Do It Once: An Embarrassingly Simple Joint Matching Approach to Response Selection

Linhao Zhang[1], **Dehong Ma**[2], **Sujian Li**[1], **Houfeng Wang**[1]
[1]MOE Key Lab of Computational Linguistics, Peking University, Beijing, 100871, China
[2]Baidu Inc., China
{zhanglinhao, lisujian, wanghf}@pku.edu.cn
madehong@baidu.com

Abstract

Existing matching models for response selection adopt the independent matching (*IM*) approach. To complete a prediction, they have to perform *N* independent matches, where *N* is the number of response options. In this paper, we explore a joint matching (*JM*) approach which performs matching only once regardless of the number of options. The *JM* approach does not change the structure of matching component but only modifies its input and output format. It also enables a cheap but effective data augmentation method. Extensive experiments on the MuTual dataset demonstrate that, even with the simplest formulation, JM outperforms *IM* approach by a large margin and reduces training time by over half.

1 Introduction

The availability of large-scale datasets has driven the development of neural dialogue systems. One important task in dialogue systems is response selection, which plays an essential role in retrieval-based chatbots (Ji et al., 2014). It aims to select the best-matched response from a set of response options for a dialogue. As shown in Figure 1, given a dialogue context and four response options, we need to choose the only logically correct one.

Previous work in response selection follows an *independent matching (IM)* approach and computes a matching score for each of the *N* response options independently. Various matching models following this approach have been proposed (Zhou et al., 2016; Wu et al., 2017; Zhou et al., 2018; Chaudhuri et al., 2018; Tao et al., 2019; Yuan et al., 2019). Despite its success in pre-BERT era, we argue that the *IM* approach does not make full use of the ability of pretrained encoders (such as BERT and RoBERTa) to encode multiple sentences, hence may hinder both efficiency and effectiveness. Specifically, to

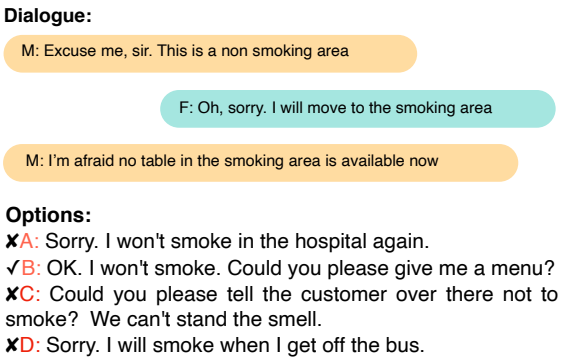

Figure 1: Example of response selection.

complete a prediction, the *IM* approach has to perform *N* independent matches, which means *N* gradient computations (where *N* is the number of response options). Besides, the dialogue context is repeatedly encoded *N* times, which further contributes to the inefficiency. The other drawback is that options in these models are independent and agnostic of each other. In reality, humans often compare all the options and utilize their correlations to make a comprehensive decision.

In this paper, we describe a *joint matching* (*JM*) approach for this task. For any matching model, we do not change its inner structure but only modify its input and output format. Specifically, we first add a special token at the start of each option, and then concatenate all options into a single sequence. The option sequence is then matched as a whole with the dialogue context. Finally, we extract vectors corresponding with the special token to calculate matching scores. Note that *JM* can complete a prediction with a single match, which means it only requires one gradient computation and context encoding. Besides, thanks to the self-attention mechanism (Vaswani et al., 2017) of BERT-based matching models, options can now directly attend to each other, rather than being agnostic.

Another advantage of *JM* approach is that it nat-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4872–4877

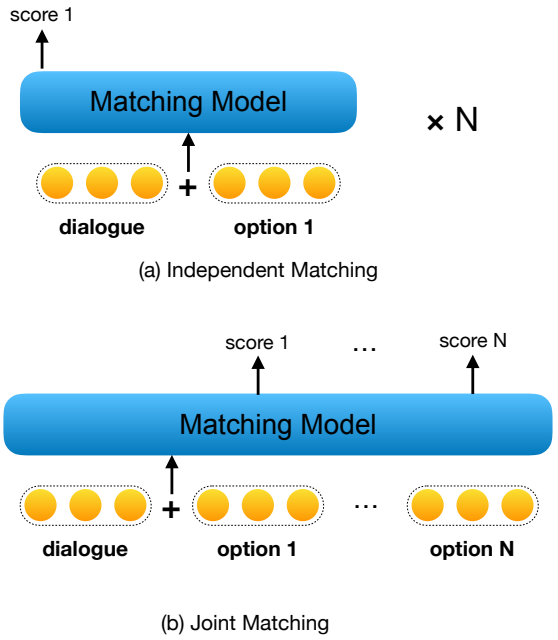

Figure 2: Overview of Independent Matching and Joint Matching.

urally enables a simple yet effective data augmentation method. The basic idea is that since options are sequentially concatenated in *JM*, new training instances can be easily created by changing the permutation order of options. Therefore, a dialogue with M response options can create at most $M!$ (factorial M) times as many training instances.

We conduct experiments on the MuTual dataset (Cui et al., 2020), a publicly available English dataset for multi-turn dialogue response selection. Results show that *JM* advances *IM* on three matching models and can significantly reduce training time. Besides, the permutation-based data augmentation method gives further improvement.

2 Model

The overview of IM and JM is shown in Figure 2. We describe the details in the following subsections[1].

2.1 Background

Given a dialogue $\mathcal{D}$ with M utterances $\{U_i\}_{i=1}^{M}$, and a set of N response options $\{O_j\}_{j=1}^{N}$, the goal of response selection is to select the logically correct option $\hat{O}$.

Previous work (Cui et al., 2020) shows that pretrained matching models define the state-of-the-art on this task. Similar to using BERT for sentence-pair classification (Devlin et al., 2019), they first concatenate the context (sentence A) and a candidate response (sentence B) as BERT input (i.e., "[CLS] Excuse me ... [SEP] Sorry ... [SEP]"). On the top of BERT, a fully-connected layer is used for transforming the [CLS] token representation to the matching score. In order to compete a prediction, M independent matchings have to be made, where M is the number of options.

2.2 Joint Matching

Instead of conducting N times of independent matches, we make the first step outside the IM framework and explore a joint matching approach for this task. We first adds a special token $[\text{OP}]^j$ at the start of the j^{th} option. It is a token used to aggregate the matching information between the context and the j^{th} option into a single vector. We then concatenate all the options into a single sequence. Formally,

$$S_O = [OP]^1 O_1 [OP]^2 O_2 ... [OP]^N O_N \quad (1)$$

For the dialogue context, we concatenate all the utterances into a single sequence. Formally,

$$S_D = U_1 U_2 ... U_M \quad (2)$$

The two sequences are then separated with a [SEP] token and fed into our pretrained encoder. Formally,

$$X = [CLS] S_D [SEP] S_O [SEP] \quad (3)$$

For any BERT-based matching model, suppose the output embeddings of the model are $H_t \in \mathbb{R}^{|X| \times d}$. To perform scoring, we first extract outputs corresponding to $[\text{OP}]^j$ and represent them as $h^{[OP]^j}$. The only new parameters learned are a score vector $W \in \mathbb{R}^d$. The probability of option j being the answer is computed as a dot product between $h^{[OP]^j}$ and W followed by a softmax over all of the options. Formally,

$$P_j = \frac{e^{W \cdot h^{[OP]^j}}}{\sum_i e^{W \cdot h^{[OP]^i}}} \quad (4)$$

The training objective is the log-likelihood of the correct answer[2].

[1] The code is at https://github.com/gitzlh/JM-Matching

[2] For the score layer, an intuitive approach is to use $h^{[CLS]}$ instead of $h^{[OP]}$ and cast the prediction into a M-classes classification. However, we found that this approach leads to poor performance.

In this way, the JM approach only needs to match the context sequence and the option sequence once. Compared with the *IM* approaches, JM is computational efficient in two ways: 1) It encodes the dialogue context only once, instead of M times. However, this benefit is partially offset by the fact that the complexity of transformer grow quadratically with the length of input 2) More importantly, IM approaches need to compute gradients M times for each training step. Besides, in each self-attention layer of the BERT-based matching model, options can directly attend to and interact with each other. This process mimics how humans solve multi-choice questions, that we often compare all the options before making the decision.

2.3 Permutation-Based Data Augmentation

Another advantage of *JM* is that it naturally enables a permutation-based data augmentation (PBDA) method, which can generate high-quality labeled data to improve response selection.

Specifically, since the input of our model is organized as Equation 3, we can create new training instances by simply changing the concatenation order of the options. For example, from $[OP]^1O_1[OP]^2O_2...[OP]^NO_N$ to $[OP]^2O_2[OP]^1O_1...[OP]^NO_N$, we create a new training example (see Figure 2). Correspondingly, the ground-truth label of the training instance may be changed. In this way, a single dialogue can create at most $M!$ times training instances.

3 Experiments

3.1 Dataset

We evaluate our model on the Mutual dataset (Cui et al., 2020), a human-labeled, open-domain and reasoning-based dataset for multi-turn response selection. Compared with previous datasets (Lowe et al., 2015; Zhang et al., 2018; Welleck et al., 2019), MuTual is more challenging since it requires some reasoning ability. Models that achieve close-to-human performance on previous datasets, still perform far behind human performance on MuTual. The statistics of MuTual are shown in Table 1. Note that since Mutual has 4 options for each dialogue, PBDA can thus creates at most 24 (4!) times as many training instances.

3.2 Settings

We use PyTorch to implement JM on three matching models. We adopt AdamW (Loshchilov and Hutter, 2018) as our optimizer, and the peak learning rate and warmup proportion are set to 1e-5 and 0.06, respectively. We use the largest batch size that fits in the memory of our GPU and use gradient accumulation for an effective batch size of 32. Dropout (Srivastava et al., 2014) is employed before the score layer with a rate of 0.1. We train our model for 15 epochs and choose the model that reports the highest R@1 on the validation set.

Following previous work (Cui et al., 2020), we evaluate our model with recall at position 1 in 4 candidates (R@1), recall at position 2 in 4 candidates (R@2) and Mean Reciprocal Rank (MRR).

	MuTual
Training set	7088
Validation set	886
Test set	886
# Avg. Turns / Dialogue	4.73
# Avg. Words / Utterance	19.57
# Options	4

Table 1: Statistics of MuTual.

4 Results

4.1 Main Performance

Table 2 gives the comparison of IM and JM on three matching models on the MuTual dataset. Note that we only experiment with pretrained matching models given that they have an overwhelming advantage over non-pretrained models. For a fair comparison, we report baseline results both from the official reports of MuTual (Cui et al., 2020) and our own implementation.

Our first observation is that RoBERTa-based models significantly outperform BERT-based models, suggesting that RoBERTa is a more powerful feature extractor. More importantly, we note that our JM approach outperforms IM approach on all three matching models. For example, BERT-JM improves over BERT-IM by 6% (absolute) R@1. We suppose that this is because the JM approach concatenates all the options as the model input, and, thanks to the self-attention mechanism, each option can directly attend to each other. In this way, JM can make a more comprehensive decision and boost the performance especially on challenging datasets like MuTual.

The conclusion holds true for larger pretrained matching models such as RoBERTa-large. As

Methods	R@1	R@2	MRR
Human performance	0.938	0.971	0.964
BERT-IM (Cui et al., 2020)	0.648	0.847	0.795
BERT-IM †	0.641	0.853	0.793
BERT-JM	**0.702**	**0.904**	**0.833**
RoBERTa-IM (Cui et al., 2020)	0.713	0.892	0.836
RoBERTa-IM †	0.770	0.912	0.868
RoBERTa-JM	**0.784**	**0.933**	**0.880**
RoBERTa-large-IM †	0.844	0.958	0.914
RoBERTa-large-JM	**0.870**	**0.973**	**0.930**

Table 2: Main results on MuTual. We can see that *JM* outperforms *IM* approach for every pretrained encoders. † means our own implementation results. Note that the JM results are achieved without data augmentation.

Methods	Forward	Backward	Total
RoBERTa-IM	116	412	528
RoBERTa-JM	**71**	**168**	**239**

Table 3: Average training time (second) per epoch. RoBERTa-IM and RoBERTa-JM both use the largest batch size available on the same GPU.

Methods	R@1	R@2	MRR
RoBERTa-JM	0.784	0.933	0.880
RoBERTa-JM 4x	0.793	**0.947**	0.887
RoBERTa-JM 8x	**0.813**	0.942	**0.896**
RoBERTa-JM 24x	0.807	0.942	0.892

Table 4: PBDA results. 4x, 8x and 24x mean augmenting the data size by 4, 8 and 24 times, respectively.

shown in Table 2, RoBERTa-large-JM brings about 3% (absolute) improvement over RoBERTa-large-IM in terms of R@1 and even surpasses human performance in terms of R@2.

4.2 Training Time

In the task of response selection, the scalability of the model becomes an issue when the number of options increases. In this subsection, We compare JM and IM with respect to training time[3]. As shown in Table 3, RoBERTa-JM reduces the training time by 55% compared with RoBERTa-IM. More detailed analysis shows that the reduction is mostly contributed to the backward propagation process. This is because to complete a prediction, RoBERTa-IM performs M independent matches and thus requires M gradient computations, a costly process. It also needs to encode the dialogue context M times, leading to computational inefficiency especially in multi-turn settings. By contrast, RoBERTa-JM requires only a single match[4].

[3]Both models are trained on a single NVIDIA TITAN Xp GPU.

[4]We note that this benefit is partially offset by the transformer's quadratic complexity with regard to the length of input. Suppose that the average length of the dialogue utterance and response option is L, then the time complexity of the IM and JM approach is $O((ML+L)^2N)$ and $O((ML+NL)^2)$, respectively

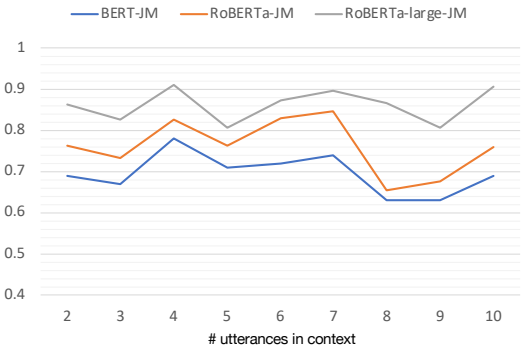

Figure 3: Performance on MuTual dev set across different contexts.

4.3 Data Augmentation

In this subsection, we conduct experiments to verify the effectiveness of PBDA.

As shown in Table 4, PBDA 4x brings a 1% (absolute) improvement in terms of both R@1 and R@2, showing that by simply permuting the options, we can create high-quality training instances. Besides, when increasing the data size by 8 times, we observe another 2% (absolute) improvement in terms of R@1. An interesting observation is that PBDA 24x does not further improve model performance, showing that there is a limit to the improvement brought by data augmentation.

4.4 Context Length

Following previous work (Wu et al., 2017), we further investigate how JM performs cross the length of context. As demonstrated in Figure 3, the performance of BERT-JM and RoBERTa-JM is generally satisfactory, except for the slight deterioration when the context has more than seven utterances. It can also deal with a short context that only has two utterances.

On the other hand, RoBERTa-large-JM consistently performs better than RoBERTa-JM, and when the context becomes longer, the gap becomes larger. It also gives more stable performance across different context lengths, further showing the strong representation ability of RoBERTa-large.

5 Conclusions

In this paper, we make the first step outside the independent matching framework and explore a joint matching approach for response selection. We also present an effective permutation-based data augmentation method. We conduct experiments on the MuTual dataset and demonstrate the effectiveness and efficiency of our approach. Besides, the proposed data augmentation further improves model performance.

6 Acknowledgments

The work is supported by National Natural Science Foundation of China under Grant No.62036001 and PKU-Baidu Fund (No. 2020BD021). The corresponding author of this paper is Houfeng Wang.

References

Debanjan Chaudhuri, Agustinus Kristiadi, Jens Lehmann, and Asja Fischer. 2018. Improving response selection in multi-turn dialogue systems by incorporating domain knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 497–507, Brussels, Belgium. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

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, Prague, Czech Republic. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

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)*, pages 496–505, Vancouver, Canada. Association for Computational Linguistics.

Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 111–120.

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)*, pages 2204–2213, Melbourne, Australia. 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, Austin, Texas. Association for Computational Linguistics.

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)*, pages 1118–1127, Melbourne, Australia. Association for Computational Linguistics.