Reinforcing Coherence for Sequence to Sequence Model in Dialogue Generation

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu and Xueqi Cheng
University of Chinese Academy of Sciences, Beijing, China
CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences
zhanghainan@software.ict.ac.cn, {lanyanyan, guojiafeng, junxu, cxq}@ict.ac.cn

Abstract
Sequence to sequence (Seq2Seq) approach has gained great attention in the field of single-turn dialogue generation. However, one serious problem is that most existing Seq2Seq based models tend to generate common responses lacking specific meanings. Our analysis show that the underlying reason is that Seq2Seq is equivalent to optimizing Kullback–Leibler (KL) divergence, thus does not penalize the case whose generated probability is high while the true probability is low. However, the true probability is unknown, which poses challenges for tackling this problem. Inspired by the fact that the coherence (i.e. similarity) between post and response is consistent with human evaluation, we hypothesize that the true probability of a response is proportional to the coherence degree. The coherence scores are then used as the reward function in a reinforcement learning framework to penalize the case whose generated probability is high while the true probability is low. Three different types of coherence models, including an unlearned similarity function, a pretrained semantic matching function, and an end-to-end dual learning architecture, are proposed in this paper. Experimental results on both Chinese Weibo dataset and English Subtitle dataset show that the proposed models produce more specific and meaningful responses, yielding better performances against Seq2Seq models in terms of both metric-based and human evaluations.

1 Introduction
This paper focuses on the problem of single-turn dialogue generation, which is expected to automatically generate an appropriate response for a given post. Following conventional data-driven generation framework of statistical machine translation, most existing neural conversation models are based on a Seq2Seq architecture [Sutskever et al., 2014]. In these models, a recurrent neural network (RNN) encoder is first utilized to encode the input post to a vector, and another RNN decoder is then used to generate the response. To learn the model parameters, a maximum likelihood estimation approach is applied on the training data which consists of many post-response pairs. The intrinsic philosophy is that the true probability would be estimated by the generated probability with proper parameters.

Though Seq2Seq has the ability to generate fluent responses, one serious problem is that the generated responses are usually common, such as ‘I do not know’, ‘What does this mean?’ and ‘Haha’ [Li et al., 2016a; Mou et al., 2017]. Clearly, these kinds of responses lack specific meanings for further widening and deepening of the dialogue, which will have a bad effect on the users’ experience. Through our analysis, the main reason is that the objective of Seq2Seq is equivalent to minimizing the KL divergence between the generated probability and the true probability. However, KL divergence is not symmetric, thus it will not penalize the case whose generated probability is high while the true probability is low, which is exactly the case of common responses.

In this paper, we propose to utilize the coherence (i.e. similarity) between the generated responses and the original post as an estimation of the true probability, with inspiration comes from the fact that the similarity measure between post and response embeddings is consistent with human evaluation. Specifically, three kinds of coherence models are adopted in this paper. Firstly, an unlearned similarity function, such as cosine similarity, can be directly used as the coherence model. Secondly, the previous semantic text matching models can be regarded as good candidates for measuring the coherence between a post and its corresponding response. In this paper, we use two pretrained matching functions, i.e., GRU bilinear model [Socher et al., 2013] and MatchPyramid [Pang et al., 2016], which are representatives of two different kinds of deep matching models, i.e., representation-focused methods and interaction-focused methods. Thirdly, an end-to-end dual learning architecture similar to [Xia et al., 2016] can be adopted to jointly learn the parameters of response generation model and coherence model. After that, the coherence model is used as the reward function in a reinforcement learning framework for optimization, which will guide the learning process to penalize the case whose generated probability is high while the true probability is low.

We evaluate the proposed models on two public datasets, i.e. the Chinese Weibo and the English Subtitle dataset. Experimental results show that our models significantly outper-
form traditional Seq2Seq models and their variations with respect to both metric-based evaluations [Li et al., 2016a; Shen et al., 2017] and human judgments.

2 Related Work

The basic neural-based Seq2Seq framework for dialogue generation is inspired by the studies of statistical machine translation (SMT). Sutskever et al. [Sutskever et al., 2014] proposed the original Seq2Seq framework, which used a multi-layered Long Short-Term Memory (LSTM) to map the input sequence to a fixed dimension vector and then used another LSTM to decode the target sequence from the vector. At the same time, Cho et al. [Cho et al., 2014] followed the above architecture, and proposed to feed the last hidden state of encoder to every cell of decoder, which enhanced the influence of contexts in generation. To further alleviate the long dependency problem, Bahdanau et al. [Bahdanau et al., 2015] introduced the attention mechanism into the neural network and achieved encouraging performances. Many studies [Shang et al., 2015; Vinyals and Le, 2015] applied the above neural SMT models to dialogue generation, and gained promising performances.

Although the current Seq2Seq model is capable to generate fluent responses, these responses are usually general, such as ‘I don’t know’, ‘interesting’, and ‘what is it’. Li et al. [Li et al., 2016a] proposed a mutual information model to tackle this problem. However, it is not a unified training model, instead it still trained maximum likelihood model, and used the Maximum Mutual Information criterion only for testing to rerank the primary top-n list generated by Seq2Seq. Mou et al. [Mou et al., 2017] proposed a forward-backward keyword method which used a pointwise mutual information to predict a noun as a keyword and then used two Seq2Seq models to generate the forward sentence and the backward sentence. Xing et al. [Xing et al., 2017] proposed a joint attention mechanism model which modified the generation probability by adding the topic keywords likelihood to the generated maximum likelihood. It has to train an extra LDA model from an extra corpus to generate the topic keyword candidates. However, if the new posts are not in these topics, the user has to re-trained the LDA model to adapt the new data. Recently, SeqGAN [Yu et al., 2017] and Adver-REGS [Li et al., 2017] tried to use GAN for generation, where the discriminator score is used as a reward function. However, the meaning of this kind of reward function is not clear.

3 Sequence to Sequence Framework

We first introduce the typical LSTM-based Seq2Seq model [Bahdanau et al., 2015] used in dialogue generation.

Given a post \( X = \{x_1, \ldots, x_M \} \) as the input, a standard LSTM first maps the input sequence to a fixed-dimension vector \( h_M \) as follows.

\[
\begin{align*}
    i_k &= \sigma(W_i[h_{k-1}, w_k]), \\
    f_k &= \sigma(W_f[h_{k-1}, w_k]), \\
    o_k &= \sigma(W_o[h_{k-1}, w_k]), \\
    c_k &= f_k c_{k-1} + i_k l_k, \\
    h_k &= o_k \tanh(c_k),
\end{align*}
\]

where \( i_k, f_k \) and \( o_k \) are the input, memory, and output gate, respectively. \( w_k \) is the word embedding for \( x_k \), and \( h_k \) stands for the vector computed by LSTM at time \( k \) by combining \( w_k \) and \( h_{k-1} \). \( c_k \) is the cell at time \( k \), and \( \sigma \) denotes the sigmoid function. \( W_i, W_f, W_o \) and \( W_l \) are parameters.

Then another LSTM is used as the decoder to map the vector \( h_M \) to the ground-truth response \( Y = \{y_1, \ldots, y_N\} \). Given the context vector \( h_M \) and the previous generated words \( \{y_1, \ldots, y_{k-1}\} \), the decoder is typically trained to predict the next word \( y_k \). In other words, the decoder defines a probability over the output \( Y \) by decomposing the joint probability into conditionals by the chain rule in probability theory.

Usually the attention mechanism is further introduced to the above Seq2Seq framework. Instead of directly using \( h_M \) as the context vector in the decoder, we let the new context vector, denoted as \( s_k \), to be dependent on the whole sequence \( \{h_1, \ldots, h_M\} \), where each \( h_k \) contains information about the input sequence with a strong focus on the parts surrounding the \( k \)-th word of the input sentence. Specifically, the context vector \( s_k \) is usually defined as a weighted sum of these \( h_k \):

\[
    s_k = \sum_{k=1}^{M} \alpha_{ik} h_k.
\]

The weight \( \alpha_{ik} \) of each representation \( h_k \) is computed as:

\[
    \alpha_{ik} = \frac{\exp(e_{ik})}{\sum_{j=1}^{M} \exp(e_{ij})},
\]

\[
    e_{ik} = v^T \tanh(W_1 h_{i-1} + W_2 h_k),
\]

where \( v^T, W_1 \) and \( W_2 \) are learned parameters. \( e_{ik} \) is an alignment model which scores how well the inputs around position \( k \) and the output at position \( i \) match, which is based on the LSTM hidden state \( h_{i-1} \) (just before emitting \( y_i \)), and \( h_k \) of the input sentence.

Given a set of training data \( D = \{(X,Y)\} \), Seq2Seq assumes that these data are i.i.d. sampled from the probability \( P_g \), and the following negative log likelihood is used as the objective for minimization.

\[
    \mathcal{L} = - \sum_{(X,Y) \in D} \log P_g(Y|X).
\]

4 Motivation

Though the above Seq2Seq model has the ability to generate fluent responses for a given post, the responses are usually common [Li et al., 2016a; 2016b; 2017]. Through our data analysis, we find that the empirical probabilities of these generated responses are very low. Specifically, we define two metrics, i.e. hit rate and hit probability as follows.

\[
    Hit_R = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n} \sum_{j=1}^{n} I((X_i, Y_{ij}) \in \mathcal{D}),
\]

\[
    Hit_P = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n} \sum_{j=1}^{n} I((X_i, Y_{ij}) \in \mathcal{D}) \times P_c(Y_{ij}|X_i)
\]

where \( X_i, i = 1, \ldots, m \) stands for the \( i \)-th posts in the training data, and \( Y_{ij}, j = 1, \ldots, n \) stands for the \( j \)-th generated response with respect to the post \( X_i \), \((X_i, Y_{ij}) \in \mathcal{D}) \) is an
The simplest way to measure the coherence is to directly use an unlearned similarity function. For example, we can use the cosine function to act as the coherence model.

$$r_{cos}(X,G) = \frac{\langle h(X), h(G) \rangle}{\|h(X)\| \|h(G)\|}$$

where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the inner product and $L_2$ norm, respectively. $h(X)$ and $h(G)$ stand for the sentence representations of post and generated response. Following previous practice [Shen et al., 2017], we directly use the mean over the word embeddings in the sentence as the sentence representations in this paper.

### 5.2 Pretrained Semantic Matching Functions

Recently, extensive semantic text matching models [Pang et al., 2016; Wan et al., 2016b] have been proposed to capture the coherence relationship between two texts. In this paper, we utilize two pretrained matching function, i.e., bilinear model [Socher et al., 2013] with GRU and MatchPyramid [Pang et al., 2016] to measure the coherence. The reason to choose these models is that they are representatives of the two different categories of text matching models [Guo et al., 2016], i.e., representation focused methods and interaction focused methods.

---

**Table 1:** The average similarity(%) of post and its generated responses with human score on STC.

<table>
<thead>
<tr>
<th>human score</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cosine similarity</td>
<td>60.19</td>
<td>57.39</td>
<td>60.30</td>
<td>62.06</td>
<td>66.41</td>
</tr>
</tbody>
</table>

---

1. the response is nonfluent or logically wrong;
2. the response is fluent but not related with the post, including the case when some un-related common response;
3. the response is fluent and weakly related, but it’s common which can response many other posts;
4. the response is fluent and strongly related with its post;
5. the response is fluent and strongly related, which is like following a real person’s tone.

As shown in Table 1, the similarity measure is consistent with the human evaluation. This finding inspires us the true probability of a response is highly likely to be proportional to the coherence score (i.e. similarities) between this response and its post. Therefore, it is natural to utilize the coherence score to act as a reward function to penalize the case whose generated probability is high while the true probability is low.

---


2In order to eliminate the influence of same X and Y which similarity score is perfect but the human score is not likely to be high, we didn’t calculate their similarity score in this results.
**GRU Bilinear Model**

GRU bilinear model is a typical representation focused semantic matching model, which focuses on learning the semantic representations of each sentence. Given the post $X$ and the generated response $G$, a GRU model is utilized to encode them to sentence embeddings $h(X)$ and $h(G)$. Then a bilinear function is used to model the transformation from post to response:

$$r_{bilinear}(X, G) = h(X) \cdot W \cdot h(G),$$

where $W$ is the transformation matrix.

**MatchPyramid**

MatchPyramid [Pang et al., 2016] is a typical interaction focused semantic matching model, which is capable to capture different levels of interaction signals. For the dialogue generation task, the coherence pattern between post and response may not lie in the global sentence-level, as assumed in cosine similarity and GRU bilinear, but in some local evidences. For example, post and responses usually have some similar keywords or phrases [Mou et al., 2017]. Therefore, MatchPyramid can also be viewed as a suitable coherence model, denoted as $r_{mp}$. Specifically, the word embedding matrix between $X$ and $G$ is first constructed, and a convolutional neural network is then applied. Finally, a multiple layer perceptron is conducted to output the matching score $r_{mp}(X, G)$.

For pretraining the above GRU-bilinear and MatchPyramid models, we follow the practice of traditional semantic matching [Wan et al., 2016a] and use a pairwise loss function for optimization. Specifically, for each post and positive (ground-truth) response pair in the training data, we randomly sample five negative responses, chosen from other post’s ground-truth responses, to construct the training data. Then the training criterion is to learn appropriate parameters to rank the positive response higher than the negative ones.

### 5.3 Dual Learning Architecture

Both GRU bilinear model and MatchPyramid are capable to capture the complex coherence relationship between post and response, however, the learning of coherence and maximum likelihood are isolated. In this paper, we propose an end-to-end dual learning architecture to jointly learn the parameters of coherence and maximum likelihood. Dual learning is first proposed by Xia et al. [Xia et al., 2016] for statistical machine translation. We apply it here because it is suitable to fulfill the requirement to represent the coherence between post and generated response, which can be described as two-agent communication process:

1. The first agent understands the post, and sends a message to the second agent, which converts the message from post to response.
2. The second agent understands the response, and measures the coherence of the message from the post and the response, which is propagated to the first agent.
3. According to the coherence measure suggested by the second agent’s, the first agent modifies the dialogue generation model.
4. The above three steps can also be started with the second agent to obtain a dual joint learning architecture.

The dual-learning agents are two standard sequence to sequence models: the first agent generates the response for the given post and the second agent generates post for the given response. In the first step, given the post $X$, the first agent generates the response $G_1$ and sends $G_1$ to the second agent. The second agent calculates the conditional probability of $P(X|G_1)$ as its coherence measure and tells the first agent. Finally, the first agent uses the coherence measure to guide the response generation process. The second agent then starts the communication. For the given ground-truth response $Y$, it first generates the response $G_2$ and sends it to the first agent. The first agent calculates the conditional probability of $P(Y|G_2)$ as its coherence measure, and tells the second agent. Finally, the second agent uses the coherence measure to guide the post generation process. The above process repeated coordinately until convergence. Therefore, there are two coherence models $r_{dual1}(X, G)$ and $r_{dual2}(X, G)$, which can be defined as follows.

$$G_1 = \arg \max P_1(G_1|X),$$
$$r_{dual1}(X, G_1) = \log P_2(X|G_1),$$
$$G_2 = \arg \max P_2(G_2|Y),$$
$$r_{dual2}(Y, G_2) = \log P_1(Y|G_2),$$

where $P_2(X|G_1)$ is the probability of generating post $X$ given the sentence $G_1$ generated by the first agent, and $P_1(Y|G_2)$ is the probability of generating response $Y$ given the sentence $G_2$ by the second agent, where $P_1$ and $P_2$ are two different Seq2Seq models.

### 5.4 Optimization

Inspired by recent work of [Li et al., 2016b; 2017; Yu et al., 2017], we treat the problem of generating high quality responses with maximum likelihood as a reinforcement learning problem, in which the coherence model acts as the reward function and is observed when the model arrives at the end of the sequence. The reinforcement learning model $P_{RL}$ is initialized by the pretrained Seq2Seq with attention model $P_{Seq2Seq}$. Specifically, an action is the dialogue response to generate. The action space is infinite since arbitrary-length sequences can be generated. A state is denoted by the post in the single-turn dialogue generation, which is usually transformed to a vector representation. A policy takes the form of an LSTM encoder-decoder, and is defined by its parameters. Note that we are using a stochastic representation of the policy, i.e., a probability distribution over actions given states. The reward function is defined by the coherence model for each action. For the unlearned similarity function and pre-trained semantic matching functions, the reinforcement learning architecture is shown in Figure 1(a). Given an input post $X$, Seq2Seq first takes an action to generate a response $G$, and then it obtains the reward $r(X, G)$ from the coherence models. We use the policy gradient methods [Sutton et al., 2000] for optimization, with the expected reward defined as:

$$J(\theta) = \mathbb{E}[r(X, G)].$$

(4)

The gradient is estimated as:

$$\nabla J(\theta) = r(X, G) \nabla \log P_{RL}(G|X),$$

(5)
Please note that the coherence scores of the pretrained matching functions need to be normalized, due to the varied ranges. In this paper we use min-max normalization for \( r_{bi} \) and \( r_{mp} \). Take bilinear model as an example, we randomly select some negative sentences \( \{G'_1, \ldots, G'_m\} \) generated from Seq2Seq, and then calculate the scores: \( \mathcal{S} = \{r_{bi}(X, G), r_{bi}(X, G'_1), \ldots, r_{bi}(X, G'_m)\} \). The min-max normalized coherence score \( r(X, G) \) is then defined as follows:

\[
r(X, G) = \frac{r_{bi}(X, G) - \min(p)}{\max(p) - \min(p)},
\]

where \( \min(p) \) and \( \max(p) \) are the minimum value and maximum value of \( \mathcal{S} \).

The reinforcement learning process of dual learning architecture is more complex, as shown in Figure 1(b). There are two opposite reinforcement learning processes. For the first reinforcement learning process, the first agent (post→response) generates a response \( G_1 \) for a given the input post \( X \), and then obtains the reward \( r_{dual1} \) from the second agent. The expected reward and gradient are defined as:

\[
J_1(\theta) = \mathbb{E}[r_{dual1}(X, G_1)],
\]

\[
\nabla J_1(\theta) = r_{dual1}(X, G_1) \nabla \log P_{\text{RL}}(G_1|X).
\]

For the second reinforcement learning process, the second agent (response→post) generates a post \( G_2 \) for a given response \( Y \), and then receives the reward \( r_{dual2} \) from the first agent. The expected reward and gradient are defined as:

\[
J_2(\theta) = \mathbb{E}[r_{dual2}(Y, G_2)],
\]

\[
\nabla J_2(\theta) = r_{dual2}(Y, G_2) \nabla \log P_{\text{RL}}(G_2|Y).
\]

Both aforementioned gradients are plugged into the policy gradient methods [Sutton et al., 2000] to optimize the reinforcement learning process, shown in Fig. 1.

6 Experiments

6.1 Experimental Settings

Datasets. We use two public datasets in our experiments. The Chinese Weibo dataset, named STC [Shang et al., 2015], consists of 3,788,571 conversational post-response pairs extracted from the Chinese Weibo website and cleaned by the data publishers. We randomly split the data to training, validation, and testing sets, which contains 3,000,000, 388,571 and 400,000 pairs, respectively. We also use an English conversation data, named OpenSubtitles3 (OSDb) dataset, to test our model. OSDb is a large, open-domain dataset containing roughly 60M-70M scripted lines spoken by movie characters. We randomly selected a subset of OSDb dataset in our experiment, which contains 3,800,000 post-response pairs. We split the data to 3,000,000, 400,000 and 400,000 for training, validation and testing, respectively.

Baseline Methods. Five baseline methods are used in the comparison, including traditional Seq2Seq [Sutskever et al., 2014], RNN autoencoder(RNN-encdec) [Cho et al., 2014], Seq2Seq with attention(Seq2Seq-att) [Bahdanau et al., 2015], MMI [Li et al., 2016], Back-MMI [Li et al., 2016b] and Adver-REGS [Li et al., 2017]. Since we have designed three kinds of coherence models, we have four versions of our model, denoted as Seq2SeqCo-cos, Seq2SeqCo-bi, Seq2SeqCo-MP, Seq2SeqCo-dual, respectively. We first introduce the input embeddings. For STC, we utilize character-level embeddings trained from the STC training dataset rather than word-level embeddings, due to the word sparsity, segmentation mistakes and unknown Chinese words which lead to inferior performance than character-level [Hu et al., 2015]. For OSDb, we use word embeddings trained by word2vec on a large Wikipedia corpus.

In the training process, the dimension is set to be 300, the size of negative sample is set to be 3, and the learning rate is 0.05. Then we introduce the settings on learning parameters in the deep architecture. For a fair comparison among all the baseline methods and our methods, the numbers of hidden nodes are all set to 300, and batch sizes are set to 200. Stochastic gradient decent (SGD) is utilized in our experiment for optimization, instead of Adam, because SGD yields better performances in our experiments. The learning rate is set to be 0.5, and adaptively decays with rate 0.99 in the optimization. We run our model on a Tesla K80 GPU card with Tensorflow.

Evaluation Measures. We use both quantitative metrics and human judgements in section 4 to evaluate the proposed models. Specifically, we use two kinds of metrics for quantitative comparisons. The first one kind is traditional metrics, such as PPL and BLEU score[Xing et al., 2017]. They are both widely used in natural language processing, and here we use them to evaluate the quality of the generated responses. The other kind is to evaluate the degree of diversity of the generated responses. In this paper, we use distinct [Li et al., 2016a; 2016b], which calculates the number of distinct unigrams and bigrams in the generated responses. If a model often generates common responses, the distinct will be low.

6.2 Experimental Results

The quantitative evaluation results are shown in Table 2. From the results, we can see that both Back-MMI and Adver-REGS outperform traditional Seq2Seq baselines in terms of BLEU, PPL and distinct measures. That’s because both Back-MMI and Adver-REGS further consider some reward
functions in the optimization process. Back-MMI uses a predefined reward function to penalize generating common responses, but the effect is quite limited. Adver-REGS uses a learned discriminator to define the reward function, though flexible, it is not clear whether the reward function truly penalizes the common responses. Among the four proposed coherence models, the end-to-end dual learning approach performs the best, because of its capability to jointly learning all parameters for fitting the true distribution of the large scale conversation data. Our models obtain higher BLEU and lower PPL than baseline models. Take the BLEU score on STC dataset for example, the BLEU score of dual learning model is 2.925, which is significantly better than that of Back-MMI and Adver-REGS, i.e., 2.917 and 2.918. These results indicate that our models generate responses with higher quality. The distinct scores of our models are also higher than baseline models, which indicate that our models can generate more specific responses. That’s because we directly penalize the case when generated probability is high and true probability is low. We also conducted the significant test, and the result shows that the improvements of our model are significant on both Chinese and English datasets, i.e., p-value < 0.01. In summary, our coherence models produce more fluent and diverse results, as compared with baseline methods.

The human evaluation results are shown in Table 3, in which the percentage of sentences belonging to each grade and the averaged grade are given to evaluate the quality of generated responses, and kappa [Fleiss, 1971] value is also given to demonstrate the consistency of different annotators. From the results, our four coherent models significantly outperform baseline methods. Take STC as an example. The averaged score of Seq2SeqCo-dual is 3.3, which is significantly better than that of Back-MMI and Adver-REGS, i.e., 3.2 and 3.0, respectively. The percentage of strongly related sentences (i.e., the sum of grade ‘4’ and ‘5’) of Seq2SeqCo-dual is 58.8%, which is significantly better than that of Back-MMI and Adver-REGS, i.e., 54.2% and 47.1%.

Here we show some generated responses for demonstration. Specifically, Table 4 gives one post, its ground-truth responses, and the generated responses from different models. We can see that Seq2Seq-att, MMI, Back-MMI and Adver-REGS all produce common responses, such as ‘It is amazing’, ‘I don’t care’ and ‘I don’t know what you mean’. However, our models give interesting responses with specific meanings. We have obtained similar observations for many other posts, but we have to omit them for space limitations.

We also computed the hit rate and hit probability as defined in our motivation. The values on the dataset of STC are 0.008449 and 0.001689 for Seq2SeqCo-dual, which are much larger than Seq2Seq-att, as described before. Therefore, our coherence models have the ability to generated responses with higher true probability.

<table>
<thead>
<tr>
<th>model</th>
<th>BLEU</th>
<th>PPL</th>
<th>distinct-1</th>
<th>distinct-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>STC Dataset</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>2.867</td>
<td>21.28</td>
<td>0.414</td>
<td>4.52</td>
</tr>
<tr>
<td>RNN-encdec</td>
<td>2.878</td>
<td>21.08</td>
<td>0.381</td>
<td>4.87</td>
</tr>
<tr>
<td>Seq2Seq-attn</td>
<td>2.894</td>
<td>19.86</td>
<td>0.437</td>
<td>5.03</td>
</tr>
<tr>
<td>MMI</td>
<td>2.906</td>
<td>18.65</td>
<td>0.431</td>
<td>5.43</td>
</tr>
<tr>
<td>Back-MMI</td>
<td>2.917</td>
<td>18.57</td>
<td>0.441</td>
<td>5.82</td>
</tr>
<tr>
<td>Adver-REGS</td>
<td>2.918</td>
<td>18.57</td>
<td>0.433</td>
<td>5.74</td>
</tr>
<tr>
<td>Seq2SeqCo-cos</td>
<td>2.919</td>
<td>18.65</td>
<td>0.445</td>
<td>5.68</td>
</tr>
<tr>
<td>Seq2SeqCo-bi</td>
<td>2.918</td>
<td>18.64</td>
<td>0.438</td>
<td>5.87</td>
</tr>
<tr>
<td>Seq2SeqCo-MP</td>
<td>2.918</td>
<td>18.56</td>
<td>0.448</td>
<td>5.79</td>
</tr>
<tr>
<td>Seq2SeqCo-dual</td>
<td>2.925</td>
<td>18.52</td>
<td>0.440</td>
<td>5.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>model</th>
<th>human score distribution(%)</th>
<th>Ave.</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>STC Dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNN-encdec</td>
<td>17.4 40.2 21.7 18.5 2.2</td>
<td>2.48</td>
<td>0.454</td>
</tr>
<tr>
<td>Seq2Seq-attn</td>
<td>18.7 34.1 20.9 19.8 6.6</td>
<td>2.62</td>
<td>0.517</td>
</tr>
<tr>
<td>MMI</td>
<td>13.6 30.0 12.4 39.2 4.8</td>
<td>2.92</td>
<td>0.461</td>
</tr>
<tr>
<td>Back-MMI</td>
<td>12.6 21.8 11.4 41.4 12.8</td>
<td>3.2</td>
<td>0.469</td>
</tr>
<tr>
<td>Adver-REGS</td>
<td>13.7 25.5 13.7 41.2 5.9</td>
<td>3.00</td>
<td>0.456</td>
</tr>
<tr>
<td>Seq2SeqCo-cos</td>
<td>11.6 28.0 9.6 42.8 8.0</td>
<td>3.08</td>
<td>0.473</td>
</tr>
<tr>
<td>Seq2SeqCo-bi</td>
<td>11.6 23.6 9.6 42.8 12.4</td>
<td>3.21</td>
<td>0.428</td>
</tr>
<tr>
<td>Seq2SeqCo-MP</td>
<td>11.2 27.2 9.4 42.0 8.4</td>
<td>3.11</td>
<td>0.426</td>
</tr>
<tr>
<td>Seq2SeqCo-dual</td>
<td>11.2 20.8 9.2 44.4 14.4</td>
<td>3.30</td>
<td>0.422</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>model</th>
<th>human score distribution(%)</th>
<th>Ave.</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSDb Dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNN-encdec</td>
<td>2.316 26.13 1.24 5.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq-attn</td>
<td>2.310 26.02 1.15 5.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMI</td>
<td>2.282 25.99 1.38 5.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Back-MMI</td>
<td>2.323 25.78 1.45 6.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adver-REGS</td>
<td>2.413 25.1 1.49 6.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2SeqCo-cos</td>
<td>2.393 25.73 1.33 6.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2SeqCo-bi</td>
<td>2.422 25.67 1.27 6.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2SeqCo-MP</td>
<td>2.397 25.3 1.52 6.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2SeqCo-dual</td>
<td>2.422 24.45 1.42 6.83</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The metric-based evaluation results(%).

Table 3: The human evaluation on STC(top) and OSDb(bottom).
7 Conclusion

In this paper, we first analyzed the problem of Seq2Seq in generating common responses. We find that the main reason is that the objective of Seq2Seq is equivalent to KL divergence, which does not penalize the case whose generated probability is high while the true probability is low. Then we hypothesized that the true probability can be estimated by the coherence score between post and response, inspired by our statistical findings that the similarity measure between post and response embeddings is consistent with human evaluation. Then we defined three kinds of coherence models, and used them as the reward function in a reinforcement learning framework. Experimental results on both Chinese Weibo and English Subtitle dataset showed that our models significantly outperform baselines, including traditional Seq2Seq and some recent proposed models which are related.

Acknowledgments

This work was funded by the 973 Program of China under Grant No. 2014CB340401, the National Natural Science Foundation of China (NSFC) under Grants No. 61425016, Grant No. 2014CB340401, the National Natural Science Foundation of China under Grants No. 2016QY02D0405, and 2016102, and the National Key R&D Program Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, and the National Key R&D Program of China under Grants No. 2016QY02D0405.

References


