Boosting Dialog Response Generation

Wenchao Du
Language Technologies Institute
Carnegie Mellon University
wenchaod@cs.cmu.edu

Alan W Black
Language Technologies Institute
Carnegie Mellon University
awb@cs.cmu.edu

Abstract

Neural models have become one of the most important approaches to dialog response generation. However, they still tend to generate the most common and generic responses in the corpus all the time. To address this problem, we designed an iterative training process and ensemble method based on boosting. We combined our method with different training and decoding paradigms as the base model, including mutual-information-based decoding and reward-augmented maximum likelihood learning. Empirical results show that our approach can significantly improve the diversity and relevance of the responses generated by all base models, backed by objective measurements and human evaluation.

1 Introduction

Sequence-to-sequence models (Sutskever et al., 2014) has become one of the most popular approaches to dialog systems, for it provides a high degree of automation and flexibility. On the other hand, they are known to suffer from the "dull-response" problem (Li et al., 2015). Various research attempts have been made to improve the diversity of responses generated by sequence-to-sequence models. One line of research investigate alternatives to maximum likelihood learning and decoding, which is believed to be the main cause of monotonicity. (Li et al., 2015) employed a decoding objective based on mutual information between contexts and responses; (Li et al., 2017a) used reinforcement learning techniques for training the decoder to generate responses that maximize pre-defined rewards instead of perplexities; (Li et al., 2017b; Xu et al., 2017) adopted adversarial learning, in which a generator is trained to deceive a discriminator that tries to differentiate between generated responses and human responses. Beside changing training and decoding objectives, (Liu et al., 2018; Lison and Bibauw, 2017) considered reweighting data points by penalizing those with overly frequent responses or by emphasizing high-quality responses. (Serban et al., 2017; Zhao et al., 2017) introduced stochastic latent variables into their models to capture discourse information on an inter-utterance level. (Shao et al., 2017) experimented with a novel segment-based training and decoding paradigm to help mitigate the problem of redundancy and contradiction.

Yet another type of approach has not been investigated in the literature in the context of response generation – boosting and ensembling, despite having been studied for machine translation (Xiao et al., 2010; Zhang et al., 2017). Being a long established machine learning method (Freund and Schapire, 1997), the process typically involves iteratively training multiple models on reweighted instances according to the error of the previous models and combining these models. The idea has been recently revived and extended to generative models and image generation, which also suffers from diversity problem (Tolstikhin et al., 2017; Grover and Ermon, 2018). In computer vision, the state-of-the-art models tend to generate a few categories of objects all the time and ignore the rest, known as the problem of "missing modes". Boosting has been shown to significantly improve the coverage of image generation models.

For language generation, given the prior success with data re-weighting and bootstrap approach (Zhang et al., 2017; Liu et al., 2018), we believe dialog response generation may benefit from boosting as well. In this work, we designed a principled framework of boosting response generation, based on the recently developed theory of boosting generative models. Moreover, we combined boosting with different training and/or decoding paradigms, and empirically show that boosting can invariably improve them, in both quantitative and

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 38–43
Florence, Italy, July 28 - August 2, 2019.

qualitative evaluation.

2 Preliminaries

For standard sequence-to-sequence approaches, training of models and decoding for generations are done through maximum likelihood estimation:

$$\log p(y \mid x) = \sum_{i=1}^{n} \log p(y_i \mid y_1 \dots y_{i-1}, x) \quad (1)$$

where x is the source (or context) and y is the target (or response). (Li et al., 2015) proposed a decoding objective based on mutual information of x and y to improve diversity:

$$MMI(x, y) = \log p(y \mid x) - \lambda p(y) \quad (2)$$

The conditional probability of y given x is estimated from sequence-to-sequence models, and the marginal probability of y from a separately trained language model.

Reward-augmented maximum likelihood learning (RAML) (Norouzi et al., 2016) incorporates task rewards into maximum likelihood training. An exponential payoff distribution is defined:

$$s(y \mid y^*; \tau) = \frac{1}{Z(y^*, \tau)} \exp\{r(y, y^*)/\tau\} \quad (3)$$

where y^* is the true target, r is a pre-defined reward function, and τ is temperature parameter. The model is trained to minimize the KL-divergence of the conditional distribution of y and the payoff distribution:

$$\sum_{x,y^*} D_{KL}(s(y \mid y^*) \mid\mid p(y \mid x)) = - \sum_{x,y^*} \sum_{y} s(y \mid y^*) \log p(y \mid x) + const \quad (4)$$

In multiplicative boosting, the density estimate of at each iteration T is given by:

$$q_T = h_T^{\alpha_T} q_{T-1} = \frac{\prod_{t=1}^{T} h_t^{\alpha_t}}{Z_T} \quad (5)$$

where h_t is t^{th} model's estimate, and α_t is models' weights. The goal of boosting is to approximate better the true distribution, P. It is shown in (Grover and Ermon, 2018) that if the model at each iteration can optimize for a re-weighted distribution of the following form perfectly:

$$d_t \propto (\frac{p}{q_t})^{\beta_t} \quad (6)$$

the distance of models' density estimate and the true distribution is decreasing, that is,

$$D_{KL}(P \mid\mid Q_t) \leq D_{KL}(P \mid\mid Q_{t-1}) \quad (7)$$

In equation (5) - (7), the density estimates are for the joint distribution of x and y. We make an additional assumption that the sources are uniformly distributed so that $p(x, y) = \frac{1}{n} p(y \mid x)$, for the ease of applying the boosting algorithm to sequence-to-sequence training.

The true distribution P is usually set to be uniform to boost the coverage of generative models. One of our innovations in this work is extending it to the exponential payoff distribution in RAML setting. The decreasing property of KL-divergence still holds, as the theoretical analysis is very much similar to that in (Grover and Ermon, 2018).

3 Design

We discuss some practical considerations when applying boosting framework to response generation problem.

3.1 Data Reweighting

In the generative boosting method of (6), the weights of data are inversely proportional to the perplexities of the responses. However, it is observed in experiments that the generic responses do not always have low perplexities. If not handled properly, such responses end up being boosted, and become the frequently generated responses at the next iteration.

In search for a consistent way to penalize generic responses with high perplexities, we first considered the discriminative boosting approach introduced in (Grover and Ermon, 2018). A discriminator is trained to differentiate between generated responses and human responses. The weights of data after discriminative boosting is the density ratio from the discriminator. The idea is closely related to generative adversarial learning (Goodfellow et al., 2014). However, in our case it is difficult to apply such approach. Because the generated responses are very limited, most classifiers can easily memorize all of them. The discriminators end up assigning extremely high probabilities to most of the human responses, and close-to-zero densities to generated responses. In other words, the amount of negative examples is

Model	Win	Loss	Tie
MLE	$37.6 \pm 6.4\%$	$17.6 \pm 4.0\%$	$44.8 \pm 6.4\%$
MMI	$36.0 \pm 9.2\%$	$16.8 \pm 6.8\%$	$47.2 \pm 8.8\%$
RAML	$44.8\% \pm 10.8\%$	$16.8 \pm 4.8\%$	$38.4 \pm 12.4\%$

Table 1: Human evaluation results. "Win" stands for the boosted model winning.

too small to train a discriminator to obtain good decision boundaries and generalization.

Instead, we resort to a simple rule-based discriminator. At each iteration, we maintain a list of most frequently generated responses, C_t. We choose a binary function to decide whether two responses, y, z, are similar, denoted by $sim(y, z)$. The discriminator is defined as

$$D_t(y) = \begin{cases} c & \text{if } \exists y_0 \in \bigcup_t C_t, sim(y, y_0) = 1 \\ 0.5 & \text{otherwise} \end{cases} \tag{8}$$

And the weights of data at round t is given by

$$d_t(x, y) \propto (\frac{p(x, y)}{q_t(x, y)})^{\beta_t} \frac{D_t(y)}{1 - D_t(y)} \tag{9}$$

In our experiments, the similarity function is chosen to be a predicate of whether there is an n-gram overlap with $n \geq 4$. We chose to be aggressive and set $c = 0$, so responses that are similar to those generated by previous models are excluded. The sizes of C_t is chosen to be around 20 so that the amount of training data reduces by about 10 percent at each iteration.

In our experiments, we include bootstrapping as an additional baseline. At each iteration, 80% of the data are randomly sampled for training and validation.

3.2 Model Combination

At decoding time, due to the discrete nature of text data, the optimization for the response that has highest probability (or mutual information) is intractable, so we use the following heuristics. Candidate responses are generated from the single best model using beam search. The candidates are then scored by all models, and the one with the highest average score is chosen. The model weights α_t are set to be uniform.

Since each model are trained on data with different weights, their un-normalized probability density estimates may have different scales. Hence, at decoding time, scores of each model are z-normalized with mean and standard deviation calculated from the training data.

3.3 Other Details

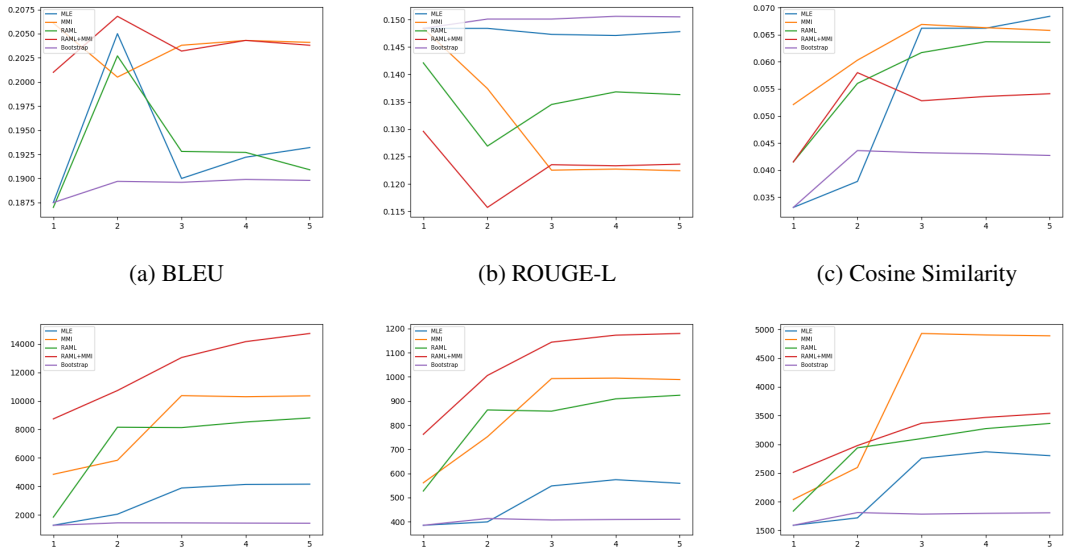
For RAML, the reward function is based on tf-idf matching – that is, the sum of products of term frequency and inverse document frequency of each word, divided by lengths. The rationale is to encourage models to include key content words in their generations. Empirically, we observed that RAML with aforementioned reward can generate better responses than MLE baseline even without boosting. The temperature parameter τ is set to be 0.1. To approximate the expectation term in the objective of RAML, three additional responses with highest rewards are selected from training data for each message-response pair in the beginning. We do not sample new responses at the following iterations for the sake of fair comparison. We set β_t in equation (6) to be $\frac{1}{bt}$ where b is between 10 and 20, and is tuned on validation set.

4 Experiments

We evaluate our algorithm on single-turn conversations from Persona Dataset (Zhang et al., 2018). Participants are instructed to converse according to their given personalized background. In the preparation of training data, persona descriptions are prepended to the sources, and all trailing punctuations are truncated from the responses.

We use a standard sequence-to-sequence architecture with attention mechanism. Both encoder and decoder are LSTMs with hidden size of 512 and input size of 300. Attentional contexts are weighted sums of hidden states of words in personas. We use Adam optimizer to train the model with learning rate of 0.001. All model parameters including word embeddings are randomly initialized between -0.1 and 0.1.

In addition to the base models mentioned before, we investigate the combination of RAML and MMI, in which models are trained with RAML and decoded with MMI.



(a) BLEU (b) ROUGE-L (c) Cosine Similarity

(d) Inertia (e) Number of unigrams (f) Number of trigrams

Figure 1: Quantitative results. X-axis is for iteration and y-axis for metrics. The numbers at iteration 1 represent the base models.

4.1 Quantitative Evaluation

We employ two standard word-overlap-based metrics, BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). We also performed embedding-based evaluation. We embed the responses using the word averaging approach by (Arora et al., 2016), and measure the cosine similarity of the embeddings of generated responses and true responses. To measure the diversity of the responses, we perform k-means clustering on their embeddings with 10 clusters, and measure the inertia. The larger inertia indicates more diversity. We also show statistics on number of distinct n-grams.

As can be seen in Figure 1, the general trend of boosting is that performance drastically improves up to the third model, then it slowly gets better or stays the same. Boosting is far better than bootstrapping. Boosting can improve lexical-level semantic similarity between generate responses and true responses, measured by cosine similarity. While BLEU scores only fluctuate in a tight range, ROUGE-L suffered from boosting a little, when used on base models that can generate more diversified responses. But we do not consider BLEU and ROUGE the most important metrics. Diversity measures, including count of distinct n-grams and inertia of clusters, are significantly improved by boosting. Combining RAML and MMI seems to give an advantage in BLEU (mainly because generated responses are longer), inertia, and number of unigrams.

4.2 Qualitative Evaluation

To ensure the diversified responses are as relevant as before boosting, we ask 5 annotators to evaluate a randomly sampled subset of 100 examples from each base model against its boosted counterpart. Each context are paired with two responses – one from the base model and one from the boosted model. The annotators are asked to choose the most appropriate response, or tie if they are equal. The results are shown in Table 1. On average, about 38 to 47 percent of the time the annotators showed no preferences, and boosted models beat base models for 36 to 45 percent of the trials. Note that all individual tests show annotators preferred the boosted model over the base model, except for one case, where the annotator chose MMI base model over the boosted model slightly more often. We also provide an example of generated responses in Table 2.

5 Conclusion

We investigated the use of boosting to improve the diversity and relevance of dialog response generation, with various training and decoding objectives including mutual-information-based decoding and reward-augmented maximum likelihood learning. Our combination of boosting and RAML for response generation is novel, and its combination

Context	my family lives in alaska . it is freezing down there .
Human	i bet it is oh i could not
Baseline	what do you do for a living
Boosted	do you live near the beach ? i live in canada

Table 2: Examples of generated responses from baseline sequence-to-sequence model and its boosted counterpart.

with MMI gives some of the most diversified results. Quantitative evaluation shows our method can substantially improve the diversity without harming the quality of generated responses. Our human evaluation provides evidence that diversified responses by boosting are even more appropriate than those generated from baseline models.

Acknowledgments

This material is based upon work supported by the National Science Foundation (Award No. 1722822). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation, and no official endorsement should be inferred.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Aditya Grover and Stefano Ermon. 2018. Boosted generative models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

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

Jiwei Li, Will Monroe, Tianlin Shi, Sėbastien Jean, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Pierre Lison and Serge Bibauw. 2017. Not all dialogues are created equal: Instance weighting for neural conversational models. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 384–394. Association for Computational Linguistics; Stroudsburg, PA.

Yahui Liu, Wei Bi, Jun Gao, Xiaojiang Liu, Jian Yao, and Shuming Shi. 2018. Towards less generic responses in neural conversation models: A statistical re-weighting method. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2769–2774.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. 2017. Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems*, pages 5424–5433.

Tong Xiao, Jingbo Zhu, Muhua Zhu, and Huizhen Wang. 2010. Boosting-based system combination for machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 739–748. Association for Computational Linguistics.

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.

Dakun Zhang, Jungi Kim, Josep Crego, and Jean Senellart. 2017. Boosting neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 271–276.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2204–2213.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 654–664.