A Prospective-Performance Network to Alleviate Myopia in Beam Search for Response Generation

Zongsheng Wang[1], **Yunzhi Bai**[2*], **Bowen Wu**[1], **Zhen Xu**[3*], **Zhuoran Wang**[1,4], **Baoxun Wang**[1]
[1]Tricorn (Beijing) Technology Co., Ltd, Beijing, China
[2]Telecom ParisTech, Paris, France
[3]Harbin Institute of Technology, Harbin, China
[4]Institute of Internet Industry, Tsinghua University, Beijing, China
[1] {wangzongsheng, wubowen, wangzhuoran, wangbaoxun}@trio.ai
[2] yunzhi_bai@outlook.com
[3] zxu@insun.hit.edu.cn

Abstract

Generative dialog models usually adopt beam search as the inference method to generate responses. However, small-width beam search only focuses on the limited current optima. This deficiency called myopic bias ultimately suppresses the diversity and probability of generated responses. Although increasing the beam width mitigates the myopic bias, it also proportionally slows down the inference . To alleviate the myopic bias in small-width beam search, this paper proposes a Prospective-Performance Network (PPN) to predict the future reward of the given partially-generated response, and the future reward is defined by the expectation of the partial response appearing in the top-ranked responses given by a larger-width beam search. Enhanced by PPN, the decoder can promote the results with great potential during the beam search phase. The experimental results on both Chinese and English corpora show that our method is capable of increasing the quality and diversity of generated responses, with inference efficiency well maintained.

1 Introduction

In recent years, Neural Response Generation (NRG) (Vinyals and Le, 2015; Shang et al., 2015) with sequence-to-sequence (Seq2Seq) structures (Sutskever et al., 2014a; Bahdanau et al., 2015) has been widely studied and adopted in open-domain dialog systems such as XiaoIce (Shum et al., 2018). Many studies have been done to generate target responses meeting desirable proprieties including emotion (Zhou et al., 2017), diversity (Li et al., 2016), etc., or to explore issues in decoding step such as exposure bias, loss-evaluation mismatch (Ranzato et al., 2016) and label bias (Wiseman and Rush, 2016).

Most NRG systems adopt the beam search algorithm to generate responses given queries. In brief, beam search explores the possible responses by storing only top-ranked ones as candidates at each time step. Though it is a useful prediction strategy, depending on beam width, beam search more or less suffers from its nature of focusing solely on current optimal results. Consequently, beam search tends to ignore some partial sequences which might lead to better future outcomes, especially if its beam width is too small. This deficiency is called the *myopic bias* (He et al., 2017). Recently, He et al. (2017) and Li et al. (2017) proposed methods that take account of future BLEU of partial sequences as the future reward during beam search, to alleviate the myopic bias in Neural Machine Translation (NMT). Their experiments show that such methods are capable of improving the BLEU scores of generated translations.

However, several studies point out that BLEU is weakly correlated with human judgments in response generation tasks (Liu et al., 2016; Mou et al., 2016). Unlike in machine translation, where the semantic

[*] Contribution dring internship at Tricorn Technology.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3608–3618
Santa Fe, New Mexico, USA, August 20-26, 2018.

	distinct-1	distinct-2	log-probability	relevance
beam width = 10	0.2831	0.4277	-9.2474	0.7500
beam width = 50 (top10)	**0.4151**	**0.5536**	**-7.4490**	**1.0934**

Table 1: Evaluation results of beam search with width 10 and 50

distribution of appropriate translations given a source sentence is narrow, in response generation tasks, the semantic information of possible responses for one query can be highly diverse. Therefore it is inappropriate to use BLEU, which only takes responses in the training dataset as ground truth, as future reward to solve the myopic bias on response generation. Otherwise, the diversity of generated results might be suppressed.

In this paper, we introduce a new perspective for reducing the myopic bias in response generation. In NRG, the degree of myopic bias for beam search is negatively correlated with its beam width: a greedy search (beam width = 1) myopically stores only the top candidate at each time step; while a larger-width beam search is capable of storing more candidates with potentially higher future probability. Therefore, in this work, for a partial response generated from a small-width beam search, we define its future reward as if it will present in the top responses generated from a beam search with a larger width in future time steps. This presence indicates the potential probability one partial response's successors can reach in the future time steps, therefore taking it as future reward captures the future probability of one generated partial sequence from beam search.

Furthermore, given a query, we design a simple but effective neural network to estimate the future reward for a partial response. Based on this prediction, we re-rank the generated partial responses at each time step, so that we encourage beam search to consider the future probability information of each partial response and generate final results similar to those from a larger width beam search, without proportionally increasing the time cost.

2 Beam Width Analysis

2.1 Beam Search Overview

Given a query $\mathbf{x}$, a K-width beam search stores K candidates denoted as $C_t^K = \{\mathbf{y}_t^k |\ k \in [1, K]\}$ at time step t. At next time step $t+1$, it expands each candidate by words w from vocabulary V. The size of vocabulary V is denoted as $|V|$, so that we have in total $K \times |V|$ potential candidates written as $\{[\mathbf{y}_t^k, w] \mid k \in [1, K], w \in V\}$, with corresponding scores:

$$\text{score}(\mathbf{y}_t^k, w|\mathbf{x}) = \text{score}(\mathbf{y}_t^k|\mathbf{x}) + \log p(w|\mathbf{x}, \mathbf{y}_t^k)\ , k \in [1, K], w \in V \tag{1}$$

The top-K potential candidates are then selected as the candidates in time step $t+1$.

2.2 Influence of Beam Width on Generated Responses

The size of beam width affects results returned from a beam search significantly. Let us assume that there are two beam searches, one with a small beam width K_s and the other with a large beam width K_l. At each time step t, we have two sets of candidates $C_t^{K_s} = \{\mathbf{s}_t^{k_s} |\ k_s \in [1, K_s]\}$ and $C_t^{K_l} = \{\mathbf{l}_t^{k_l} |\ k_l \in [1, K_l]\}$ from corresponding two beam searches. The top-K_s results from the K_l-width beam search tend to benefit from the following properties:

- Higher Probability: For a $\mathbf{l}_t^{k_l}$ ranked lower than K_s in $C_t^{K_l}$, its successors might be ranked above top-K_s in the future time steps, as long as one of its future transition probabilities $p(w|\mathbf{l}_t^{k_l}, \mathbf{x})$ is high enough. Such responses, unfortunately can not be retrieved by the small-width beam search due to its limited beam width. Consequently, for those top-K_s responses from a large-width beam search, their probabilities are more likely to be higher in average. Since in Seq2Seq model, the ideal output given a query x is the response y with the maximum $p(y|x)$, a response with higher probability indicates that it is closer to the optimal result.

- Higher Diversity: In the small-width beam search, many responses in $C_{t+1}^{K_s}$ are generated from a same $\mathbf{s}_t^{k_s}$, because of a dominate score of $\mathbf{s}_t^{k_s}$ compared to other candidates. By contrast, in the large-width beam search, we have a higher chance to observe more top-K_s of $C_{t+1}^{K_l}$ generated from different $\mathbf{l}_t^{k_l}$, since there exist more potential candidates $\mathbf{l}_t^{k_l}$ with high future probabilities $p(w|\mathbf{l}_t^{k_l}, \mathbf{x})$. Therefore the larger beam width leads to a higher diversity in generated sequences generally.

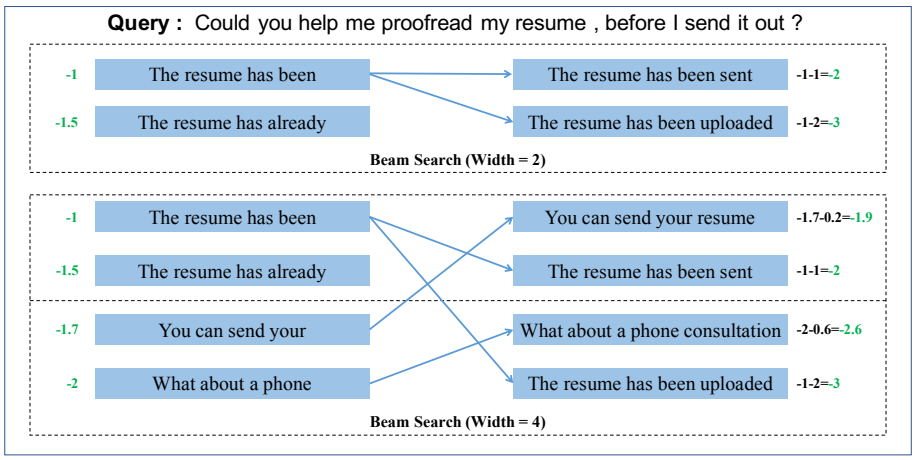

Figure 1: An example of partial responses generated from beam searches with width 2 and 4. The values in green are the corresponding log-probabilities of partial responses.

Figure 1 shows an example where larger beam width helps to generate more diversified results with higher probabilities. Benefited from the more stored candidates, the top-2 generated results from beam search (width =4) are more diversified and have higher probabilities.

To analyze the impact of the beam width on generated responses quantitatively, we employ evaluation methods including distinct, log probability and human evaluation to evaluate the top-10 responses generated by a Seq2Seq model with beam width 10 and 50 respectively. Distinct defined in Li et al. (2016) captures the level of diversity within the generated responses, log-probability is the probability computed by Seq2Seq model during inference after logarithm transformation. The details of above metrics and model training are further described in Section 4.4. The evaluation results in Table 1 show that the top-10 responses from beam width = 50 have higher diversities and probabilities, and are ranked higher by annotators, which is consistent with our hypothesis of beam width.

3 Prospective-Performance Network for NRG

Although a large-width beam search generates responses with higher probability and diversity, increasing beam width proportionally slows down the inference process. Therefore, to retrieve better responses and meanwhile maintain the inference efficiency, we propose Prospective-Performance Network to estimate the future rewards of partial responses in the inference procedure of NRG. The estimated future rewards is then incorporated in the small-width beam search to simulate the performance of a large-width beam search.

3.1 Future Reward

Given a K_l-width beam search we want to simulate, at time step t it generates a set of partial responses $C_t^{K_l} = \{\mathbf{y}_t^{k_l}, k_l \in [1, K_l]\}$. For one partial response $\mathbf{y}_t$, its future reward with regard to beam width of K_l is defined as

$$v(\mathbf{y}_t|\mathbf{x}, K, K_l) = \begin{cases} 1 & \text{if } \mathbf{y}_t \text{ in the top-}K \text{ responses (truncated at } t) \text{ from } C_{t+n}^{K_l} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

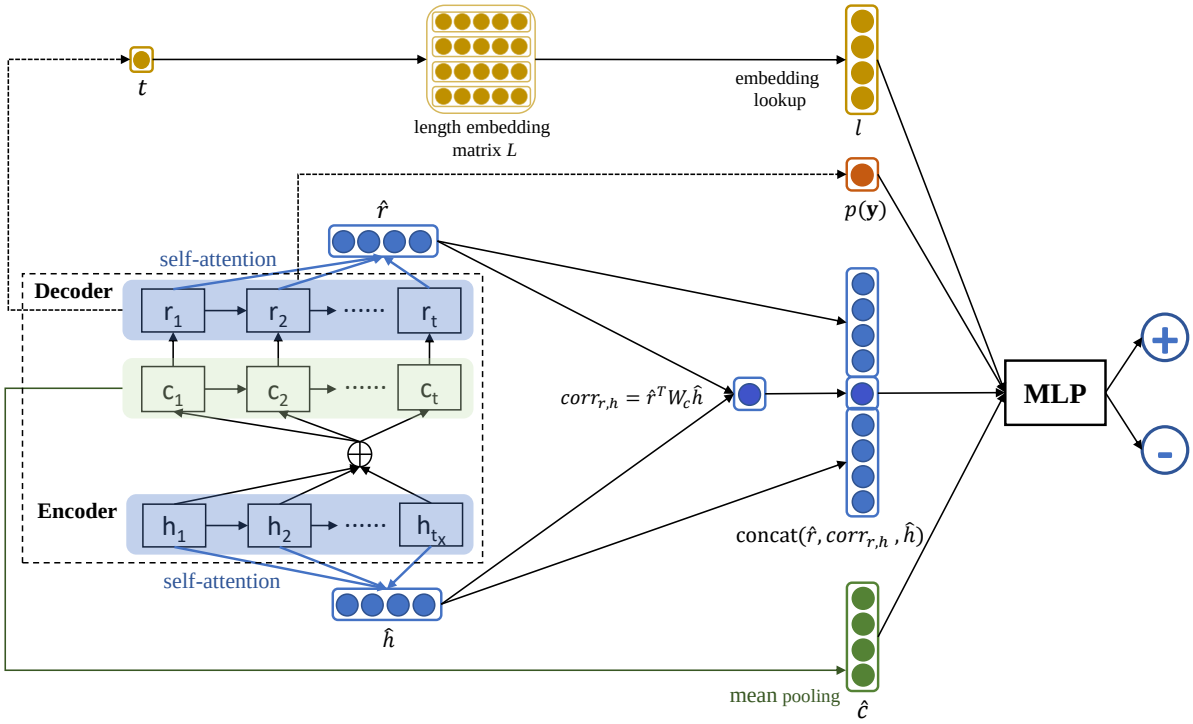

Figure 2: Structure of Prospective-Performance Network

In other words, if a partial response presents in the top-K responses from the given K_l-width beam search ($K < K_l$) at future time steps, we assign it with a positive future reward. The lookahead factor n indicates the prospective level of system.

For the example in Figure 1 where K=2, K_l=4, t=4 and n=1, at time step 4, partial responses "You can send your" and "The resume has been" are attributed with future rewards of 1, because their successors "You can send your resume" and "The resume has been sent" are the top-2 results at the next time step; while "The resume has already" and "What about a phone" are attributed with future rewards of 0. We expect that after reranking the results using future rewards, "You can send your" and "The resume has been" can be ranked at top-2 at time step 4, so that their successors, which have a higher probability at time step 5, can be retrieved through a beam search with beam width of only 2.

3.2 Prospective-Performance Network Structure

In practice, given one partial response and its query, its future reward is unknown without results from a large-width beam search as references. Therefore, we propose Prospective Performance Network (PPN) as a future reward estimator. To fully exploit the information from Seq2Seq encoder-decoder framework in decoding process, PPN is designed with the following four components:

- *Semantic Component:* The Semantic Component captures the semantic information of queries and partial responses. Firstly, to extract most semantics from queries and partial responses, it adopts self-attention mechanism to project encoder hidden states $[h_1, h_2, \ldots, h_{T_x}]$ and decoder hidden states $[r_1, r_2, \ldots, r_t]$ into $\hat{h}$ and $\hat{r}$, specifically:

$$
\begin{aligned}
u_i^h &= tanh(W_a^h h_i + b_a^h) \\
a_i^h &= \frac{\exp((u_i^h)^T u_w^h)}{\sum_{i=1}^{T_x} \exp((u_i^h)^T u_w^h)} \\
\hat{h} &= \sum_{i=1}^{T_x} a_i^h h_i
\end{aligned}
\qquad \text{and} \qquad
\begin{aligned}
u_i^r &= tanh(W_a^r h_i + b_a^r) \\
a_i^r &= \frac{\exp((u_i^r)^T u_w^r)}{\sum_{i=1}^{t} \exp((u_i^r)^T u_w^r)} \\
\hat{r} &= \sum_{i=1}^{t} a_i^r r_i
\end{aligned}
\tag{3}
$$

where W_a^h ,W_a^r, b_a^h, b_a^r, u_w^h and u_w^r are the self-attention parameters. In addition, bilinear transformation is used to further catch the correlation between $\hat{h}$ and $\hat{r}$, such that $corr_{r,h} = \hat{r}^T W_c \hat{h}$. Then the Semantic Component concatenates $\hat{h}$, $corr_{r,h}$ and $\hat{r}$ as a semantic information representation s.

- *Attention Component:* In the Attention Component, mean pooling is used to transfer the context $[c_1, c_2, \ldots, c_t]$ into context representation $\hat{c} = \frac{1}{t}\sum_{i=1}^{t} c_i$. This representation extracts the attention context from the Seq2Seq model.
- *Length Component:* In general, the information provided by short and long partial responses is significantly different from each other. Thus the Length Component is created to summarize the length information of partial responses, it transfers the response length into a length vector l by a length embedding matrix L, different for each time step.
- *Probability Component:* The probability of one partial response $\mathbf{y}$ largely determines the generative probabilities of its successors, therefore Probability Component is employed to extract the current probability $p(\mathbf{y})$ of each input partial response. It adds the log-probability scores for the current partial response, as an singe floating point number.

Finally, PPN concatenates all representations s, $\hat{c}$, l and $p(\mathbf{y})$ as the input of a multilayer perceptron, to estimate the future reward of a partial response $\mathbf{y}$. The whole procedure can be formulated as follows:

$$u = [l,\ p(\mathbf{y}),\ s,\ \hat{c}] \tag{4}$$

$$\hat{v}(\mathbf{y}|\mathbf{x}, K, K_l) = \sigma(W_{mlp}u + b_{mlp}) \tag{5}$$

The estimated future reward $\hat{v}(\mathbf{y}|\mathbf{x}, K, K_l)$ is used as part of the ranking scores in beam search (see details in Subsection 3.4).

3.3 Training Data Generation

Since we aim to generate the top-K_s responses from K_l-width beam search using a smaller search space of K_s, PPN is trained using samples generated from the K_l-width beam search, so that it can be used to estimate the partial responses' future rewards with regard to beam width of K_l. As mentioned in section 2.2, top-K_s responses from K_l-width beam search benefit from properties of higher probability and diversity. Therefore K here is set as K_s, so that a partial response with positive future reward is also associated with above desirable properties. The process of training data generation for PPN is shown in Algorithm 1.

Algorithm 1 Generate PNN training data

Input Small beam width K_s, large beam width K_l, maximum search depth L, candidates in every time step in large beam search $C_t = \{\mathbf{y}_t^1, \mathbf{y}_t^2, \cdots, \mathbf{y}_t^{K_l}\}$, lookahead factor n.

1: **Initialization:** Set $Pset = \emptyset$ as positive samples set, Set $Nset = \emptyset$ as negative samples set, $C_{pre} = \emptyset$ as predecessors set, $t = 0$ as initial time step.
2: **repeat**
3: $\quad t = t + 1$
4: $\quad C_{pre} \leftarrow \{\mathbf{y}[1:t] \mid \mathbf{y} \in C_{t+n}\}$
5: $\quad Pset \leftarrow C_t \cap C_{pre}[1 : K_s]$
6: $\quad Nset \leftarrow C_t \cap C_{pre}[K_s : K_l]$
7: **until** $t + n = L$
8: **Output:** $Pset, Nset$

After training data generation, samples from positive samples set are labeled as 1, while those from negative samples set are labeled as 0. The sum of cross-entropy loss is taken as the loss function to optimize PPN.

3.4 Inference using PPN

The output of PPN provides information related to the future performance of a partial response in the K_l-width beam search, therefore integrating it into the decoding step of a K_s-width beam search helps to alleviate the myopic bias.

For a partial response generated from a K_s-width beam search, its generative probability $P(\mathbf{y}|\mathbf{x})$ is combined with its future reward $\hat{v}(\mathbf{y}|\mathbf{x}, K_s, K_l)$ estimated by PPN. For efficiency, we only compute future rewards on K_l candidates with the highest probabilities at each time step. Among these K_l candidates, those top-K_s responses with the highest combined scores:

$$score(\mathbf{y}|\mathbf{x}) = \log P(\mathbf{y}|\mathbf{x}) + \alpha \times \log \hat{v}(\mathbf{y}|\mathbf{x}, K_s, K_l) \tag{6}$$

are chosen as the candidates after re-ranking. The detailed inference process is shown in Algorithm 2.

Algorithm 2 Beam search with PPN

Input Input query $\mathbf{x}$, Seq2Seq model $P(\mathbf{y}|\mathbf{x})$, vocabulary V, small beam width K_s, large beam width K_l, PNN model $\hat{v}(\mathbf{y}|\mathbf{x}, K_s, K_l)$, maximum search depth L, hyperparameter α.

Initialization: Set $S = \emptyset$ as output set, $C = \emptyset$ as candidate sets, $t = 0$ as time step.
repeat
  $t = t + 1$
  $C_{expand} \leftarrow \{\mathbf{y}_i + [w] \mid \mathbf{y}_i \in C, w \in V\}$
  $C_{K_l} \leftarrow \{\text{top } K_l \text{ candidates that maximize } \log P(\mathbf{y}|\mathbf{x}) \mid \mathbf{y} \in C_{expand}\}$
  $C \leftarrow \{\text{top } (K_s - |S|) \text{ candidates that maximize } \log P(\mathbf{y}|\mathbf{x}) + \alpha \times \log \hat{v}(\mathbf{y}|\mathbf{x}, K_s, K_l) | \mathbf{y} \in C_{K_l}\}$
  $C_{complete} \leftarrow \{\mathbf{y} \mid \mathbf{y} \in C, \mathbf{y}[-1] = EOS\}$
  $C \leftarrow C \backslash C_{complete}$
  $S \leftarrow S \cup C_{complete}$
until $|S| = K_s$ *or* $t = L$
Output: $\mathbf{y} = argmax_{\mathbf{y} \in S \cup C} \log P(\mathbf{y}|\mathbf{x})$

3.5 Time Analysis

In the decoding step of Seq2Seq, its time complexity is dominated by the vocabulary size $|V|$, because of the large-scale computation in the operation of probability distribution projection; while the time complexity of implementing PPN is much smaller than that of decoding, with the help of the simple structure of PPN. Therefore, the time complexity of incorporating PPN in K_s-width beam search is on the same scale as that of the vanilla K_s-width beam search, which means incorporating PPN into beam search preserves the inference efficiency.

4 Experiment

4.1 Dataset

To evaluate our approach, we conduct experiment on a Chinese SNS corpus, which contains single-turn dialogue sessions crawled from a Chinese social network service (SNS)[1]. The preprocessing strategy applied on this dataset is following that of Wu et al. (2017). To further evaluate the effect of our method, we also implement experiment on the large-scaled OpenSubtitles Dataset[2] (Lison and Tiedemann, 2016). After that, we obtain approximately 3,000,000 and 50,000 query-response pairs from the Chinese SNS dataset, 15,000,000 and 50,000 query-response pairs from the OpenSubtitles datasets for training and validating respectively.

[1]The Chinese SNS corpus is confidential in the working organization of the authors.
[2]http://www.opensubtitles.org/

4.2 Hyperparameters

4.2.1 Seq2Seq Model

The parameters settings of the two Seq2Seq models trained using the Chinese SNS and OpenSubtitles datasets are mostly the same: the word embedding size and attention size are both set as 512, both the encoder and decoder are composed of single Long-Short-Term-Memory (LSTM) of hidden size = 512. Seq2Seq models are optimized using Adam with learning rate 0.001 and trained for 10 epochs with batch size 512. The vocabulary sizes for Chinese SNS and OpenSubtitles dataset are set to 50,000 and 40,000 respectively, and all the out-of-vocabulary words are replaced with the *"<UNK>"* token. The maximum sentence lengths at inference step are set to 15 and 30 for the Chinese SNS and OpenSubtitles dataset respectively.

4.2.2 PPN Model

The parameters of Seq2Seq structure in PPN models are retained from pre-trained Seq2Seq models and set as untrainable. In this experiment, we set K_s as 10 and K_l as 50, and α in the inference step is set as 0.5. By setting the beam width as 50, and taking 1 million randomly sampled queries from the training samples of the two datasets as inputs, we obtain a group of responses from Seq2Seq outputs, which are then used to generate the training samples following Algorithm 1. In total 12,000,000 (11,000,000) PPN training samples and 10,000 (10,000) validation samples are generated from the Chinese SNS (OpenSubtitles) dataset, with positive/negative ratio of roughly 1. Adam with learning rate 0.001 is used to optimize the PPN model, the batch size is set as 256 and the models are trained for 3 epochs. The self-attention size and position-embedding size in PPN are set as 256 and 128 respectively.

In addition, the lookahead factor n is set as 1 for performance reason. In our previous experiment, the performance of PPN when n=2 is slightly lower than the case when n=1, and drops significantly when n$>$2.

4.3 Baselines

The following models are used as the comparisons with our proposed PPN:

- *Basic Seq2Seq model* with encoder-decoder structure and vanilla beam searches (Luong et al., 2015) is constructed as the standard baseline.
- *Value Network (VN)* introduced in (He et al., 2017) takes the $\hat{v}(\mathbf{y}|\mathbf{x}, K, K_l)$ defined in Section 3.1 as the future reward instead of future BLEU proposed in the original paper. VN and PPN are trained using the same samples and adopted the same loss function.

  In VN, due to its semantic matching module and context coverage module, it contains two more fully connected layers than PPN. Therefore under the same hyperparameter scale, VN is slower at inferencing step than PPN because of its more complicated network structure.
- *Maximum Mutual Information (MMI)* (Li et al., 2016) is a popular diversity-promoting method which takes maximum mutual information as the objective function. Since the PPN is expected to promote the diversity of generated responses, MMI is included to compare with it on diversity-promoting.

  The MMI variant used in our experiment is the MMI-antiLM.

The training and inference process for all models in our experiments are carried out under the same computational environment: a single Nvidia K80 GPU.

4.4 Evaluation Metrics

The proposed PPN, together with other baselines, are automatically evaluated in terms of the **similarity** toward large-width beam search, **diversity** and **efficiency**.

- **Similarity**: We measure the successfulness of one model with a small beam width K_s (denoted as *model-bw-K_s*) on approximating a standard large-width K_l beam search (*Seq2Seq-bw-K_l*) using

	Similarity		Diversity		Efficiency
	Coverage	**Log-prob**	**Distinct-1**	**Distinct-2**	**Time Cost**
Seq2Seq-bw50-top10	-	-7.4490	0.4151	0.5536	1.9496
Seq2Seq-bw10	0.4009	-9.2474	0.2831	0.4272	**0.4399**
MMI-bw10	0.4166	-8.9286	0.2858	0.4323	0.6233
VN-bw10	0.5030	-8.3672	0.3183	0.4715	0.6027
PPN-bw10	**0.5515**	**-8.0738**	**0.3555**	**0.5107**	0.5667

Table 2: Automatic evaluation results on the Chinese SNS dataset.

	Similarity		Diversity		Efficiency
	Coverage	**Log-prob**	**Distinct-1**	**Distinct-2**	**Time Cost**
Seq2Seq-bw50-top10	-	-5.0762	0.4338	0.4585	2.2996
Seq2Seq-bw10	0.4103	-6.9216	0.3341	0.4295	**0.4131**
MMI-bw10	0.4363	-6.8118	0.3240	0.4184	0.6089
VN-bw10	0.4880	-6.4787	0.3550	0.4449	0.7058
PPN-bw10	**0.5980**	**-5.8012**	**0.3711**	**0.4613**	0.5827

Table 3: Automatic evaluation results on the OpenSubtitles dataset.

coverage and **log-probability**. Coverage of *model-bw-*K_s is defined as the ratio of its generated responses presented in top-K_s responses from *Seq2Seq-bw-*K_l. A model with a higher coverage indicates it generates more responses ranked top by the compared large beam search. Log-probability is the mean probability of generated top-K_s responses after logarithm transformation ($log(p(\mathbf{y}|x))$) during inference. Generally speaking, a model with a closer log-probability compared to *Seq2Seq-bw-*K_l is more desirable.

- **Diversity**: Diversity of generated responses from each model is measured by the **distinct-1** and **distinct-2**, which are calculated respectively by the number of distinct unigrams and bigrams in the set of generated responses divided by the total number of generated tokens (Li et al., 2016).
- **Efficiency**: To evaluate the inference efficiency of each model, we also compare the **time cost** in seconds on generating the responses set given one query.

Besides automatic evaluations, the qualities of generated responses from each model are manually evaluated in terms of **relevance** and **grammar correctness**. In total, 300 query-response pairs generated from each model trained using the Chinese SNS dataset are randomly sampled. For each query-response pair, 3 annotators are invited to evaluate its grammatical correctness as 0 (grammatically incorrect) or 1 (grammatically correct), and relevance as 0 (irrelevant), 1 (acceptable) or 2 (great).

5 Results

5.1 Automatic Evaluation

Table 2 and 3 show the automatic evaluation results on the Chinese SNS and OpenSubtitles datasets. Here *Seq2Seq-bw10* and *Seq2Seq-bw50-top10* stand for two basic Seq2Seq models with beam width of 10 and 50 respectively. In addition, only top-10 ranked responses in *Seq2Seq-bw50-top10* are taken into account when evaluating. *MMI-bw10*, *VN-bw10* and *PPN-bw10* refer to the MMI, VN and PPN model respectively, and their beam widths are also set as 10.

It can be observed that *PPN-bw10* obtains the highest coverage and closest log-probability compared to *Seq2Seq-bw50-top10* on both datasets, which indicates that our proposed PPN is capable of simulating a larger-width beam search. In addition, with the help of more explicit method to capture semantics along with the length and probability information from corresponding components, the proposed PPN model is more effective on capturing the future reward than the VN, reflected by the higher coverage of *PPN-bw10* compared to *VN-bw10*.

	Grammar	Relevance
Seq2Seq-bw50-top10	0.8709	1.0934
Seq2Seq-bw10	0.6236	0.7500
MMI-bw10	0.6280	0.7680
VN-bw10	0.7135	0.8567
PPN-bw10	**0.8022**	**0.9505**

Table 4: Human evaluation results on the Chinese SNS dataset. The scores are means over 300 samples.

In terms of diversity, PPN also significantly improves the distinct-1 and distinct-2 compared to the vanilla beam search. It is worth noting that the distinct-1 and distinct-2 of *PPN-bw10* are higher than those of *MMI-bw10*. The PPN exploits the nature of high-diversity in large-width beam search to promote the diversity of responses, and the experiment results show some evidence that such method is more effective than MMI on diversity promotion.

As expected, the time cost of *Seq2Seq-bw50-top10* is approximately 5 times longer than that of *Seq2Seq-bw10*, while *PPN-bw10* only raises the inference time by around 10% meanwhile achieves a significant improvement on quality of responses compared to *Seq2Seq-bw10*. It proves the feasibility of our proposed PPN in practice.

The p-values of PPN against three baseline models on coverage, log-probability, distincts are all smaller than 5e-6, which indicates that the improvements of performance from PPN are significant.

5.2 Human Evaluation

The human evaluation result is shown in Table 4. The responses from *Seq2Seq-bw50-top10* and *PPN-bw10* are annotated with the highest and the second highest grammatical correctness and relevance. The result further reinforces our deduction that a larger beam width improves the quality of generated responses, and the PPN is capable of approximating a large beam width search. In addition, we test the consistency of the human evaluation using Fleiss' kappa (Fleiss and Cohen, 1973). For grammatical correctness and relevance, their Fleiss' kappa on all models are around 0.6 and 0.4 respectively, which can be both considered as "moderate agreement". The p-values of PPN against three baseline models on grammar and relevance scores are all below 0.05.

5.3 Further Analysis

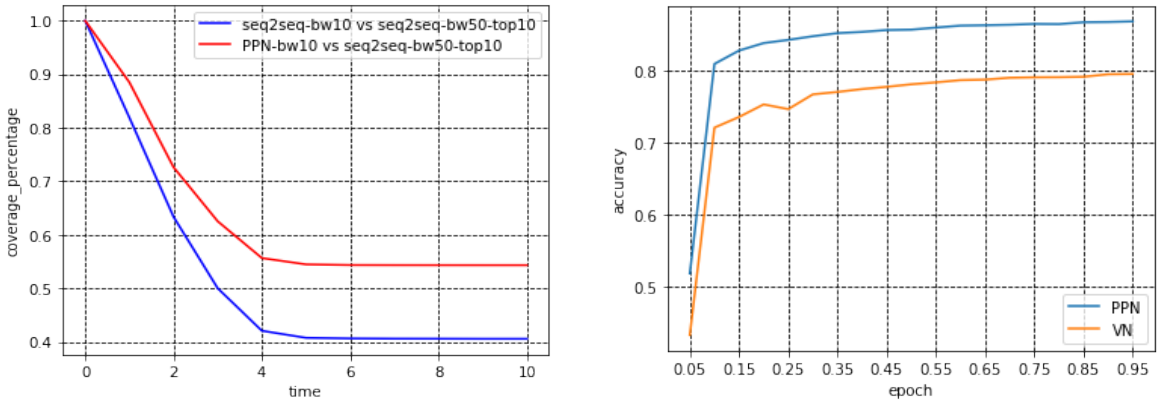

Figure 3: Left: Coverage evolution between PPN and vanilla Seq2Seq over time step. Right: Accuracy evolution on validation set during training of PPN and VN, Chinese SNS dataset.

To analyze the performance of PPN in detail, two plots regarded to the comparisons of coverage and prediction accuracy between PPN and other baseline models are shown in Figure 3. It can be observed in the left plot of Figure 3 that the coverages of *PPN-bw10* are always higher than those of *Seq2Seq-bw10*,

which indicates that PPN generates more top responses from large-width beam search consistently over time. Moreover, the right plot in Figure 3 shows the accuracy of PPN on classifying the samples in validation dataset throughout the training of PPN and VN. The higher accuracies of PPN further prove that PPN is more effective on estimating the future reward than VN.

6 Related Work

Inspired by the success of the Seq2Seq framework on NMT (Cho et al., 2014; Sutskever et al., 2014b; Bahdanau et al., 2015), this framework has been adopted for response generation (Vinyals and Le, 2015; Shang et al., 2015) and is proved to be effective on generating responses based on given queries (Sordoni et al., 2015; Li et al., 2016; Serban et al., 2016; Xu et al., 2017).

Most NMT and NRG systems generate outputs using the beam search algorithm, which unfortunately suffers from the myopic bias. To solve the myopic bias, He et al. (2017) and Li et al. (2017) both propose method to take the future BLEU of decoder partial outputs into account in beam search. Another study indirectly related to our work is Wiseman and Rush (2016), it treats the target sequences in training set as the gold sequences, and directly training the beam search to select word instead of probability. Although these methods are proved to be effective on NMT, it might be inappropriate to directly apply them on NRG, since appropriate responses for one query are highly diverse in terms of semantics. By contrast, the proposed method exploits the nature of beam search width to alleviate the myopic bias.

7 Conclusions

In this paper, we have described our attempt on reducing the myopia in beam search for NRG. In detail, we have: 1) verified the effectiveness of increasing the beam width on relieving the myopic bias; 2) proposed a future reward to alleviate the myopic bias of canonical Seq2Seq-based NRG model, and specially designed a perspective-performance network to quantify the future reward reasonably; 3) presented a new decoding strategy on the basis of the perspective-performance network to generate top-ranked responses given by a large-width beam search. The experiment results show the effectiveness and efficiency of our method. The proposed method is especially useful on online conversational agents, where the speed of response generation is of great importance. In the future, to better estimate the future reward, we will explore different model structures and new training data generation strategies.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tieyan Liu. 2017. Decoding with value networks for neural machine translation. In *Advances in Neural Information Processing Systems 30*, pages 177–186.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Learning to decode for future success. *arXiv preprint arXiv:1701.06549*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3349–3358.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1577–1586.

Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From eliza to xiaoice: Challenges and opportunities with social chatbots. *arXiv preprint arXiv:1801.01957*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL-HLT*, pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014a. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014b. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 496–505.

Zhen Xu, Bingquan Liu, Baoxun Wang, SUN Chengjie, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 617–626.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2017. Emotional chatting machine: emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*.