Adaboost with Auto-Evaluation for Conversational Models

Juncen Li[1], **Ping Luo**[2,3], **Ganbin Zhou**[2,3], **Fen Lin**[1], **Cheng Niu**[1]
[1] WeChat Search Application Department, Tencent, China
[2] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
{juncenli,felicialin,niucheng}@tencent.com, luop@ict.ac.cn

Abstract

We propose a boosting method for conversational models to generate more human-like dialogs. In our method, we consider the existing conversational models as weak generators and apply the Adaboost to update those models. However, conventional Adaboost cannot be directly applied on conversational models, since conventional Adaboost cannot adaptively adjust the weight on the instance for subsequent learning. This results from the conventional methods based on the simple comparison between the true output $\boldsymbol{y}$ (to an input $\boldsymbol{x}$) and its corresponding predicted output $\boldsymbol{y}'$, cannot effectively evaluate the learning performance on $\boldsymbol{x}$. To address this issue, we develop the Adaboost with Auto-Evaluation (called *AwE*). In AwE, an auto-evaluator is proposed to evaluate the predicted results, which makes Adaboost applicable to conversational models. Furthermore, we present the theoretical analysis that the training error drops exponentially fast only if certain assumption over the proposed auto-evaluator holds. Finally, we empirically show that AwE visibly boosts the performance of existing single conversational models and also outperforms the other ensemble methods for conversational models.

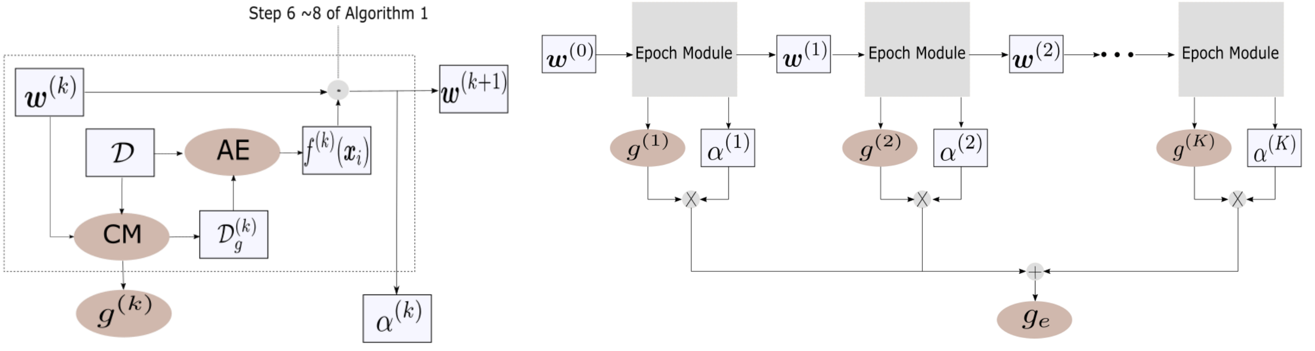
1 Introduction

In the light of these advances, recent years witness an increasing research interests in applying the end-to-end neural methods to conversational models [Shang *et al.*, 2015; Sordoni *et al.*, 2015; Serban *et al.*, 2016; 2017]. These previous studies aim to improve conversational models. First, more sophisticated neural architectures are proposed, such as attention mechanism [Bahdanau *et al.*, 2015] and hierarchical neural network [Serban *et al.*, 2016], are proposed to depict the mapping between input and output more carefully. Second, some other learning methods are applied to updated conversational models. For example, reinforcement learning [Li *et al.*, 2016], adversarial learning [Li *et al.*, 2017] and adaptation methods [Li *et al.*, 2018] are used to modify conversational models. However, two problems of conversational models are still not solved completely. Firstly, existing conversational models may generate responses which are not suitable to input posts in any scenarios. Secondly, these models tend to generate several highly generic responses, for example, “I don’t know”.

With the above two problems, the performance of existing conversational models are unsatisfactory. Here, they can be seen as weak generators. To update weak generators, we extend the ensemble method Adaboost for conversational models. Adaboost [Freund and Schapire, 1997] is an effective ensemble framework to improve the performance of supervised learning (originally for single-variable output). The key to Adaboost is its adaptive feature where subsequent weak learners are tweaked in favor of those instances on which previous weak-performance learners . Thus, it needs an automatic method to evaluate the learning performance on each instance.

However, for conversational models the automatic evaluation is not trivial. Consider the dialog generation task. Let $\boldsymbol{y}$ and $\boldsymbol{y}'$ be the true and predicted response to an input post $\boldsymbol{x}$. To be a good response to $\boldsymbol{x}$, $\boldsymbol{y}'$ is not required to be exactly equal to $\boldsymbol{y}$, but is only required to be grammatically correct and semantically relevant to input $\boldsymbol{x}$. Thus, for conversational models, the evaluation of the predicted responses is not trivial, therefore conventional Adaboost cannot be applied on conversational models directly.

In this paper, we study the extension of Adaboost for conversational models. To this end, we develop the framework of Adaboost with Auto-Evaluation (called *AwE*). In AwE, an auto-evaluator is proposed to evaluate predicted results. Specifically, motivated by the discriminator in Generative Adversarial Nets [Goodfellow *et al.*, 2014], a classifier is trained to distinguish the true and predicted responses for the input posts. Then, we believe that the conversational model performs well on $\boldsymbol{x}$ only if the classifier makes a *false-positive* error on $\boldsymbol{x}$. Thus, this classifier can be used as an auto-evaluator to adjust the weights on instance for the next-round learning in Adaboost. Furthermore, we theoretically analyze the training error of AwE. The training error drops exponentially fast only if certain assumption over the proposed auto-evaluator holds. Finally, we do some empirical experiments to evaluate our method. We demonstrate that AwE visibly



(a) Epoch module

(b) Whole process

Figure 1: Illustration of Adaboost with Auto-Evaluation: (a) Epoch module of AwE, where AE is the automatic-evaluator, CM is the conversational model, $\boldsymbol{w}^{(k)}$ is the weight set of instances, $g^{(k)}$ is the trained CM, $\mathcal{D}$ is the input dataset, $\mathcal{D}_g^{(k)}$ is the generated dataset predicted by the trained CM, $\alpha^{(k)}$ is the weight of $g^{(k)}$ which is related to the error of $g^{(k)}$, $f^{(k)}(\boldsymbol{x}_i)$ represents the performance of $g^{(k)}$ on $\boldsymbol{x}_i$, here $f^{(k)}(\boldsymbol{x}_i) = 1$ means $g^{(k)}$ performs well on $\boldsymbol{x}_i$, and vice versa. (b) Whole process of AwE, where g_e is the ensemble model.

boosts the performance of single model and also outperforms the other ensemble methods for conversational models.

2 Related Work

In the light of end-to-end neural system of statistical machine translation (SMT) [Yin *et al.*, 2016; Cho *et al.*, 2014; Bahdanau *et al.*, 2015], researches on neural conversational models have made some progress. Shang et al. [2015] introduced three types of encoding schemes as extensions of attention method. They found that hybrid scheme performs better than the other two schemes in generating responses. Instead of focusing on one-round dialog, Serban et al. [2016] devised a hierarchical neural network. They encoded a sequence of words into an utterance vector and used the utterance vector of previous sentences as context information. Then they [2017] extended the hierarchical neural network by adding a parallel RNN encoder, which encodes the high-level coarse tokens, into the previous framework. Another way to improve conversational models is to solve the high frequency responses problem and increase the response diversity. Jiwei et al. [2016] brought reinforcement learning into the dialog system to overcome the high frequency responses challenge.

Adaboost was proposed by Freund and Schapire [1997]. It was applied to improve performance of word recognition in handwritten documents [Schwenk and Bengio, 1998], face recognition [Sun *et al.*, 2012] . It was also used to update structured learning with explicit evaluation labels [Cortes *et al.*, 2014]. All of those tasks can be automatically evaluated by simple comparison between the true output $\boldsymbol{y}$ (to an input $\boldsymbol{x}$) and its corresponding predicted output $\boldsymbol{y}'$. However, to our best knowledge, Adaboost has not been employed on conversational models.

We propose AwE which employs Adaboost to update conversational models. Our method designs an auto-evaluator inspired by the discriminator of Generative Adversarial Nets (GAN) [Goodfellow *et al.*, 2014]. GAN and its variants, such as LAPGAN [Denton *et al.*, 2015] and DCGAN [Radford *et al.*, 2015], have made great progress on image generation. GAN consists of two components: a generator and a discriminator. The generator generates images which are similar to real images. And the discriminator differentiates real images and fake images which are generated by the generator. For application of GAN on natural language processing (NLP), Lantao et al. [2017] proposed SeqGAN to generate text.

3 Adaboost with Auto-Evaluation

In this section, we introduce Adaboost with Auto-Evaluation (AwE). Firstly, we describe the training process of AwE. As shown in Fig.1, the main training process of AwE is to train several models iteratively. We regard every training epoch as an epoch module and the framework of our method is to combine those modules. Then we propose a Weighted Beam Search method as the inference method for AwE. Finally, we prove an upper bound of the training error when certain assumption on the auto-evaluators holds.

3.1 Training Process

The epoch module is shown in Fig.1(a). It has two inputs. The first input is a dataset $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$, where $\boldsymbol{x}_i$ is input post, $\boldsymbol{y}_i$ is output response, N is the number of input-output pairs in the dataset. The other input is a weight set $\boldsymbol{w}^{(k-1)}$. $\boldsymbol{w}^{(k-1)}$ is leveraged to control the attentions of conversational model (CM) on input-output pairs. A CM mainly focuses on the input-output pairs where the CM of previous epoch module obtains unsatisfactory performance. In addition, we initialize all weights equally, which means every pair is equally important at the beginning.

There are two main models in the epoch module: conversational model (CM) and auto-evaluation (AE). Our method can be easily used on the neural conversational models which are based on sequence-to-sequence format, such as [Cho *et al.*, 2014; Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015; Serban *et al.*, 2016; Shang *et al.*, 2015]. Here, the method in [Cho *et al.*, 2014] is adopted. We can utilize $\boldsymbol{w}^{(k-1)}$ and $\mathcal{D}$ to train CM. Using the trained model, we then generate

Algorithm 1 AUTO-EVALUATION ADABOOST

```
Require:
    Dataset, $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{N}$, where $\boldsymbol{x}_i$ is input, $\boldsymbol{y}_i$ is output, N is the number of input-output pairs
    Conversational model, CM
    Auto-evaluator, AE
    Iteration number, $K$
Ensure: Ensemble model $g_e(\boldsymbol{x}, \boldsymbol{y})$
 1: $\boldsymbol{w}^{(0)} \leftarrow \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$ //$\boldsymbol{w}$ is the set of weights
 2: for $k = 1$ to $K$ do
 3:     $g^{(k)} \leftarrow \mathrm{CM}\left(\mathcal{D}, \boldsymbol{w}^{(k-1)}\right)$
 4:     $e^{(k)} \leftarrow \mathrm{AE}\left(\left\{\mathcal{D}, \mathcal{D}_g^{(k)}\right\}\right)$
        // $\mathcal{D}_g^{(k)}$ is the generated dataset predicted by the trained CM
 5:     $f^{(k)}(\boldsymbol{x}_i) \leftarrow \begin{cases} 1 & , e^{(k)}(\boldsymbol{x}_i, \boldsymbol{y}'_i) = 1 \\ -1 & , e^{(k)}(\boldsymbol{x}_i, \boldsymbol{y}'_i) = -1 \end{cases}, \forall i$
 6:     $\varepsilon^{(k)} = \sum_{i=1}^{N} \boldsymbol{w}_i^{(k-1)} I\left[f^{(k)}(\boldsymbol{x}_i) \neq 1\right]$,
        where $I(a) = \begin{cases} 1 & , \text{if } a \text{ is true} \\ 0 & , \text{if } a \text{ is false} \end{cases}$
 7:     $\alpha^{(k)} \leftarrow \frac{1}{2}\log\left(\frac{1-\varepsilon^{(k)}}{\varepsilon^{(k)}}\right)$
 8:     $\forall i, w_i^{(k)} \leftarrow \frac{1}{Z} w_i^{(k-1)} \exp\left(-\alpha^{(k)} f^{(k)}(\boldsymbol{x}_i)\right)$ .
        //$Z$ is the sum of $w_i^{(k)}$ that are used for normalization
 9: end for
10: return $g_e(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{k=K} \alpha^{(k)} g^{(k)}(\boldsymbol{x}, \boldsymbol{y})$
```

a new dataset with inputs in $\mathcal{D}$, and define this dataset as $\mathcal{D}_g = (\boldsymbol{x}_i, \boldsymbol{y}'_i)_{i=1}^{N}$, where $\boldsymbol{y}'_i$ is the output of $\boldsymbol{x}_i$.

For AE, AwE employs an auto-evaluator to evaluate outputs generated by CM. All evaluation methods which can effectively evaluate results of conversational models can be utilized in our method. Here, we employ a learned evaluator which is motivated by GAN. The AE that we apply uses Recurrent Neural Network (RNN) encoder with Gated Recurrent Unit (GRU) to encode the input and output. Then the last hidden vectors of input and output are combined as a hidden vector of each input-output pair. Finally, we apply Softmax for classification. For the training data of the AE, both $\mathcal{D}$ and $\mathcal{D}_g$ are used. All input-output pairs in $\mathcal{D}$ are labeled by 1 and all the pairs in $\mathcal{D}_g$ are labeled by -1. Then we use the labeled input-output pairs in $\mathcal{D}$ and $\mathcal{D}_g$ to train the AE. The trained auto-evaluator $e^{(k)}$ can help us to judge if the CM performs well on every input $\boldsymbol{x}_i$ in the dataset. If $e^{(k)}(\boldsymbol{x}_i, \boldsymbol{y}'_i) = 1$, then $\boldsymbol{y}'_i$ can be regraded as real output and is suitable to $\boldsymbol{x}_i$, which means the CM performs satisfactorily on $\boldsymbol{x_i}$. Whereas, if $e^{(k)}(\boldsymbol{x}_i, \boldsymbol{y}'_i) = -1$, then $\boldsymbol{y}'_i$ can be regraded as fake output, which means the CM performs unsatisfactorily on $\boldsymbol{x}_i$ and $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ are not learned well by CM. We define a function $f^{(k)}(\boldsymbol{x}_i)$ to represents the performance of CM on $\boldsymbol{x}_i$, where $f^{(k)}(\boldsymbol{x}_i) = 1$ means the CM performs well on $\boldsymbol{x}_i$, and vice versa. Thus the relation of $f^{(k)}(\boldsymbol{x}_i)$ and $e^{(k)}(\boldsymbol{x}_i, \boldsymbol{y}'_i)$ can be represented by the step 5 in Algorithm 1.

After getting $f^{(k)}(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in \mathcal{D}$, we can calculate the weight of the trained CM $\alpha^{(k)}$ and update $\boldsymbol{w}^{(k)}$ according to the rules of step from 6 to 8 in the Algorithm 1. Then we obtain three outputs of the epoch module: new weights set $\boldsymbol{w}^{(k)}$, the trained CM $g^{(k)}$ and $\alpha^{(k)}$.

The whole process of training AwE is shown in Fig.1(b). In the whole process, we combine several epoch modules. $\boldsymbol{w}^{(k)}$ can be regarded as the connector between two epoch modules. $\boldsymbol{w}^{(k)}$ is output by the k-th epoch module and is the input of the $(k+1)$-th epoch module. It stores the information of which pairs are unsatisfactorily learned and guides the training of CM in the $(k+1)$-th epoch module.

Except for $\boldsymbol{w}^{(k)}$, the outputs of epoch modules are required by the ensemble process. We combine $g^{(k)}$ by adding them up according to their own weights $\alpha^{(k)}$, which is shown in step 10 of Algorithm 1. $\alpha^{(k)}$ is inversely proportional to total error of $g^{(k)}$. That means $g^{(k)}$ whose error is lower plays a more important role in the ensemble model.

3.2 Weighted Beam Search for Sequence Generation

In the inference process, we need to generate an output $\boldsymbol{y'} = \left\{y^{(1)}, y^{(2)}, \dots, y^{(L)}\right\}$ (where L is the length of the output) for a given input $\boldsymbol{x}$. Specifically, we select the $\boldsymbol{y'}$ by maximizing probability with g_e:

$$\begin{aligned} \boldsymbol{y'} &= \underset{\boldsymbol{y}}{\operatorname{argmax}}\ g_e(\boldsymbol{x}, \boldsymbol{y}) \\ &= \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{k=1}^{K} \alpha^{(k)} g^{(k)}(\boldsymbol{x}, \boldsymbol{y}) \\ &= \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{k=1}^{K} \alpha^{(k)} p^{(k)}(\boldsymbol{y}|\boldsymbol{x}) \end{aligned} \tag{1}$$

Using the above equation, we must traverse all plausible outputs. This method is time-consuming. To reduce the time-complexity, we propose the Weighted Beam Search method which generates the output word by word. Specially, when generating the first words, we obtain the conditional probability of first words using:

$$p\left(y^{(1)}|\boldsymbol{x}\right) = \sum_{k=1}^{K} \alpha^{(k)} p^{(k)}\left(y^{(1)}|\boldsymbol{x}\right) \tag{2}$$

The first words of outputs can be selected according to $p\left(y^{(1)}|\boldsymbol{x}\right)$. M (beam search size) words with highest $p\left(y^{(1)}|\boldsymbol{x}\right)$ are selected as possible words. In the step of generating sub-sequence $\left\{y^{(1)}, \dots, y^{(n)}\right\}$ $(n > 1)$, Weighted Beam Search firstly uses the added conditional probability as the ensemble conditional probability:

$$\begin{aligned} & p\left(y^{(n)}|\boldsymbol{x}, y^{(1)}, \dots, y^{(n-1)}\right) \\ & = \sum_{k=1}^{K} \alpha^{(k)} p^{(k)}\left(y^{(n)}|\boldsymbol{x}, y^{(1)}, \dots, y^{(n-1)}\right) \end{aligned} \tag{3}$$

Secondly, with $p\left(y^{(1)}, \dots, y^{(n-1)}|\boldsymbol{x}\right)$ from the previous step of generating $\left\{y^{(1)}, \dots, y^{(n-1)}\right\}$, we can get the conditional probability of sub-sequence $p\left(y^{(1)}, \ldots, y^{(n)} | \boldsymbol{x}\right)$ using:

$$\begin{aligned} & p\left(y^{(1)}, \ldots, y^{(n)} | \boldsymbol{x}\right) \\ & = p\left(y^{(n)} | \boldsymbol{x}, y^{(1)}, \ldots, y^{(n-1)}\right) \\ & \cdot p\left(y^{(1)}, \ldots, y^{(n-1)} | \boldsymbol{x}\right) \end{aligned} \tag{4}$$

Then we can select M possible sub-sequences $\left\{y^{(1)}, \ldots, y^{(n)}\right\}$ with highest conditional probability $p\left(y^{(1)}, \ldots, y^{(n)} | \boldsymbol{x}\right)$.

3.3 Analyzing the Training Error of AwE

Here, we use $\boldsymbol{y}_i'$ to represent the output generated by the ensemble model and $e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right)$ to represent the evaluation result of $\boldsymbol{y}_i'$ to $\boldsymbol{x}_i$. If $e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) = 1$, then $\boldsymbol{y}_i'$ is suitable to $\boldsymbol{x}_i$ and vice-versa. Thus, $\frac{1}{N} \sum_{i=1}^{N} I\left[e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) \neq 1\right]$ can represent the training error of AwE. Next, we propose some assumptions on the auto-evaluator and then propose an upper bound to this training error based on this assumption.

Assumption 1.

$$e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) = \operatorname{sgn}\left(\sum_{k=1}^{K} \alpha^{(k)} e^{(k)}\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right)\right) \tag{5}$$

The assumption actually requires the evaluation on $\boldsymbol{y_i'}$ (predicted by the ensemble method) to be the linear sum of the evaluation results from the K individual auto-evaluators. We argue that this assumption is reasonable since each individual auto-evaluator $e^{(k)}$ contributes the final evaluation with the weight $\alpha^{(k)}$. With this assumption we have the following theorem for the upper bound of training error for AwE.

Theorem 1. *The upper error bound of AwE on training data is as follows:*

$$\frac{1}{N} \sum_{i=1}^{N} I\left[e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) \neq 1\right] \leq \exp\left(-2 \sum_{k=1}^{K} {\gamma_k}^2\right) \tag{6}$$

where $\gamma_k = \frac{1}{2} - \varepsilon^{(k)}$.

Proof. First, let

$$f\left(\boldsymbol{x}_i\right) = \operatorname{sgn}\left(\sum_{k=1}^{K} \alpha^{(k)} f^{(k)}\left(\boldsymbol{x}_i\right)\right) \tag{7}$$

Then, based on the theorem on the upper error bound for Adaboost [Freund and Schapire, 1997] we have

$$\frac{1}{N} \sum_{i=1}^{N} I\left[f\left(\boldsymbol{x}_i\right) \neq 1\right] \leq \exp\left(-2 \sum_{k=1}^{K} {\gamma_k}^2\right) \tag{8}$$

Then, to prove Theorem 1, we need to prove

$$\frac{1}{N} \sum_{i=1}^{N} I\left[e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) \neq 1\right] = \frac{1}{N} \sum_{i=1}^{N} I\left[f\left(\boldsymbol{x}_i\right) \neq 1\right] \tag{9}$$

The above Eq. 9 can be proved as follows:

Training	posts	536,639
Training	responses	542,083
Training	pairs	773,315
Validation	posts	20,000
Validation	responses	25,086
Validation	pairs	28,949
Test	posts	1000

Table 1: Some statistics of the dataset

$$\begin{aligned} & \frac{1}{N} \sum_{i=1}^{N} I\left[e\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right) \neq 1\right] \\ & = \frac{1}{N} \sum_{i=1}^{N} I\left[\operatorname{sgn}\left(\sum_{k=1}^{K} \alpha^{(k)} e^{(k)}\left(\boldsymbol{x}_i, \boldsymbol{y}_i'\right)\right) \neq 1\right] && (10) \\ & = \frac{1}{N} \sum_{i=1}^{N} I\left[\operatorname{sgn}\left(\sum_{k=1}^{K} \alpha^{(k)} f^{(k)}\left(\boldsymbol{x}_i\right)\right) \neq 1\right] && (11) \\ & = \frac{1}{N} \sum_{i=1}^{N} I\left[f\left(\boldsymbol{x}_i\right) \neq 1\right] && (12) \end{aligned}$$

where Eq. 10 follows from Assumption 1, Eq. 11 follows from Step 5 in Algorithm 1 and Eq. 12 follows from Eq. 7.

□

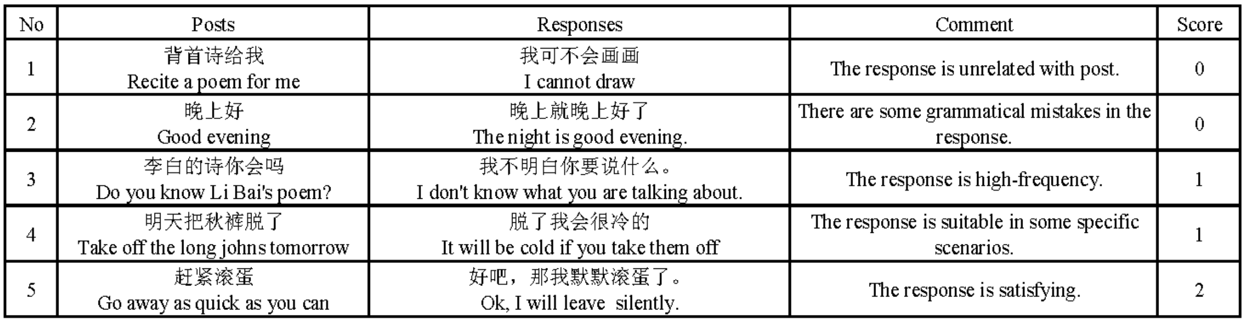
From Theorem 1, we get the conclusion that with the growth of K, the upper error bound of the training data decreases exponentially. In other words, we can increase K to improve the performance of the ensemble method over the training data. However, when K becomes too large, the model may overfit the training data. Thus, finding a suitable value of K is important. In this paper, we select the value of K according to performance of AwE on validation dataset.

4 Experiments

We examine the performance of our method on dialog generation tasks and compare with three benchmarks in experiments. Most of conversational models based on sequence-to-sequence format can be used in our method. For convenience, we choose RNN Encoder-Decoder with GRU [Cho *et al.*, 2014] as the conversational model in our experiments. For fair comparison, all the benchmarks are based on the RNN Encoder-Decoder with GRU.

4.1 Dataset

We collect nearly 14 million post-response pairs from Tencent Weibo . Removing spams and advertisements from that dataset, there are only 803,716 high-quality post-response pairs retained. We randomly divide this data into training, validation and test set with no overlap posts. In other words, we insure that there are no posts appearing in two different sets. Table 1 shows some statistics of the dataset used in this paper. Then we build the dictionary with Chinese characters and high frequency Chinese words. The dictionary consists



No	Posts	Responses	Comment	Score
1	背首诗给我 Recite a poem for me	我可不会画画 I cannot draw	The response is unrelated with post.	0
2	晚上好 Good evening	晚上就晚上好了 The night is good evening.	There are some grammatical mistakes in the response.	0
3	李白的诗你会吗 Do you know Li Bai's poem?	我不明白你要说什么。 I don't know what you are talking about.	The response is high-frequency.	1
4	明天把秋裤脱了 Take off the long johns tomorrow	脱了我会很冷的 It will be cold if you take them off	The response is suitable in some specific scenarios.	1
5	赶紧滚蛋 Go away as quick as you can	好吧，那我默默滚蛋了。 Ok, I will leave silently.	The response is satisfying.	2

Table 2: Human evaluation examples

Methods	BLEU
Seq2Seq	3.85
Maximum	3.92
Bagging	3.86
AwE	**4.27**

Table 3: BLEU of AwE and three benchmarks

Methods	Proportion
Seq2Seq	21.1%
Maximum	27.8%
Bagging	27.1%
AwE	19.8%

Table 4: Proportion of high frequency responses

of 17,395 items and we use these items to split the sentence of post-response pairs.

4.2 Benchmarks

We use three benchmarks in this paper which are all based on RNN Encoder-Decoder with GRU.

- Seq2Seq: This method is RNN Encoder-Decoder with GRU [Cho *et al.*, 2014].
- Maximum: It is an ensemble method which is similar to the ensemble method in [Zhou *et al.*, 2017]. It combines the results of several generators by the way of selecting results with maximum scores.
- Bagging: This ensemble method is widely used in reading comprehension and question answering [Wang *et al.*, 2017a; 2017b; Seo *et al.*, 2017]. It is an ensemble method which combines several different models by averaging scores of those models and choosing the response with highest average score.

4.3 Experimental Details

The experimental details in this paper are as follows:

- We use 1-layer GRU with 512 cells for both the encoder and the decoder.
- We use different word embeddings for the encoder and the decoder as suggested in [Shang *et al.*, 2015]. Both embedding dimensions are set to 128.
- We initialize all parameters with the uniform distribution between -0.1 and 0.1. And We set the minibatch size to 256.
- We use beam search method to do the generation and we set beam size to 10.

5 Results and Analysis

We use two evaluation methods to compare the performance between our method and three benchmarks: BLEU and human evaluation method [Shang *et al.*, 2015].

5.1 Human Evaluation

We use human evaluation method to evaluate our models and three benchmarks referring to [Shang *et al.*, 2015]. To prevent human annotation bias, we mix generated results of all models up and let four labelers score the same result set independently. All the labelers come from a professional company and have at least one-year experience of labeling dialog system. The human evaluation examples are shown in Table 2. The score ranges from 0 to 2 indicating bad, normal and good respectively.

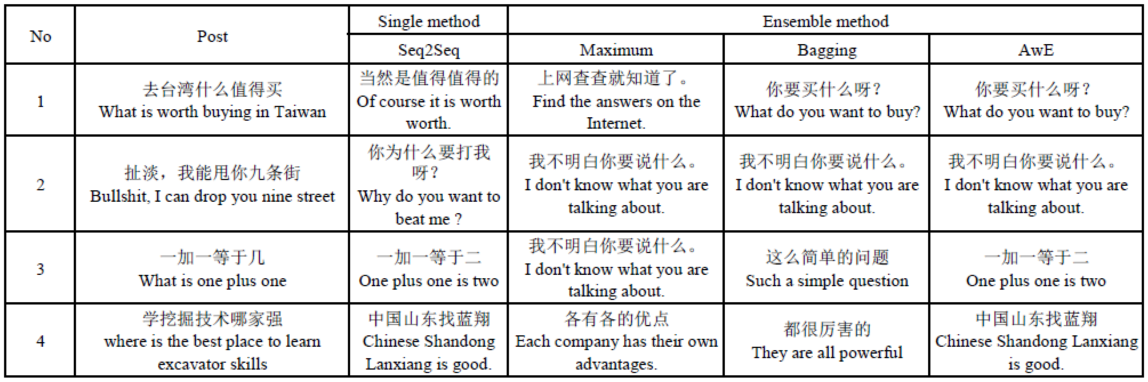
- Bad(0): The generated response is not semantically relevant to the post or there are some grammatical mistakes in the response.
- Normal(1): The generated response has no grammatical mistakes and is semantically relevant to the post. But it is a high frequency response or it can only be suitable to the post in some specific scenarios.
- Good(2): The generated response is free of mistakes and semantically relevant to the post. Additionally, it is a very satisfying response to the post.

5.2 Results

Table 3 shows the BLEU (BLEU-4) of our methods and three benchmarks. We can find that AwE outperforms three benchmarks. Also we can see that there is nearly no difference among three benchmarks. The main reason we conjecture is that the two ensemble benchmarks do not generate more similar responses to the reference responses than single model. Instead, they generate more high frequency responses, which is shown in Table 4. High frequency responses are the responses that appear many times in the dataset and can response to many different posts, such as "I don't know", "I

Methods	Mean Score	Good(2)	Normal(1)	Bad(0)	Agreement
Seq2Seq	0.903	14.1%	62.2%	23.7%	0.480
Maximum	0.905	11.7%	67.1%	21.2%	0.505
Bagging	0.906	12.1%	66.4%	21.5%	0.436
AwE	**0.989**	**15.2%**	**68.5%**	**16.3%**	0.412

Table 5: Human evaluation results



No	Post	Single method	Ensemble method		
		Seq2Seq	Maximum	Bagging	AwE
1	去台湾什么值得买 What is worth buying in Taiwan	当然是值得值得的 Of course it is worth worth.	上网查查就知道了。 Find the answers on the Internet.	你要买什么呀？ What do you want to buy?	你要买什么呀？ What do you want to buy?
2	扯淡，我能甩你九条街 Bullshit, I can drop you nine street	你为什么要打我呀？ Why do you want to beat me ?	我不明白你要说什么。 I don't know what you are talking about.	我不明白你要说什么。 I don't know what you are talking about.	我不明白你要说什么。 I don't know what you are talking about.
3	一加一等于几 What is one plus one	一加一等于二 One plus one is two	我不明白你要说什么。 I don't know what you are talking about.	这么简单的问题 Such a simple question	一加一等于二 One plus one is two
4	学挖掘技术哪家强 where is the best place to learn excavator skills	中国山东找蓝翔 Chinese Shandong Lanxiang is good.	各有各的优点 Each company has their own advantages.	都很厉害的 They are all powerful	中国山东找蓝翔 Chinese Shandong Lanxiang is good.

Table 6: Responses generated by several methods

don't know what you are talking about" , "Such a simple question". Because nearly all top 100 responses of training dataset are high frequency responses, we use those responses to represent high frequency responses in Table 4. But, we must make it clear that high frequency responses are not limited to top 100 responses of training dataset.

The human evaluation results of experiments are shown in Table 5. We use agreement [Fleiss and others, 1971] to measure the inter-rater consistency. The agreement is represented by Fleiss' kappa which ranges from 0 to 1. All the agreement of methods in Table 5 range from 0.4 to 0.6, which means all the results are in the same level of agreement that is "Moderate agreement" [Landis and Koch, 1977].

From Table 5, we find that two ensemble benchmarks have lower ratio of bad responses than Seq2Seq. However they perform worse than Seq2Seq on the ratio of good responses. The main reason is that those two methods generate more high frequency responses, which are semantically relevant to the posts but are not good.

Compared to benchmarks, AwE is superior to them. As for the comparison with Seq2Seq, our method generates more normal and good responses. This may result from two reasons. Firstly, AwE is an ensemble method, which leads to it generating more semantically relevant and grammatically correct responses. Secondly, the auto-evaluator in AwE helps our method to obtain more good responses. The auto-evaluator is trained by generated pairs and real pairs whose labels are different. It assigns $f^{(k)}(\boldsymbol{x}_i) = -1$ if $\boldsymbol{y}'_i$ is high frequency and $\boldsymbol{y}_i$ is not. Thus $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ will obtain high weights in the next epoch, if $\boldsymbol{y}'_i$ is high frequency and $\boldsymbol{y}_i$ is not. By this way, our method can get more information from low frequency and semantically relevant responses and reduce the probability of high frequency responses being generated, which leads to more good responses being generated. From Table 4, we can also see that our method generates less high frequency responses. In addition, two ensemble benchmarks do not have the auto-evaluator so that our method generates more good responses than them.

5.3 Case Study

Table 6 shows some responses generated by three benchmarks and AwE. From the first and the second examples, we can see that Seq2Seq generate semantically irrelevant responses to posts or responses with grammatical mistakes, whereas the three ensemble methods generate some semantically relevant responses without mistakes. This, as we conjecture, is caused by ensemble methods can always get better performance than single methods. From the third and fourth examples, we observe that AwE generate more satisfying responses than Maximum and Bagging. This is because, although the responses generated by Maximum and Bagging are semantically relevant to posts, they are high frequency. Instead, our method will decrease the weight of instances with high frequency responses in the training process so that our method generates less high frequency and more satisfying responses.

6 Conclusion

In this study, we propose Adaboost with Auto-Evaluation (called AwE) to improve performance of existing conversational models. Specifically, our method uses an auto-evaluator to evaluate the output generated by the conversational models so that the weights on the instances can be adjusted adaptively. We also theoretically analyze the upper bound of the training error for AwE when certain assumption on the auto-evaluators holds. From empirical experiments, we conclude that AwE visibly outperforms existing conversational models and other ensemble methods.

Acknowledgments

Juncen Li, Fen Lin and Cheng Niu are supported by Tencent. Ping Luo and Ganbin Zhou were supported by the National Natural Science Foundation of China under Grant No. 61473274, and the National Key Research and Development Program of China under Grant No. 2017YFB1002104. We thank Wilson Tam and Yijun Xiao for their constructive advices. We also thank the anonymous IJCAI reviewers for their helpful feedback.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[Cortes *et al.*, 2014] Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. Ensemble methods for structured prediction. In *ICML*, 2014.

[Denton *et al.*, 2015] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[Fleiss and others, 1971] J.L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 1971.

[Freund and Schapire, 1997] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*. 2014.

[Landis and Koch, 1977] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977.

[Li *et al.*, 2016] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. In *ACL*, 2016.

[Li *et al.*, 2017] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *CoRR*, abs/1701.06547, 2017.

[Li *et al.*, 2018] Juncen Li, Ping Luo, Fen Lin, and Bo Chen. Conversational model adaptation via KL divergence regularization. In *AAAI*, 2018.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2015.

[Schwenk and Bengio, 1998] Holger Schwenk and Yoshua Bengio. Training methods for adaptive boosting of neural networks for character recognition. In *NIPS*, 1998.

[Seo *et al.*, 2017] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.

[Serban *et al.*, 2016] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 2016.

[Serban *et al.*, 2017] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, 2017.

[Shang *et al.*, 2015] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *ACL*, 2015.

[Sordoni *et al.*, 2015] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *ACL*, 2015.

[Sun *et al.*, 2012] Helei Sun, Jie Shen, and Bin Chen. Lbp based fast face recognition system on symbian platform. *AASRI Procedia*, 2012.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[Wang *et al.*, 2017a] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, 2017.

[Wang *et al.*, 2017b] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, 2017.

[Yin *et al.*, 2016] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering. In *IJCAI*, 2016.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

[Zhou *et al.*, 2017] Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. Mechanism-aware neural machine for dialogue response generation. In *AAAI*, 2017.