Towards a Fast Response Selection: Selecting the Optimal Dialogue Response Once for All

Anonymous ACL submission

Abstract
Response selector, as an essential component of dialogue systems, aims to pick out an optimal response in a candidate pool to continue the dialogue. The current state-of-the-art methods are mainly based on an encoding paradigm called Cross-Encoder (Urbanek et al., 2019), which separately encodes each context-response pair and ranks the responses according to their fitness scores. However, such a paradigm is both inefficient and ineffective. Specifically, it has to repeatedly encode the same context for each response, which results in heavy inference cost. Also, without considering the relationship among the candidates, it is difficult to tell which one is the best candidate purely based on the fitness score of each candidate. To address this problem, we propose a new model called Panoramic-Encoder, which accepts all candidates and the context as inputs at once and allows them to interact with each other through a specially designed attention mechanism. Our method also allows us to naturally integrate some of the effective training techniques, such as the in-batch negative training. Extensive experiments across four benchmark datasets show that our new method significantly outperforms the current state-of-the-art while achieving approximately 3× speed-up at inference time.

1 Introduction
Nowadays, dialogue systems have gained increasing attention in the natural language processing community. Depending on the implementation, they can be categorized as retrieval-based (Lowe et al., 2015; Tao et al., 2019; Yuan et al., 2019) or generation-based (Vinyals and Le, 2015; Serban et al., 2016). The former one proceeds the conversation by selecting an optimal response from a candidate pool, while the latter continues the conversation using a proper response generated by a sequence-to-sequence model. Recent studies have shown that the generated-based solution can be a preferable choice in a dialogue system due to its intriguing property to generate more diverse and coherent responses (Roller et al., 2021). In such a solution, selecting an optimal response in the candidate pool also plays a vital role with the rise of an approach, called “sample-and-rank” (Adiwardana et al., 2020) in advanced generation-based chatbots (Zhang et al., 2020; Roller et al., 2021; Bao et al., 2021). The pipeline of this approach consists of first generating multiple response candidates from the generator and then selecting the best candidate as the response to the user by a selector. In this paper, we are particularly interested in improving the response selection part in the pipeline.

An increasing research efforts shows that the ad...
vent of Transformer (Vaswani et al., 2017) and pre-trained models (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020) has led to remarkable progress in various natural language understanding tasks, including the dialogue response selection in our interest. Built on top of BERT (Devlin et al., 2019), Cross-Encoder (Urbanek et al., 2019) has become the workhorse in response selection task for its superior performance compared to other paradigm. It jointly encodes the historical context with every single candidate response, and gives a matching score per candidate. Despite its great performance, it still remains an open problem with its obvious defects. Having such issues in mind, we propose a new paradigm, called Panoramic-Encoder, integrated with a novel Candidates Attention Mechanism (CAM), for the task. The defects and our solutions can be summarized as follows:

1. The prevalent paradigm of the response selection task is modeled as a binary classification problem. That is, a network produces a matching score for each dialogue pair, concatenated by a given context and a response. Accordingly, selecting a response from a pool with such processing causes frequent recomputation of the lengthy context, which significantly increases the inference cost. In this paper, the proposing Panoramic-Encoder re-formulates the process as a “multiple-choice” problem, where all candidates can be assessed simultaneously. As shown in Figure 1, the proposing paradigm can select an optimal response with a one-shot prediction, thereby vastly boosting the inference efficiency.

2. The existing methods only consider the relatedness between the historical context and per every response, without interacting with different candidates. Thus, it cannot separate the ground truth from some hard distractors, as suggested in Figure 2. Our Panoramic-Encoder can mitigate this issue in a subtle way. In our design, the context and all candidates are concatenated and then fed to the encoder. With the proposing attention mechanism, relationships among all candidates can be perceived, and the optimal response can be highlighted.

3. Several practical techniques have been discovered to train a powerful response selector in recent studies (Gu et al., 2020; Xu et al., 2020). However, some useful tricks, e.g., in-batch negative training, cannot be naturally integrated into the Cross-Encoders (Humeau et al., 2019). Our Panoramic-Encoder does the rescue of the compatibility issue by its novel architecture.

We conduct experiments on four benchmark datasets: PersonaChat (Zhang et al., 2018), Ubuntu Dialogue Corpus V1 (Lowe et al., 2015), Ubuntu Dialogue Corpus V2 (Lowe et al., 2017), and Douban Conversation Corpus (Wu et al., 2017). Results show our work achieves new state-of-the-art and accelerates the inference speed by a large margin. For instance, one of our models achieves an absolute improvement in $R_{10}@1$ by 2.9% with approximately $3\times$ faster inference speed on the Ubuntu Dialogue Corpus V2 dataset.

## 2 Related Work

In this section, we discuss various works that have been proposed to progress the dialogue response selection task. Besides improvements on model architectures, researchers also proposed some important training techniques such as in-batch negative training, domain post-training, etc. We will also introduce some of these important techniques in this section and briefly describe how our new method seamlessly integrate them into the new paradigm.
2.1 Model Architecture

Cross-Encoder (Urbanek et al., 2019) is the current state-of-the-art dialogue response selection method and widely used in many advanced chatbots (Bao et al., 2020). Like the typical BERT design (Devlin et al., 2019), such an architecture jointly encodes the concatenated context and response to make a prediction. Another popular architecture called Bi-Encoder (Reimers and Gurevych, 2019) encodes the context and the candidate separately, then scores the relatedness between their representations. Due to its simplicity, Bi-Encoder often serves as a baseline method when a new dataset was introduced (Lowe et al., 2015; Dinan et al., 2018). It is also computationally more efficient because candidate representations can be cached and reused once they are created. However, in generation-based chatbots, all the context and responses are newly generated, and because of that, people nowadays prefer Cross-Encoder over Bi-Encoder as the former one yields better results (Urbanek et al., 2019; Humeau et al., 2019). Cross-Encoder gets better results because it allows context and response to interact with each other in the feature space. That is to say, all the response representations are context-aware. However, this context-aware characteristic does not come for free, it requires Cross-Encoder to separately encode context for each candidate responses, which makes it much slower in inference. By encoding all the response candidates together with the context through a specifically designed attention method, our Panoramic-Encoder kills two birds with one stone. It does not only take a context-aware concept a step forward to become context-other-responses-aware, but also removes the necessity of computing context representation multiple times.

2.2 In-batch Negative Training

In contrastive learning, in-batch negative training is a standard recipe to generate representations with better uniformity and alignment (Fang et al., 2020; Gao et al., 2021). However, as stated in Humeau et al. (2019), despite the effectiveness of in-batch negative training for response selection, the Cross-Encoder architecture is problematic to recycle the in-batch negative representations because the context and the response are jointly processed. Li et al. (2021) attempt to adapt contrastive learning to this task with a specially designed strategy and obtain a significant performance gain. Our work differs from previous works in that it provides a seamless usage of in-batch negative training. Since the candidates are concatenated in the Panoramic-Encoder, it is natural to use the other labels in the same batch as negatives. Our study demonstrates that in-batch negative training is an essential technique for response selection.

2.3 Adding Speaker Change Information

Being aware of the speaker change information proves to be important for training a good model on dialogue data. There are two commonly used strategies to achieve this: adding speaker-aware embedding to the token representation and adding special tokens to segment utterances from different speakers. Wolf et al. (2019) and Wang et al. (2020) equip dialogue generation with these approaches while Lu et al. (2020) and Gu et al. (2020) verify their necessities for the response selection task. We adopt the special tokens strategy for its simplicity.

2.4 Domain Post-training

Post-training targets on improving the domain adaptation of pre-trained models in a self-supervised manner. It leverages additional domain-specific data through a second stage of pre-training. This method is compatible with all architectures since it is in an independent step. Whang et al. (2020) and Han et al. (2021) validate the usefulness of post-training on response selection. We also demonstrates that combining this method further improves the effectiveness of the Panoramic-Encoder.

2.5 Auxiliary Training Tasks

To further utilize target data, Xu et al. (2020) and Whang et al. (2021) investigate some self-supervised learning objectives such as next session prediction, utterance restoration, incoherence detection, masked language modeling, etc., as a auxiliary tasks that jointly trained with the response selection task. To keep the simplicity of our work, we only take masked language model(MLM) as our auxiliary task.

3 Method

This section first proposes a new paradigm for the dialogue response selection task. This fresh view inspires us to develop a Panoramic-Encoder architecture with three novel candidate attention mechanism. We also integrate some existing effective techniques, e.g., in-batch negative training, into our Panoramic-Encoder seamlessly.
3.1 Binary Classification vs. Multi-choice Selection

The multi-turn response selection has long been modeled as a binary classification task. Given a dialogue context $c = \{u_1, u_2, \ldots, u_N\}$, where $u_k, k = 1, \ldots, N$ denotes a single utterance from either speaker, the response selection task is required to choose an optimal response from a candidate pool, denoted by $p = \{r_1, r_2, \ldots, r_M\}$. Every candidate $r_i$ is paired with the context $c$, e.g., $m(c, r_i)$. A non-linear function is optimized to predict the value of 1 for a proper match and 0 otherwise.

To improve its effectiveness and efficiency, we propose a new paradigm for the response selection task. With the dialogue context $c = \{u_1, u_2, \ldots, u_N\}$ and a candidate pool $p = \{r_1, r_2, \ldots, r_M\}$, the selector model is trained to identify the optimal choice $r_i^*$ by fitting the objective $s(c, p) = i$. That is, our paradigm can select the globally optimal response in a one-shot inference, thereby greatly saving the inference costs. In addition, since all candidates are concatenated as input, the context can simultaneously attend to all candidates and highlight the most proper one, thus improving accuracy.

3.2 Panoramic-Encoder

The innovation of paradigm inspires this design of the Panoramic-Encoder. It exploits a pre-trained transformer encoder (Vaswani et al., 2017) as a basis. As depicted in Figure 3(b) and Figure 4, it resembles the Cross-Encoder architecture (Humeau et al., 2019) but has several crucial distinctions:

1. The candidates are concatenated and jointly encoded with the input context.
2. We reuse the positional embeddings for different candidates to comply with the length limit.
3. To incorporate speaker change information, each candidate is surrounded by $[CLS]$ and $[SEP]$ tokens, and two special $[SPK]$ tokens are used to segment the sentences from alternating speakers.
4. We develop and compare several candidates attention mechanisms that allow candidate responses to interact at different level of granularity.
We analyze three different types of candidate attention mechanisms, as exhibited in Figure 5. Type (a) is identical to the all-to-all attention in Transformers. However, it has two problems. First, it has a position confusion problem. For illustration, the first token in candidate $i$ cannot distinguish its own second token from the other candidates’ because they share the same positional embeddings. Second, attention has an averaging effect, hence too much interaction make different candidates difficult to distinguish from each other. To address this problem, we forbid explicit attention between candidates and only allow context response attention (type (b)), but they can still exchange information indirectly through common connections with the context. In third type, we further enhance the interaction on the basis of context-to-response attention by allowing the attention between [CLS] heads in responses. We study the effects of these three attention mechanisms on PersonaChat and list the results in Table 1. As can be seen, the ALL-to-ALL attention gets significantly worse results than the other two. But both Context-to-Response and CLS-to-CLS attention get similar results, which indicate that a small amount of interactions among candidates should be enough to get good performance. In the subsequent experiments, we will use context-to-response (type (b)) attention as our default setting.

In the Panoramic-Encoder, as mentioned in section 3.1, instead of assessing each response respectively, it compares all candidates simultaneously to find the global optimum in one shot. The given dialogue context $c = \{u_1, u_2, \ldots, u_N\}$ and the candidate pool $p = \{r_1, r_2, \ldots, r_M\}$ are jointly encoded to yield output representations $H$. As discussed earlier, the candidate pool in our implementation consists of the gold response and the other in-batch negative samples.

$$H = \text{encode}(c, p).$$

We then obtain an aggregated embedding $E_i$ for each candidate by averaging all token representations belonging to it in $H$. After aggregation, every $E_i$ is reduced to a single logit, which is later merged and fed into a softmax operation.

$$Y_{\text{pred}} = \text{softmax}\{w(E_1), \ldots, w(E_m)\}.$$  

A ground truth label is one-hot at the index of the only positive candidate. Then the model is optimized by minimizing the cross-entropy loss between the prediction and ground truth. We also plus an auxiliary MLM loss to the original classification objective as

$$\ell = \ell_{\text{ce}} + \ell_{\text{mlm}},$$

where $\ell_{\text{ce}}$ is defined as:

$$\ell_{\text{ce}} = \text{cross entropy}(Y_{\text{pred}}, Y_{\text{label}}).$$

### 4 Experiments

#### 4.1 Dataset

- PersonaChat (Zhang et al., 2018) is a crowdsourced dataset with two-speaker talks conditioned on their given persona, containing short descriptions of characters they will imitate in the dialogue.
The statistics of four benchmark datasets are shown in Table 2. They vary greatly in volume, language, and topic. Several metrics are used to evaluate our model following previous works. We measure $R_{10}@k$ on four benchmark datasets. Mean reciprocal rank (MRR) on PersonChat is additionally calculated to conduct comparisons. $P@1$ and mean average precision (MAP) are also employed for the Douban Conversation Corpus because it contains multiple positive candidates for a given context. We also note a significant difference in the proportion of positive and negative samples between the validation and test sets in the Douban Conversation Corpus. To alleviate this discrepancy, we also utilize the in-batch negative labels during validation to determine a more applicable checkpoint at inference time.

### 4.2 Inference Speed

One of the major improvements brought by the new paradigm is that Panoramic-Encoder has a significant advantage over the baseline in terms of efficiency. It is evidently because the Panoramic-Encoder can find the optimal response among candidates in one shot rather than rank each candidate in turn. This feature remarkably reduces the number of inferences the Panoramic-Encoder requires during evaluation. However, the concatenated candidates also requires more memory allocation when computing. Therefore, for the sake of fair comparison, we control the peak GPU memory usages of all models to the same value by assigning them different batch sizes. We run experiments on a single NVIDIA A100-SXM4-40GB with CUDA 11.1. The results in Table 3 verify that our model is able to complete inference for all test cases in Ubuntu Dialogue Corpus V2 with approximately 3X speed up.

### 4.3 Effectiveness of Each Component

As mentioned earlier, the novel architecture change in Panoramic-Encoder addresses the compatibility issue of in-batch negative training and seamlessly incorporates some other effective techniques. * represents the full effect of a Panoramic-Encoder model. Bold values are the most significant drops in performance. The last component is innovative in our work, where the response concatenation allows the application of in-batch negative training.
Table 5: Evaluation on four benchmark datasets. All results reported in the table are fine-tuned on the naive BERT-base (Devlin et al., 2019) model without any post-training. Average and standard deviation are calculated on three runs with different seeds.

<table>
<thead>
<tr>
<th>Models</th>
<th>Ubuntu Corpus V1</th>
<th>PersonaChat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1@1 R@1@2 R@1@5</td>
<td>R@2@1 MRR</td>
</tr>
<tr>
<td>BERT (Devlin et al., 2019)</td>
<td>0.781 0.890 0.980</td>
<td>0.707 0.808</td>
</tr>
<tr>
<td>SA-BERT (Gu et al., 2020)</td>
<td>0.830 0.919 0.985</td>
<td>- - -</td>
</tr>
<tr>
<td>BERT-CRA (Gu et al., 2021)</td>
<td>- - -</td>
<td>0.843 0.903</td>
</tr>
<tr>
<td>Panoramic-Encoder (Ours)</td>
<td>0.859 0.938 0.999</td>
<td>0.869 0.922</td>
</tr>
</tbody>
</table>

Table 6: Panoramic-Encoder further boosts the performance of the state-of-the-art post-trained models on Ubuntu Dialogue Corpus V1. better average performances with relatively small standard deviations in almost every single metric on PersonaChat, Ubuntu V1, and Ubuntu V2 datasets. Our models also outperform previous works in four of the six metrics on the Douban Conversation Corpus, demonstrating its overall superiority. However, on this dataset, they possess larger variances during evaluation and have weaknesses in P@1 and R@1@1. We believe one conceivable explanation is as follows: First, the discrepancy between its validation and test sets (refer to Section 4.1) makes this task more challenging. Second, the Panoramic-Encoder leverages response concatenation and in-batch negative training to help the only positive sample stay distantly from the other negative samples in the semantic space. However, the presence of multiple positive candidates at inference time (but not in training) makes it confusing to rank the top one response.

Next, we build the Panoramic-Encoder upon the most advanced post-trained models UMSBERT+ (Whang et al., 2021) and BERT-FP (Han et al., 2021) to explore the upper bound of our method’s capability. Table 6 indicates the Panoramic-Encoder can further boost their performance. Our best model achieves 0.916 in R@1@1 on the Ubuntu Dialogue Corpus V1, which is the universal highest result as far as we know. Our source code and model checkpoints will be released for reproducibility and future research. Please refer to them for more training details.

4.4 Comparison to State-of-the-art

To fully demonstrate the superiority of the Panoramic-Encoder against the other state-of-the-art methods. We first initialize our implementation with the naive BERT checkpoint provided by Huggingface¹. All the reported results in Table 5 are fine-tuned on the BERT-base model (Devlin et al., 2019) without any post-training.

As we can see, the Panoramic-Encoder achieves moderate performance gains in both metrics. As described in Section 2.2, in-batch negative training has to be applied together with the architecture change (response concatenation). Unsurprisingly, they perform as the most prominent improvement and jointly augment the R@1@1 by 6.00% and MRR by 3.41%. This innovation has also led to remarkable results in the subsequent comparisons with previous state-of-the-art models.

Table 6: Panoramic-Encoder further boosts the performance of the state-of-the-art post-trained models on Ubuntu Dialogue Corpus V1.

<table>
<thead>
<tr>
<th>Models</th>
<th>Ubuntu Dialogue Corpus V1</th>
<th>Douban Conversation Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@1@1 R@1@2 R@1@5 MAP MRR</td>
</tr>
<tr>
<td>BERT (Devlin et al., 2019)</td>
<td>0.808 0.897 0.975 0.591 0.633</td>
<td>0.454 0.280 0.470 0.828</td>
</tr>
<tr>
<td>SA-BERT (Gu et al., 2020)</td>
<td>0.855 0.928 0.983 0.619 0.659</td>
<td>0.496 0.313 0.481 0.847</td>
</tr>
<tr>
<td>BERT-SL (Xu et al., 2020)</td>
<td>0.884 0.946 0.990 - - - - -</td>
<td>- - - - - - - - -</td>
</tr>
<tr>
<td>UMSBERT (Whang et al., 2021)</td>
<td>0.843 0.920 0.982 0.597 0.639</td>
<td>0.466 0.285 0.471 0.829</td>
</tr>
<tr>
<td>BERT+FGC (Li et al., 2021)</td>
<td>0.829 0.910 0.980 0.614 0.653</td>
<td>0.495 0.312 0.495 0.850</td>
</tr>
<tr>
<td>Panoramic-Encoder (Ours)</td>
<td>0.886 0.946 0.989 0.622 0.662</td>
<td>0.481 0.303 0.514 0.852</td>
</tr>
</tbody>
</table>

¹https://huggingface.co/models
A: so now i’m going to try it with the snapshot version of the PPA instead and see if it crashes again.
B: Are you looking at your computer’s CPU temperature?
A: no but i havent ever had any trouble with temp before .... where do i go to see its temp typervenche ?
Cross: i do not know. Panoramic: That depends on your hardware and kernel. The Ubuntu guys can help you with that.

A: I’m trying to backup a computer using gsync.
A: There is one file in a directory I’ve excluded, that I like to backup.
A: I listed that file and it’s path in the include file, but it doesn’t over ride the exclude file, is it possibly to do this?
B: dont exclude it..?
Cross: cheers Panoramic: example?

A: nobody know Jin.. :(
B: i guess not. try contacting the programs developers. you can find their contact info in the software centre
Cross: Thanks Panoramic: thx unfortunately the program isn’t installed from the repositories

A: any vim experts around?
B: you may want a vim chatroom
Cross: cheers. Panoramic: also google is your friend :-)P

A: i am stuck on the loading screen for xubuntu, is there a hotkey to leave it to see what it does in text?
B: ctrl+f1
Cross: ok Panoramic: sometimes it just heats up, it’s not consistent but it suggests a fan/controllr type issue

Table 7: Cherry-picked examples from the Ubuntu Corpus V2 for comparing Cross- and Panoramic- Encoder

5 Analysis and Discussion
This section provides a qualitative analysis to understand the Panoramic-Encoder further. We also discuss some limitations of our design and feasible solutions to address them.

5.1 Qualitative Analysis
We have cherry-picked some test cases from the Ubuntu Corpus V2 to analyze the advantages of our work over the Cross-Encoder. The best Cross-Encoder implementation, as presented in Section 4.3, is used for comparison, which has no response concatenation and in-batch negative training but with all other techniques. Results in Table 7 suggest that Panoramic-Encoder is able to select very diverse and coherent responses. In contrast, even though some results of the Cross-Encoder are not logically problematic, they are very generic and clearly inferior to ours.

5.2 Too Many Candidates to Fit
As described earlier, the Panoramic-Encoder is originally designed for generation-based dialogue systems. Such a task has a very small candidates pool and the length of concatenated responses is typically no longer or comparable to that of a given context. Our method can be applied to retrieval-based tasks as well. However, if there are too many candidates to fit, memory usages could limit its capability due to the $O(n^2)$ complexity of the attention mechanism. In the worst case, where only a single candidate can be processed at a time, the Panoramic-Encoder degenerates into a baseline method.

We would suggest a solution to avoid this limitation: (i) Dividing candidates into multiple groups with exercisable sizes. (ii) Applying the Panoramic-Encoder to identify the best from each group. (iii) Repeating the procedures hierarchically on previous winners if necessary, until the global optimum is determined. Moreover, giving candidates a preliminary screening is helpful to accelerate the whole process.

6 Conclusion
In this paper, we propose a new paradigm for the dialogue response selection task. To this end, we present the Panoramic-Encoder architecture that integrated with multiple novel candidates attention mechanisms. The proposed method simultaneously processes all candidate responses to select the global optimum in one-shot inference. Also, the parallel computation fashion in our paradigm allows using the in-batch negative training seamlessly, which again boosts its performance. By incorporating other common practices in training, our method pushes state-of-the-art results across four benchmarks, with significantly faster inference speed. Thorough empirical results also show the superiority of our proposal.
References


