Sampling Matters! An Empirical Study of Negative Sampling Strategies for Learning of Matching Models in Retrieval-based Dialogue Systems

Jia Li[1], **Chongyang Tao**[1], **Wei Wu**[2], **Yansong Feng**[1], **Dongyan Zhao**[1,3] and **Rui Yan**[1,3*]
[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
[2]Microsoft Corporation, Beijing, China
[3]Center for Data Science, Peking University, Beijing, China
[1,3]{lijiaa,chongyangtao,fengyansong,zhaody,ruiyan}@pku.edu.cn
[2]{wuwei}@microsoft.com

Abstract

We study how to sample negative examples to automatically construct a training set for effective model learning in retrieval-based dialogue systems. Following an idea of dynamically adapting negative examples to matching models in learning, we consider four strategies including minimum sampling, maximum sampling, semi-hard sampling, and decay-hard sampling. Empirical studies on two benchmarks with three matching models indicate that compared with the widely used random sampling strategy, although the first two strategies lead to performance drop, the latter two ones can bring consistent improvement to the performance of all the models on both benchmarks.

1 Introduction

In this work, we study the problem of response selection as an approach to implementing a retrieval-based dialogue system (Ji et al., 2014; Wang et al., 2013). A key step in response selection is measuring the matching degree between a conversation context and a response candidate. Existing studies focus on constructing a matching model with sophisticated neural architectures (Lowe et al., 2015; Zhou et al., 2016; Yan et al., 2016; Wu et al., 2017; Zhang et al., 2018; Zhou et al., 2018; Tao et al., 2019), but pay little attention to how to effectively learn such architectures from data. On the one hand, it is well known that learning of complicated neural architectures requires large-scale high quality training data; on the other hand, since human labeling is expensive and exhausting, most of the existing work just adopts a simple heuristic to automatically build a training set where human responses are treated as positive examples and negative response candidates are randomly sampled. Such a training set might contain many false negatives and trivial true negatives that are very easy to distinguish from those true positives. As a result, models with advanced architectures can only reach sub-optimal performance after learning (Wu et al., 2018).

In this paper, instead of configuring new architectures, we investigate how to improve the performance of existing matching models with a better learning method. A learning method usually involves choice of loss functions and construction of training data, and we are particularly interested in automatic training data construction, as data are often more crucial to the performance of models. The key problem in training data construction lies in how to properly choose negative examples, and our idea is that negative examples should adapt to the matching models at different learning stages. Following this idea, we consider four negative sampling strategies, namely minimum sampling, maximum sampling, semi-hard sampling, and decay-hard sampling. In the first two strategies, a response candidate that corresponds to the minimal or the maximal matching score at the current step is picked from a pool as a negative example for the next step; and in the latter two strategies, we select negative examples by considering how hard they are to the current matching models. The semi-hard sampling prefers candidates with moderate difficulty to avoid both false negatives and trivial true negatives, and the decay-hard sampling gradually increases the difficulty of negative samples with the training process going on.

We compare different sampling strategies with three matching models in different levels of complexity on two benchmarks. Evaluation results indicate that minimum sampling and maximum sampling are inferior to randomly sampling, and both semi-hard sampling and decay-hard sampling can bring consistent improvement to the perfor-

[*]Corresponding author: Rui Yan (ruiyan@pku.edu.cn).

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 1291–1296,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

mance of all the three models on both data sets.

Our contributions include (1) a systematic comparison of different sampling strategies with two benchmarks; and (2) proposal of semi-hard and decay-hard negative sampling strategies that can generally improve the performance of existing matching models on benchmarks.

2 Learning a Matching Model for Response Selection

Suppose that $\mathcal{D} = \{(c_i, \{r^+_{i,j}\}_{j=1}^{n_i^+}, \{r^-_{i,k}\}_{k=1}^{n_i^-})\}_{i=1}^N$ is a training set, where c_i is a conversation context, $\forall j \in \{1, \ldots, n_i^+\}$, $r^+_{i,j}$ is a positive response candidate that properly replies to c_i, and $\forall k \in \{1, \ldots, n_i^-\}$, $r^-_{i,k}$ is a negative response candidate that is used to indicate errors in responding to a model, then the learning problem of response selection is to estimate a matching model $g(\cdot, \cdot)$ from $\mathcal{D}$, which can be formulated as

$$\underset{\Theta}{\text{argmin}} \sum_{i=1}^{N} \Big[\sum_{j=1}^{n_i^+} L(+1, g(c_i, r^+_{i,j})) + \sum_{k=1}^{n_i^-} L(-1, g(c_i, r^-_{i,k})) \Big], \tag{1}$$

where Θ are the parameters of $g(\cdot, \cdot)$, and $L(\cdot, \cdot)$ is a loss function.

In practice, $L(\cdot, \cdot)$ is usually set as cross entropy, then the remaining problems become (1) how to define $g(\cdot, \cdot)$; and (2) how to construct $\mathcal{D}$ given that large-scale human labeling is infeasible. Existing work has paid enough effort to solving Problem (1), but only adopts a simple heuristic for Problem (2) where human responses are treated as $\{r^+_{i,j}\}_{j=1}^{n_i^+}$ (a common case is $n_i^+ = 1$ since only one response is available for a specific context), and some randomly sampled responses are utilized as $\{r^-_{i,k}\}_{k=1}^{n_i^-}$. The problem with this heuristic is that there is no guarantee on what responses will be sampled as $\{r^-_{i,k}\}_{k=1}^{n_i^-}$: some of them could be false negatives, and some could be too trivial to recognize. The clear drawback of random sampling motivates us to pursue better negative sampling strategies in training data construction, as will be elaborated in the next section.

3 Model Adaptive Negative Sampling

Our idea is to dynamically adapt negative examples to matching models in learning. The idea is inspired by how human learn knowledge: they adjust their learning materials according to their learning progress. Based on this idea, we consider four strategies to sample negative examples:

Minimum sampling: the strategy used to be exploited in answer selection (Rao et al., 2016), and here we apply it to response selection for open domain dialogue systems. Suppose that $\hat{g}(\cdot, \cdot)$ is a matching model obtained from the t-th mini-batch, then in the $(t+1)$-th mini-bach, we try to select the easiest negative example for a context c according to $\hat{g}(\cdot, \cdot)$, which can be formulated as

$$\underset{r^- \in \mathcal{R}^-}{\text{argmin}}\, \hat{g}(c, r^-), \tag{2}$$

where $\mathcal{R}^-$ is a pool of negative examples for c.

Maximum sampling: similar to minimum sampling, the strategy is also borrowed from answer selection (Rao et al., 2016), but attempts to select the hardest negative example by

$$\underset{r^- \in \mathcal{R}^-}{\text{argmax}}\, \hat{g}(c, r^-), \tag{3}$$

Semi-hard sampling: the first two strategies might be too aggressive, as the easiest negative example could bring no new information to $\hat{g}(\cdot, \cdot)$, and the hardest one could be a false negative. To avoid both cases, we propose a semi-hard sampling strategy which selects a negative sample with moderate difficulty. Formally, the strategy is defined as

$$\underset{r^- \in \mathcal{R}^-}{\text{argmin}}\ |\hat{g}(c, r^+) - \hat{g}(c, r^-) - \alpha|, \tag{4}$$

where r^+ is the positive response candidate of c[1], and α is a constant. In Equation (4), we exploit α as a margin to control the distance of matching degree between the selected r^- and r^+. The one with a matching degree closest to $\hat{g}(c, r^+) - \alpha$ is picked as a negative example.

Decay-hard sampling: to imitate the behavior that human gradually increase the difficulty of their learning materials at different stages, we propose a decay-hard sampling strategy which decays the margin in Equation (4) with the training process going on. Specifically, we consider two methods, namely exponential decay and linear decay.

[1] We assume that only one positive example is available as a common case, otherwise $\hat{g}(c, r^+)$ is replaced with $\min_{r^+ \in \mathcal{R}^+} \hat{g}(c, r^+)$.

In the first method, the margin shrinks in a exponential speed. In the t-th mini-bach, the margin α_t is defined by

$$\alpha_t = \varphi \times \exp(\omega \times t), \quad (5)$$

where $0 < \varphi < 1$ and $-1 < \omega < 0$ are parameters. In the second method, the margin linearly becomes small with the training steps, which is given by

$$\alpha_t = \lambda \times t + \theta, \quad (6)$$

where $0 < \theta < 1$ and , $-1 < \lambda < 0$ are parameters. We carefully choose θ and λ to make sure that $\alpha_t > 0, \forall t \in \{1, \ldots, T\}$, where T refers to the maximum number of iterations.

Note that descriptions above assume that for each context, only one negative example is selected from the pool. This is for a fair comparison with random sampling on benchmarks, as most of the existing work (Wu et al., 2017; Zhou et al., 2018) utilizes one negative example per context in training. It is easy to extend the strategies to sample multiple negative examples (e.g., by picking the top l examples with matching scores closest to $\hat{g}(c, r^+) - \alpha$ in semi-hard sampling).

4 Experiments

We compare different sampling strategies on two benchmarks.

4.1 Experimental Setup

The first data set we use is the Ubuntu Dialogue Corpus (Lowe et al., 2015) collected from chat logs of the Ubuntu Forum. We use the version provided by Xu et al. (2017). The data contains 1 million context-response pairs for training, and 0.5 million pairs for validation and test.

Following (Lowe et al., 2015), we employ recall at position k in n candidates (R_n@k) as evaluation metrics.

Besides the Ubuntu data, we also choose the Douban Conversation Corpus (Wu et al., 2017) as an experimental data set. The data consists of multi-turn Chinese conversations collected from Douban group[2]. There are 1 million context-response pairs for training, 50 thousand pairs for validation, and $6,670$ pairs for test.

Following (Wu et al., 2017), we employ R_n@ks, mean average precision (MAP), mean reciprocal rank (MRR)(Voorhees et al., 1999) and precision at position 1 (P@1) as evaluation metrics.

4.2 Matching Models

The following matching models are selected:

Dual-LSTM (Lowe et al., 2015): the model individually encodes a context and a response candidate with LSTMs, and then calculates a matching score based on the final states of the two LSTMs.

SMN (Wu et al., 2017): the model lets each utterance in a context interact with a response, and forms the interaction matrices into a matching vector with CNN. The matching vectors are finally accumulated with an RNN as a matching score.

DAM (Zhou et al., 2018): the model performs matching in a similar manner as SMN but represents a context and a response candidate with stacked self-attention and cross-attention.

In terms of both complexity and performance under random sampling, Dual-LSTM<SMN<DAM. Regarding to baselines, we consider two random sampling strategies. The first one is a static strategy where negative examples are fixed in the entire learning procedure. This is how existing work learns a matching model with the data described in Section 4.1, and we denote the model as Model-Base. The second one is a dynamic strategy where in each mini-batch, a negative example is randomly sampled from $\mathcal{R}^-$. This is a simplification of model adaptive sampling strategies, and we denote a model learned with this strategy as Model-Rand. We denote a model trained with minimum sampling, maximum sampling, semi-hard sampling, exponential decay-hard sampling, and linear decay-hard sampling as Model-Min, Model-Max, Model-Semi, Model-EDecay, and Model-LDecay respectively. All models are implemented with TensorFlow and tuned on the validation sets. We make sure that Model-Base achieves the performance on both data sets as that reported in (Zhou et al., 2018).

4.3 Implementation Details

For static random sampling, we just use the published training sets of both data. For the remaining sampling strategies, we randomly sample 10 responses[3] for each context from the training sets as a pool of negative examples at each epoch. Every time, one response is sampled from the pool as a negative example for a context. Models trained with different sampling strategies are

[2] https://www.douban.com/group

[3] We set the size of $\mathcal{R}^-$ as 10 to balance efficacy and efficiency.

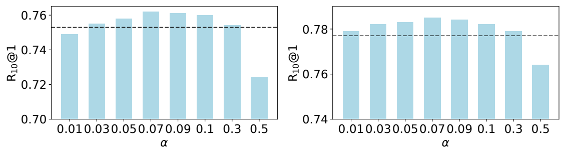
Metrics	Ubuntu Corpus				Douban Conversation Corpus					
Strategies	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	MAP	MRR	P@1	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
Dual-LSTM-Base	0.901	0.638	0.784	0.949	0.485	0.527	0.320	0.187	0.343	0.720
Dual-LSTM-Rand	0.909	0.691	0.818	0.952	0.541	0.581	0.395	0.236	0.396	0.742
Dual-LSTM-Min	0.832	0.561	0.640	0.883	0.446	0.495	0.298	0.164	0.312	0.664
Dual-LSTM-Max	0.839	0.579	0.672	0.898	0.477	0.510	0.300	0.178	0.323	0.689
Dual-LSTM-LDecay	0.915	0.701	0.830	0.955	0.543	0.586	0.405	0.242	0.402	0.749
Dual-LSTM-EDecay	0.916	0.703	0.833	0.957	0.544	0.588	0.406	0.245	0.403	0.752
Dual-LSTM-Semi	**0.918**	**0.706**	**0.835**	**0.958**	**0.546**	**0.591**	**0.408**	**0.246**	**0.405**	**0.754**
SMN-Base	0.926	0.726	0.847	0.961	0.529	0.569	0.397	0.233	0.396	0.724
SMN-Rand	0.931	0.753	0.859	0.963	0.543	0.587	0.406	0.240	0.407	0.751
SMN-Min	0.850	0.569	0.714	0.920	0.521	0.563	0.384	0.229	0.387	0.718
SMN-Max	0.859	0.667	0.789	0.944	0.523	0.565	0.388	0.227	0.392	0.721
SMN-LDecay	0.933	0.759	0.861	0.965	0.549	0.600	0.421	0.253	0.410	0.755
SMN-EDecay	0.933	0.760	0.862	0.966	0.552	0.602	0.424	0.261	0.412	0.758
SMN-Semi	**0.934**	**0.762**	**0.865**	**0.967**	**0.554**	**0.605**	**0.425**	**0.253**	**0.412**	**0.759**
DAM-Base	0.938	0.767	0.874	0.969	0.550	0.601	0.427	0.254	0.410	0.757
DAM-Rand	0.940	0.777	0.878	0.971	0.563	0.612	0.436	0.261	0.427	0.783
DAM-Min	0.923	0.721	0.841	0.958	0.539	0.585	0.408	0.247	0.407	0.744
DAM-Max	0.928	0.736	0.852	0.962	0.551	0.596	0.421	0.258	0.411	0.754
DAM-LDecay	0.942	0.784	0.880	0.973	0.573	0.617	0.442	0.270	0.437	0.789
DAM-EDecay	0.943	0.784	0.882	0.973	0.575	0.621	0.444	0.272	0.439	0.793
DAM-Semi	**0.944**	**0.785**	**0.883**	**0.974**	**0.580**	**0.623**	**0.450**	**0.279**	**0.443**	**0.796**

Table 1: Evaluation results of different sampling strategies on the two data sets. Numbers in bold indicate the best strategies for the corresponding models on specific metrics.

tuned with the same validation sets and evaluated with the same test sets (i.e., the original released sets). We vary α in semi-hard sampling in $\{0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5\}$, and choose 0.07. In decay-hard sampling, we set ω, φ, λ, and θ as $-1.5e-5$, 0.1, $-8.75e-7$, and 0.1 on respectively.

4.4 Evaluation Results

Table 1 reports evaluation results on the two data sets. We can see that both semi-hard sampling and decay-hard sampling can generally improve the three matching models on both data sets. Minimum sampling and maximum sampling are consistently worse than random sampling, which verified the statement we make in Section 3 that the two strategies are too aggressive. Decay-hard sampling is a little worse than semi-hard sampling. The reason might be that false negatives are introduced to learning by decay-hard at late stages of training. Dynamic random sampling is better than static random sampling, because models can leverage more negative examples from the pool. It is worth noting that the proposed sampling strategies does not change the elapsed time for prediction, despite a little more training time. On the other hand, we do see improvement on the two data sets. Therefore, we believe it is worth paying a little more training time but obtaining the improvement.



(a) SMN (b) DAM

Figure 1: The effect of α to semi-hard sampling on the Ubuntu data. Dashed lines represent performance of the dynamic random sampling strategy.

Besides the comparison of different sampling strategies, we are also interested in how the hyperparameter α affects the performance of semi-hard sampling. Figure 1 shows how the performance of SMN and DAM changes with respect to different margins (i.e., α). We observe a similar trend for both models: they first increase monotonically until the margin reaches 0.07, and then drop as the margin increases. Particularly, the performance of both models with the semi-hard sampling strategy is worse than that with the random sampling strategy when the margin reaches 0.5. The reason behind the phenomenon is that when the margin is small, semi-hard sampling is similar to maximum sampling and will introduce false negatives into learning, while when the margin is large, semi-hard sampling is like minimum sampling and is prone to provide trivial samples to learning.

5 Related works

Negative sampling strategies have been studied in many machine learning tasks. In the computer vision fields, Faghri et al. (2017) studies hard negatives and introduces a simple change to common loss function on image-caption retrieval tasks. Guo et al. (2018) proposes a fast negative sampler which chooses negative examples that are most likely to meet the requirements of violation according to the latent factors of image. In natural language processing fields, Kotnis and Nastase (2017) analyses the impact of negative sampling strategies on the performance of link prediction in knowledge graphs. Saeidi et al. (2017) studies the affect of a tailored sample strategy on the performance of document retrieval task. Rao et al. (2016) uses three negative strategies to select the most informative negative samples on the pairwise ranking model for answer selection. Xu et al. (2015) introduces a straightforward negative sampling strategy to improve the assignment of subjects and objects on a convolution neural network. To our best knowledge, this is the first work to empirical study of negative sampling strategies for learning of matching models in multi-turn retrieval-based dialogue systems, which may enlighten future works in the learning of retrieval-based dialogue systems.

6 Conclusions

We present minimum sampling, maximum sampling, semi-hard sampling, and decay-hard sampling as four model adaptive negative sampling strategies to learn a matching model for retrieval-based dialogue systems. Evaluation results with three models on two benchmarks indicate that although minimum sampling and maximum sampling are worse than random sampling, both semi-hard sampling and decay-hard sampling can generally improve the performance of the models on both data sets. In the future, we would like to extend our negative sampling strategies to other tasks.

Acknowledgments

We thank the reviewers for their valuable comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC No. 61876196, NSFC No. 61828302, and NSFC No. 61672058).

References

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Guibing Guo, Songlin Zhai, Fajie Yuan, Yuan Liu, and Xingwei Wang. 2018. Vse-ens: Visual-semantic embeddings with efficient negative sampling. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1913–1916. ACM.

Marzieh Saeidi, Ritwik Kulkarni, Theodosia Togia, and Michele Sama. 2017. The effect of negative sampling strategy on capturing semantic similarity in document embeddings. In *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*, pages 1–8.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1–11, Florence, Italy. Association for Computational Linguistics.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82. Citeseer.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 935–945.

Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2018. Learning matching models with weak supervision for response selection in retrieval-based chatbots. *arXiv preprint arXiv:1805.02333*.

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, pages 496–505.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. In *Proceedings of the 2017 International Joint Conference on Neural Networks*, pages 3506–3513.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64. ACM.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752. Association for Computational Linguistics.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1118–1127.