Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism

Chongyang Tao†, Shen Gao†, Mingyue Shang†, Wei Wu‡, Dongyan Zhao‡‡, Rui Yan†∗
† Institute of Computer Science and Technology, Peking University, Beijing, China
‡ Microsoft Corporation, Beijing, China
‡‡ Beijing Institute of Big Data Research, Beijing, China
{chongyangtao,shangmy,zhaody,ruiyan}@pku.edu.cn, 63388@qq.com, wuwei@microsoft.com

Abstract
Attention mechanism has become a popular and widely used component in sequence-to-sequence models. However, previous research on neural generative dialogue systems always generates universal responses, and the attention distribution learned by the model always attends to the same semantic aspect. To solve this problem, in this paper, we propose a novel Multi-Head Attention Mechanism (MHAM) for generative dialogue systems, which aims at capturing multiple semantic aspects from the user utterance. Further, a regularizer is formulated to force different attention heads to concentrate on certain aspects. The proposed mechanism leads to more informative, diverse, and relevant response generated. Experimental results show that our proposed model outperforms several strong baselines.

1 Introduction
To build an intelligent human-computer dialogue system is of growing interest recently. Dialogue systems are generally categorized into task-oriented systems (e.g., agents or virtual assistants) and non-task-oriented systems in the open-domain (e.g., chatbots). Task-oriented dialogue systems aim at helping users to accomplish particular tasks, such as booking a restaurant and vacation scheduling, while chatbot systems are designed to freely converse with humans without any hard limits or domain constraints. As the amount of human-to-human conversational data on social media increases, such data from the open domain drives the development of dialogue systems which are regarded as hot applications in the spotlight.

In this paper, we focus on the generative conversational model for open domain dialogue systems. Most generative conversational models are implemented based on the classic sequence-to-sequence (Seq2Seq) neural network model [Sutskever et al., 2014]. The Seq2Seq model is originally proposed for machine translation and later adapted to various natural language generation tasks, such as text summarization [Rush et al., 2015] and dialogue generation [Mou et al., 2016; Yan et al., 2016; Tian et al., 2017]. However, the standard Seq2Seq model compresses all the necessary information of an input sequence into a fixed-length vector. Its performance drops rapidly as the length of an input sequence increases [Cho et al., 2014]. To address this issue, Bahdanau et al. (2015) proposed an attention mechanism applied onto the decoding sequence, which learns to align the input sequence and the output sequence jointly. Such a mechanism outperforms the basic Seq2Seq model significantly in natural machine translation. Researchers thereafter apply it to response generation for conversations in the open domain, which also yields impressive advances [Shang et al., 2015].

The Seq2Seq model with the attention mechanism seems to be a great success in dialogue systems, but it still has insufficiency. Previous research has revealed that Seq2Seq with attention mechanism based dialogue system tends to suffer from generating trivial and universal responses [Li et al., 2016a]. Recall that the Seq2Seq model with attention mechanism is originally designed for machine translation. Although neural-based machine translation and conversation generation can both be treated as a translation from an input sentence to an output sentence, the impact of attention mechanism in the decoding stage is disparate. In machine translation, the attention mechanism helps to correctly align each target word with the relevant words, which agrees well with human’s intuition. But it is less interpretable when it comes to conversation generation. Compared with machine translation, there is few one-to-one word alignments between the input sentence and the output sentence in conversation generation task. Thus it is not an appropriate strategy for conversation genera-
tion that the decoder focuses on different information at every time-step. Cho et al. (2014) has indicated that the generation process of a better response has a relatively more centralized attention distribution.

Given an utterance, humans tend to focus on certain aspects to respond, rather than disperse attention to every word. We note that an utterance could have many valid responses that focus on different aspects. As shown in Table 1, candidate 1 emphasizes on “go out”, while the attention of candidate 2 is on “eat hotpot”. Thus it is intuitive to guide the decoder, when generating different response, to pay attention to different aspects.

The standard attention mechanism for the Seq2Seq model fails to revolve around different aspects of the input query for conversations. It tries to align different responses with the same aspect of the input utterance and results in a severe problem by aligning universal terms among input and output utterances. What’s more, such attention mechanism is unconstrained and tends to generate more dispersed attention distribution over the input utterance. Researchers have revealed that Seq2Seq model with standard mechanism suffers from generating trivial, universal and informativeness responses [Li et al., 2016], which concur with our observations and explanations.

To address the problem of universal responses and more importantly, to bring diversity into conversations, we propose a Multi-Head Attention Mechanism (MHAM) for Seq2Seq model. To be more specific, we first project hidden states of the encoder to different semantic spaces through learnable projection matrices. Then the standard attention mechanism is applied by the decoder for all semantic spaces to jointly attend to information from different representation subspaces of the encoder hidden states at the decoding time-step. Cho et al. (2014) has indicated that the encoder network sequentially reads the word in X and encodes it as a context vector c through a recurrent neural network (RNN). The decoder network sequentially generates a reply Y = (y1, y2, ···, yn) with context vector c as input. The Seq2Seq models are typically trained with the objection function:

\[ p(Y|X) = \prod_{t=1}^{M} p(y_t|c, y_1, \cdots, y_{t-1}) \] (1)

The encoder RNN computes the context vector c as follows:

\[ h_t = f(x_t, h_{t-1}); \quad c = h_N \] (2)

where \( h_t \) is the hidden state at time \( t \). \( f(\cdot) \) is a non-linear activation function, which can be a logistic function, the sophisticated long short-term memory (LSTM) unit [Hochreiter and Schmidhuber, 1997], or the recently proposed gated recurrent unit (GRU) [Chung et al., 2014]. We employ LSTM as \( f(\cdot) \) in our paper.

The decoder RNN generates word by word conditioned on the context vector \( c \) and the decoder hidden state \( s_t \). The output probability distribution \( o_t \in \mathbb{R}^{D_v} \) (\( D_v \) denotes the vocabulary size) over vocabulary at time \( t \) can be calculated as:

\[ s_t = f(c, y_{t-1}, s_{t-1}) \]

\[ o_t = \text{softmax}(y_{t-1}, s_t) \] (4)

In primitive Seq2Seq model, \( c \) is the same for generating all output words. To alleviate this problem, the attention mechanism [Bahdanau et al., 2015] is usually adopted to allow the decoder to pay different attention to each part of input at every timestep. The attention mechanism computes a different \( c_t \) which is the wighted sum of hidden states of the encoder, \( c_t = \sum_{i=1}^{n} a_{t,i} h_i \), where \( a_{t,i} \) is the attention weight over \( h_i \) at time \( t \) and indicates how much the \( i \)-th word contributes to generating the \( j \)-th word. \( a_{t,i} \) is usually defined as:

\[ e_{t,i} = g(s_t, h_i); \quad a_{t,i} = \text{softmax}(e_{t,i}) \] (5)

where \( g \) is a function that calculates the similarity between \( h_i \) and \( s_t \). In this paper we use bilinear function as \( g(s_t, h_i) = v^T \tanh(W_h h_i + W_s s_t) \), where \( v, W_h \) and \( W_s \) are parameter matrices.

2.2 Multi-head Attention Mechanism

The context vector obtained by traditional attention mechanism focuses on a specific representation subspace of the input sequence. Such context vector is expected to reflect one aspect of the semantics in the input. However, a sentence usually involves multiple semantics spaces, especially for a long sentence. In this paper, we propose a multi-head attention mechanism for Seq2Seq model to allow the decoder RNN to jointly attend to information from different representation subspaces of the encoder hidden states at the decoding
process. The idea of multi-head has been applied to learn the sentence representation in self-attention [Lin et al., 2017; Vaswani et al., 2017].

Formally, we first project hidden states of the encoder to $K$ different semantic spaces through different learnable projection matrices as follows:

$$h^k_i = W^k_p \cdot h_i \quad k \in (1, \ldots, K); i \in (1, \ldots, N)$$  \hspace{1cm} (6)

where $W^k_p \in \mathbb{R}^{d \times d}$ is the learnable projection matrix for the $k$-th semantic space, and $d$ is the dimension of hidden units of the encoder and decoder RNN.

We perform standard attention mechanism for all semantic spaces to obtain multiple attention probability distributions over the input words, and then those distributions are used to generate $K$ different context vectors $\{c^1_t, \ldots, c^K_t\}$ that focus on different components of the input sentence. To be specific, for $k$-th semantic space, $\alpha^k_{t,i}$ denotes the attention weight over the $i$-th encoder hidden state $h^k_i$ at time $t$, which is defined in Eq. 7. Then we can compute the context vector $c^k_t \in \mathbb{R}^d$ through a weighted sum of the encoder hidden states.

$$\alpha^k_{t,i} = \frac{e^{g(h^k_t, s^k_i)}}{\sum_i e^{g(h^k_t, s^k_i)}}; \quad c^k_t = \sum_{i=1}^{T} \alpha^k_{t,i} \cdot h^k_i$$  \hspace{1cm} (7)

For each decoding timestep, we need to combine all context vectors for word generation. Simply, we can employ concatenation strategy or pooling strategy. In this paper, we adopt a soft-attention approach to combine $K$ context vectors. Specifically, we first calculate a weight vector for context vectors conditioned on the final hidden state $h_N$ of encoder. Then the final context vector $c^\text{final}_t$ is obtained by a weighted sum of all context vectors from different semantic spaces.

$$r = \text{softmax}(W_q \cdot h_N)$$  \hspace{1cm} (8)

$$c^\text{final}_t = \sum_{k=1}^{K} r_k \cdot c^k_t$$  \hspace{1cm} (9)

where $W_q \in \mathbb{R}^{K \times d}$ is a trainable parameter and softmax function ensures all weights sum up to 1. $r_k$ is the weight for $k$-th head. We can directly replace $c$ with $c^\text{final}_t$ in Eq. 3 for generating words in the decoder RNN.

### 2.3 Penalty Term

We can notice that there is a potential drawback of the multi-head attention mechanism, i.e., the context vectors can suffer from redundancy problem if the attention mechanism always provides similar attention weight for all semantic spaces. Motivated by recently work [Bousmalis et al., 2016; Lin et al., 2017], we introduce a penalty term, which can not only penalize the redundancy of attention weight vectors across different aspects of the source sentence but also encourage the decoder to attend to a specific aspect consistently.

For each head, we first calculate the average accumulated attention weight on each source word. Formally, the average accumulated attention weight on $i$-th input word for $k$-th head can be calculated as:

$$\delta^k_i = \frac{1}{M} \sum_{t=1}^{M} \alpha^k_{t,i}$$  \hspace{1cm} (10)

where $M$ is the length of the decoder. The above computation is equivalent to performing a mean pooling across different decoding time (input words) and over different semantic spaces.

We can obtain $K$ average accumulated attention weight vectors for all semantic spaces. $\delta^k$ represents the attention vector for $k$-th semantic space and $\sum \delta^k = 1$. We concatenate all those vectors into a matrix $\Delta = \{\delta^1 \oplus \delta^2 \oplus \cdots \oplus \delta^K\} \in \mathbb{R}^{K \times N}$. We define the loss via a soft subspace orthogonality constraint between the attention weight vector of each space (head) as follows:

$$\mathcal{L}_{\text{penalization}} = \|\Delta \cdot \Delta^T - I\|^2_F$$  \hspace{1cm} (11)

where $\| paternal \|_F$ denotes the squared Frobenius norm and $I \in \mathbb{R}^{K \times K}$ is an identity matrix. For any non-diagonal elements $(\Delta \cdot \Delta^T)_{mn}(m \neq n)$, it represents a summation of element-wise product of $\delta^m$ and $\delta^n$. In the extreme case, when the two
attention vectors are orthogonal, \((\Delta \cdot \Delta^T)_{mn}\) will be 0. Otherwise, \((\Delta \cdot \Delta^T)_{mn}\) will be a positive value. We can notice that the elements on the diagonal of \((\Delta \cdot \Delta^T)\) will be forced to approximate 1 since we subtract an identity matrix from \((\Delta \cdot \Delta^T)\). Such a penalty term will encourage attention vector for each head to focus on as few input words as possible. In the most extreme case, the attention vectors for each head all concentrate on a single word and different heads attend to different words.

Finally, the loss function of our model can be defined as:

\[
\mathcal{L} = \lambda \mathcal{L}_{\text{task}} + \gamma \mathcal{L}_{\text{penalization}}
\]

where \(\mathcal{L}_{\text{task}} = -\log p(y|x)\) is the negative log-likelihood loss function for sequence generation. \(\lambda\) and \(\gamma\) are hyper-parameters that control the interaction of the loss terms. \(\lambda + \gamma = 1\). In our model, all parameters are randomly initialized and automatically updated through back-propagation algorithm.

### 3 Experiment

#### 3.1 Dataset

We evaluated our model on a massive Chinese conversation corpus crawled from an online forum Douban\(^1\). After we removed low-quality query-reply pairs, there remain 1,600,243 pairs for model training, 10,000 pairs for model validation, and 2,000 pairs for model testing. We performed Chinese text segmentation by the Jieba word tokenizer\(^2\). The query contains on average 13 words and reply contains on average 11 words.

#### 3.2 Implementation Details

In our model, the vocabulary size is 63,000. We employ 610-dimensional word embeddings, which are randomly initialized in the beginning and trained during the training process, and 1,000-dimensional hidden units in the encoder and decoder RNN, following [Yao et al., 2017]. We utilize Adaptive Sub-gradient Methods (AdaGrad) [Duchi et al., 2011] optimizer on mini-batches of size 32, with learning rate 0.15 and gradient clipping 2. The number of head is \(K = 5\). As the number of heads increases, the semantic subspace of some heads become similar. 5 heads can attend most semantic parts in a query. We train our model in 300K iterations (about 10 epochs) and keep the best model on the validation set. For decoding process, we use beam search with a beam size of 5 and select the top-1 generated reply for evaluation. For the coefficient of penalty term, we take the hyper-parameters which achieve the best performance on the validation set via a small grid search. Finally, we choose \(\lambda\) as 0.95 and \(\gamma\) as 0.05.

#### 3.3 Evaluation

Researchers usually employ BLEU [Papineni et al., 2002] as an evaluation metric for generative dialogue systems. However, BLEU measures word overlap between the generated reply and the ground truth, which is too strict for evaluating dialogue systems due to significant diversity in the space of valid replies to a given context. Besides, [Liu et al., 2016; Tao et al., 2018] conduct empirical experiments and show weak correlation between BLEU and human annotation. In this paper, we consider three embedding-based metrics (including Embedding Average, Embedding Greedy and Embedding Extreme) to evaluate our model, following several recently studies on dialog systems [Serban et al., 2017; Xu et al., 2017]. The three metrics compute the similarity between the generated reply and reference reply according to the word embedding.

We also use human evaluation in our experiment since automatic evaluation metrics may not always consistent with human perception [Stent et al., 2005]. Three educated annotators are invited to judge the quality of 200 randomly sampled replies generated by different models. We show human annotators a dialogue query along with replies generated from each model. Annotators judge the quality of the replies by rating an integer score among 0, 1, and 2. A score of 2 indicates a relevant, natural and informative reply; 1 indicates that the reply is relevant and natural, but is too universal; 0 indicates a bad reply that is either dis-fluent or semantically irrelevant. In our experiment, the average Cohen’s kappa score is 0.3366, indicating that annotators reach good agreement.

#### 3.4 Baselines

We compare our model with several state-of-the-art neural conversation models. RNNatt is a Seq2Seq architecture with soft attention mechanism [Bahdanau et al., 2015], which has been widely adopted as a baseline for comparison. Note that

<table>
<thead>
<tr>
<th>Model</th>
<th>Embedding Average</th>
<th>Embedding Greedy</th>
<th>Embedding Extrema</th>
<th>Human Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNatt [Bahdanau et al., 2015]</td>
<td>0.5469</td>
<td>0.3385</td>
<td>0.4491</td>
<td>0.73</td>
</tr>
<tr>
<td>MMI-anti [Li et al., 2016a]</td>
<td>0.5158</td>
<td>0.3112</td>
<td>0.4149</td>
<td>0.75</td>
</tr>
<tr>
<td>HGFU [Yao et al., 2017]</td>
<td>0.5651</td>
<td>0.3404</td>
<td>0.4639</td>
<td>0.79</td>
</tr>
<tr>
<td>MHAM</td>
<td>0.5705</td>
<td>0.3460</td>
<td>0.4688</td>
<td>0.98</td>
</tr>
<tr>
<td>CMHAM</td>
<td>0.5889</td>
<td>0.3608</td>
<td>0.4830</td>
<td>1.23</td>
</tr>
</tbody>
</table>

Table 2: Reply evaluation using embedding-based metrics as well as human evaluation.

<table>
<thead>
<tr>
<th>Model</th>
<th>distinct-1</th>
<th>distinct-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNatt [Bahdanau et al., 2015]</td>
<td>0.0201</td>
<td>0.0812</td>
</tr>
<tr>
<td>MMI-anti [Li et al., 2016a]</td>
<td>0.0304</td>
<td>0.1484</td>
</tr>
<tr>
<td>HGFU [Yao et al., 2017]</td>
<td>0.0101</td>
<td>0.0537</td>
</tr>
<tr>
<td>MHAM</td>
<td>0.0406</td>
<td>0.1561</td>
</tr>
<tr>
<td>CMHAM</td>
<td>0.0502</td>
<td>0.1749</td>
</tr>
</tbody>
</table>

Table 3: Results of diversity evaluation in terms of system-level diversity.

\(^1\)http://www.douban.com
\(^2\)https://pypi.python.org/pypi/jieba/
this model is a special case of our model when the head $K = 1$ without the penalty term. **MMI-anti** [Li et al., 2016a] is a Seq2Seq model with Maximum Mutual Information (MMI) between inputs and outputs as the objective function, which aims at generating more diverse responses. **HGFU** [Yao et al., 2017] is also a neural generative dialogue model, which incorporates an additional pre-trained cue word into the decoding process in a "soft" manner to generate a more meaningful response.

### 3.5 Experiment Results

Table 2 shows the performance of our model and the baselines in terms of embedding-based evaluation metrics as well as human evaluation. We can see that our proposed models outperform baseline models, which indicates that coupling the Seq2Seq model with a multi-head attention mechanism is a better method for response generation tasks.

We also notice that the performance of our model is better than HGFU [Yao et al., 2017]. This is due to the different content-introducing mechanism. Concretely, a predicted cue word is incorporated into the decoding process in HGFU. Differently, our model introduces auxiliary information from the query itself, which makes our generated reply more relevant to the given query. This can be proved in our case study (see Table 4). Besides, we find that the predicted cue words in HGFU are not always pertinent or appropriate, which further has a direct impact on reply generation.

Furthermore, **CMHAM** achieves better performance than **MHAM** in terms of embedding-based metrics, which demonstrates the effectiveness of the penalty term. As mentioned above, the penalty term encourages the decoder to attend different aspects of the source query, which introduce much more information during the reply generation. The results of human evaluation can also demonstrate the strength of **CMHAM**.

Our conclusions above can also be supported by a case study. As the examples shown in Table 4, we can see that our model can usually generate more meaningful and informative replies. Furthermore, our model appears to be better at generating more relevant replies compared with baseline models. We attribute the improvement of our model to the multi-head attention mechanism which allows the model to attend to information jointly from different representation spaces, so as to better understand the utterance.

### 3.6 Further Analysis

To further investigate the effectiveness of the obtained context vector for each head, we generate $K$ replies from $K$ different heads of attention from model. Concretely, the context vector for $i$-th head $c^i_t$ is concatenated with the hidden state $s_t$ of decoder, and then we use the concatenated vector to predict a word distribution at time $t$. We apply beam search for each head with a beam size of 5 and select the top-1 generated reply as the final output of this head.

We adopt the **distinct-1** and **distinct-2** metrics proposed by Li et al. (2016a) to measure the informativeness and diversity of the generated replies. The **distinct-1** (distinct-2) measures the ratio of distinct unigrams (bigrams for distinct-2). The results are shown in Table 3. It can be seen that our model has the best performance both in **distinct-1** and **distinct-2**. Besides, we can notice that **CMHAM** is better than **MHAM**. This implies our proposed penalty term in objective function can improve the diversity of generated replies.

Table 5 shows an example and its generated replies for different heads. We can observe that five heads generate reasonable replies, and each reply attends to specific semantic aspects of the input query. Figure 2 exhibits the visualization of attention weight of the first four heads. We can see that different attention heads can attend to different semantic parts of the input query. For instance, the generated replies of **Head-1** and **Head-2** mainly focus on “Philips” and “razor” in the query, while **Head-3** and **Head-4** mainly attend to “Taobao” and “affordable”. As the case shown, our model has the capability to generate different replies that focus on specific semantic parts of the input query.

### 4 Related Work

As the general Seq2Seq dialogue model with attention mechanism suffers from generating trivial and universal responses, various attempts have been made to tackle this problem. One line of research has focused on improving the output diversity. Li et al. (2016a) proposed a method that uses Maximum Mutual Information (MMI) between inputs and outputs as the objective function. This approach penalized unconditionally high frequency response to reduce generic responses. Li et al. (2016b) introduced a diversity-promoting decoding algorithm by adding an intra-sibling ranking term to the standard beam search algorithm, which favors choosing hypotheses.
from diverse parents. Such method focuses on diversifying the output of the decoder at word-level. Zhou et al. (2017) proposed a mechanism-aware machine with the framework of encoder-diverter-decoder that models the responding mechanisms as latent embeddings. Recent work from Zhao et al. (2017) combined general Seq2Seq model with conditional variational auto-encoders, which introduces a latent variable to capture discourse-level variations. Different from previous research, our work addresses the universal responses issue by exploring the multi-aspects information in conversations. It is inspired by the intuition that, given an utterance, people are likely to partially focus on a certain aspect.

On the other hand, some researchers have adapted content introducing to alleviate the problem. Xing et al. (2017) incorporated topic information as prior knowledge into the Seq2Seq framework with attention mechanism to encourage the model to generate more topic coherent responses. Mou et al. (2016) presented a method that uses the point-wise mutual information to predict a keyword and makes the word explicitly occur in the generated response. This method is to some extent rigid. Yao et al. (2017) also adapted the approach of predicting a cue word from the query, but it proposed an implicit method to utilize the cue word. However, since the cue word is predicted only by the query, it has the risk of having low relatedness with the whole conversation. Instead of predicting a word, our approach utilizes attention mechanism with multi-head structure to partially focus on words in a previous utterance, which is more intuitive.

Attention mechanisms have become an integral part of the Seq2Seq framework, thus many efforts have been made to improve the attention architecture. Lin et al. (2017) presented a self-attention mechanism to extract different aspects of a sentence into multiple vector-representations. Such method could also be referred as multi-head attention mechanism. Vaswani et al. (2017) also adapted this method in neural machine translation task, as it allows the model to jointly attend to information from different representation subspaces at different positions. To enhance the performance of multi-head mechanism in dialogue systems, our approach incorporates a penalty term to force the attention of each head to concentrate on a certain aspect, and to control the multiplicity.

5 Conclusion

In this paper, we propose a Seq2Seq model with multi-head attention mechanism, which can attend to different semantic parts of an input query for the decoder to explicitly generate reply. We call it Multi-head Attention Aware Dialog System (MHAM). Experiments show that our model outperforms the existing neural-based dialogue models in terms of both automatic evaluation metrics and human judgement.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001) and the National Science Foundation of China (No. 61672058). Rui Yan was sponsored by CCF-Tencent Open Research Fund and Microsoft Collaborative Research Program.
References


