extracting a span from the background is far from enough for BBCs, even replacing QANet with a more stronger extraction-based one.

6 Analysis

6.1 Reference vs. generation decoding

To analyze the effectiveness of reference and generation decoding, we compare the results of RefNet with only reference decoding (*force reference*) and with only generation decoding (*force generation*) in Table 3. Note that *force generation* is better than GTTP because there are two differences[3]. First, we use a matching layer to get the context-aware background representation in Eq. 2, while GTTP only uses basic background representations without such a matching operation. Second, we use the hidden states of the background and context to jointly initialize the decoding states in Equa-

[3]We use the code released by Moghe et al. (2018) https://github.com/nikitacs16/Holl-E

Table 4: Case study. **Bold face** indicates the true span in the current turn.

	Example 1	**Example 2**
	Background: ... but if you like ben stiller , go see " meet the fockers " . dustin 's antics will favorite character was jack (the older one) , because he was so serious but always plotting and putting up a front . i think it was **$ 279,167,575** awards ascap film and television music awards 2005 top box office films mtv ...	**Background**: ...being captured by boris and onatopp . **bond arrives in st . petersburg and meets his cia contact , jack wade (joe don baker) .** wade agrees to take bond to the hide-out of a russian gangster , valentin zukovsky (robbie coltrane) , whom bond had shot in the leg and given a permanent limp years before ...
	Human1: that name is so ridiculous but funny . **Human2**: first off , the writers did not miss a single opportunity to play off of the name " focker " . **Human1**: yeah , i heard it was a pretty successful movie overall .	**Human1**: that was a good seen . **Human2**: what did you like about the movie ? **Human1**: i liked his friend , jack wade .
CaKe	i agree , ben stiller , go see " meet the fockers " .	let them pout and go back to macgyver reruns .
QANet	**$279,167,575**	**bond arrives in st. petersburg and meets his cia contact, jack wade (joe don baker).**
RefNet	it made **$ 279,167,575** at the box office .	i loved the part where **bond arrives in st . petersburg and meets his cia contact , jack wade (joe don baker) .**

Table 5: Analysis of switcher loss on mixed-short background version. **Bold face** indicates leading results. Significant improvements over the best competitor are marked with * (t-test, $p < 0.05$).

	Without SL		With SL	
	SR	MR	SR	MR
F1	37.13	43.42	**41.86***	**48.46***
BLEU	28.96	31.63	**30.33***	**33.97***
ROUGE-1	41.27	46.67	**42.11***	**47.35***
ROUGE-2	30.65	35.98	**31.35***	**36.53**
ROUGE-L	36.02	41.86	**36.70***	**41.88**

tion Eq. 14, while GTTP only uses the single representation of background to initialize it. We can see that force reference and force generation are comparable if working alone. The contributions of reference and generation decoding are complementary as the combination brings further improvements on all metrics, demonstrating the need for both.

6.2 Switcher loss

To verify the effectiveness of the switcher loss $\mathcal{L}_s(\theta)$ in Eq. 17, we compare RefNet with and without training switcher loss, as shown in Table 5. We find that the overall performance increases in terms of all metrics with switcher loss, especially on F1. It means that the switcher loss is an effective component, which better guides the model to choose between *reference decoding* and *generation decoding* at the right time in the right place of a conversation by additional supervision signal. The obvious increase of F1 further shows that at the right time to cite a semantic unit may bring higher accuracy.

6.3 Case study

We select some examples from the test set to illustrate the performance of RefNet, CaKe and QANet, as shown in Table 4. One can see that RefNet can select the right semantic unit from the background or generate fluent tokens at appropriate time and position, resulting in more informative and appropriate responses. For instance, in Example 1, RefNet identifies the right semantic unit "$279,167,575" within the background, which is combined with "it made" ahead and followed by "at the box office" to form a more natural and conversational response. The second example indicates that RefNet can locate longer semantic units accurately. In contrast, the responses by QANet lack naturality. The responses by CaKe are relatively inconsistent and irrelevant. In the first example, CaKe breaks the complete semantic unit "if you like ben stiller " and throws out the part "if you like ".

There are also some cases where RefNet does not perform well. For example, we find that RefNet occasionally selects short or meaningless semantic units, such as "i" and "it." This indicates that we could further improve reference decoding by taking more factors (e.g., the length of semantic units) into consideration.

7 Conclusion and Future Work

In this paper, we propose RefNet for the Background Based Conversation (BBCs) task. RefNet incorporates a novel *reference decoding* module to generate more informative responses while retaining the naturality and fluency of responses. Experiments show that RefNet outperforms state-of-the-art methods by a large margin in terms of both automatic and human evaluations.

A limitation of RefNet is that it needs boundary annotations of semantic units to enable supervised training. In future work, we hope to design a weakly supervised or unsupervised training scheme for RefNet in order to apply it to other datasets and tasks. In addition, we will consider more factors (e.g., the length or frequency of semantic unit) to further improve the reference decoding module of RefNet.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work is supported by the Natural Science Foundation of China (61972234, 61902219, 61672324, 61672322), the Natural Science Foundation of Shandong province (2016ZRE27468), the Tencent AI Lab Rhino-Bird Focused Research Program (JR201932), the Fundamental Research Funds of Shandong University, Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), and the Innovation Center for Artificial Intelligence (ICAI).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 4171–4186.

Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*.

Gao, J.; Galley, M.; Li, L.; et al. 2019. Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval* 13(2-3):127–298.

Ghazvininejad, M.; Brockett, C.; Chang, M.-W.; Dolan, B.; Gao, J.; Yih, W.-t.; and Galley, M. 2018. A knowledge-grounded neural conversation model. In *AAAI*, 5110–5117.

Huang, M.; Zhu, X.; and Gao, J. 2019. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*.

Jiang, S.; Ren, P.; Monz, C.; and de Rijke, M. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In *The Web Conference 2019*.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL*, 1437–1447.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL*, 110–119.

Li, Z.; Niu, C.; Meng, F.; Feng, Y.; Li, Q.; and Zhou, J. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *ACL*, 12–21.

Lian, R.; Xie, M.; Wang, F.; Peng, J.; and Wu, H. 2019. Learning to select knowledge for response generation in dialog systems. In *IJCAI*.

Liu, S.; Chen, H.; Ren, Z.; Feng, Y.; Liu, Q.; and Yin, D. 2018. Knowledge diffusion for neural dialogue generation. In *ACL*, 1489–1498.

Liu, Z.; Niu, Z.-Y.; Wu, H.; and Wang, H. 2019. Knowledge aware conversation generation with reasoning on augmented graph. In *EMNLP*, 1782–1792.

Moghe, N.; Arora, S.; Banerjee, S.; and Khapra, M. M. 2018. Towards exploiting background knowledge for building conversation systems. In *EMNLP*, 2322–2332.

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

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Qin, L.; Galley, M.; Brockett, C.; and Liu, X. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *ACL*, 5427–5436.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2383–2392.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, 1073–1083.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. In *ICLR*.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 3776–3784.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*, 3104–3112.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *NeurIPS*, 2692–2700.

Wang, S., and Jiang, J. 2017. Machine comprehension using match-lstm and answer pointer. In *ICLR*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *ACL*, 189–198.

Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; and Ma, W.-Y. 2017. Topic aware neural response generation. In *AAAI*, 3351–3357.

Xiong, C.; Zhong, V.; and Socher, R. 2017. Dynamic coattention networks for question answering. In *ICLR*.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Zhang, Y.; Ren, P.; and de Rijke, M. 2019. Improving background based conversation with context-aware knowledge pre-selection. In *SCAI*.

Zhou, H.; Young, T.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, 4623–4629.

Zhou, K.; Prabhumoye, S.; and Black, A. W. 2018. A dataset for document grounded conversations. In *EMNLP*, 708–713.