Personalization in BERT with Adapter Modules and Topic Modelling

Marco Braga[1], Alessandro Raganato[1] and Gabriella Pasi[1]

[1]Università degli Studi di Milano Bicocca, Dipartimento di Informatica, Sistemitstica e Comunicazione DISCo, Viale Sarca 336, Milano

Abstract

As a result of the widespread use of intelligent assistants, personalization in dialogue systems has become a hot topic in both research and industry. Typically, training such systems is computationally expensive, especially when using recent large language models. To address this challenge, we develop an approach to personalize dialogue systems using adapter layers and topic modelling. Our implementation enables the model to incorporate user-specific information, achieving promising results by training only a small fraction of parameters.

Keywords

Retrieval based chatbot, Personalization, Adapters, Topic Modelling

1. Introduction

As smart assistants become increasingly common in our daily lives, developing dialogue systems is attracting considerable attention. Dialogue systems are being used in various settings, such as customer services, e-commerce and healthcare, where personalized interactions can greatly impact performances. Personalization can be seen as the ability of a system to customize its responses based on a user's past behaviour and on the context of the ongoing conversation. Typically, such systems use pretrained large language model such as BERT [1], GPT [2] and T5 [3]. These models are developed following the pretrain-and-then-finetune paradigm, which involves pretraining a neural model on very large amounts of raw text using a language model objective, and then further fine-tuning it on task-specific data. Over the years, different approaches to developing dialogue systems have been explored; for example, Han et al. [4] fine-tune BERT for dialogue response selection by applying a mechanism to let the model understand the coherence between subsequent turns in a conversation, while Gu et al. [5] personalize BERT adding a *speaker embedding* to token representation, which includes information about the user's identity, to name a few. However, these methods are usually computationally expensive as they involve training all neural network parameters. To address this drawback, we explore the use of adapters [6], which are modules added between the layers of a pre-trained Transformer, to inject personal information into a network. Our proposed implementation, which we call User Adapter, receives a user embedding vector as external input allowing the encoding of personal information into the model, resulting in improved performances and accuracy compared to baseline adapters.

IIR2023: 13th Italian Information Retrieval Workshop, 8th - 9th June 2023, Pisa, Italy

CEUR Workshop Proceedings (CEUR-WS.org)

Dataset	# Samples	Pairs positive:negative	# Unique users	# User both in Dataset and Training
Training	1M	1:1	202768	202768
Validation	196k	1:9	9374	4106
Test	182k	1:9	9080	3504

Table 1
Statistics of the Ubuntu Dialogue Corpus (UDC) [7].

In the following sections, we first give an overview of the benchmark used in this work, then we describe the details of two extensions we propose, which are categorized into non-personalized and personalized models. The non-personalized models aim to adapt BERT to a dialogue system task; personalized models introduce user-related information into BERT, obtained by applying topic modelling, through adapter layers. Finally, we present the results of our evaluation with the conclusions and future work.

2. Ubuntu Dialogue Corpus

Dialogue systems can be broadly grouped into two main categories: generation-based and retrieval-based. Generation-based methods aim to create new text and generate responses within a conversation, typically using an encoder-decoder framework [8]. Retrieval-based methods, on the other hand, focus on selecting responses for single-turn conversations and are often referred to as Next Utterance Classification. This approach involves predicting whether a given utterance follows the previous dialogue or not, and can be seen as a binary classification task. In this work, we focus on the latter setting. One of the most popular benchmark data sets for retrieval-based dialogue systems is the Ubuntu Dialogue Corpus (UDC) [7], which consists of dyadic dialogues extracted from the Ubuntu support platform. Statistics about UDC are in Table 1. Each sample of the dataset includes a dialogue history (the context), the next possible utterance, the user who wrote it, and a flag indicating whether the utterance is the correct sentence that follows the dialogue history. The context consists of multiple turns of dialogue, where each turn represents a series of utterances exchanged between the participants in the conversation. Given a dialogue, the task consists of selecting the correct next utterance across 10 possible candidates.

3. The Proposed Approach

In this section, we present the models we consider and evaluate, divided into non-personalized baselines (Section 3.1), and personalized adapter models (Section 3.2). Moreover, in Section 3.3 we describe how we use topic modelling to obtain user embeddings, which are used to incorporate personal information (i.e., user-related information) into a model.

3.1. Baselines

We use different baseline models based on BERT without any personalization component: Fine-tuning (FT) and Task Adapter (TA). The first model, FT, is a fine-tuned version of BERT base (12-layer, 768-hidden, 12-heads, 110M parameters) [1] on the Ubuntu Dialogue Corpus, which requires retraining all 110 million parameters of BERT. TA models, instead, are based on the adapter architecture, which is a lightweight approach to adapting pretrained language models to a specific task. As described in [6] and shown in Figure 1, an adapter usually consists of two feedforward layers: the first layer performs a down-projection to a lower intermediate dimension, followed by a non-linear activation function (in our work RELU). Next, there is an up-projection to the original dimension. The output of an adapter layer is the sum of the input of the layer itself and the output of the up-projection layer. We test the considered models by using different intermediate dimensions, which are the only hyperparameters that control the adapter capacity, i.e. 128 and 256. It can be argued that training also the word embeddings layer may lead to better performance overall, but with an additional cost of training more parameters. Thus, we include this setting in our experiments. To make a comparison, all these experiments follow the parameters reported in the original study [1], i.e. batch size of 32 and 3 epochs. To ensure the convergence of models, as suggested by [6], the adapter weights are initialized with a normal distribution having mean equal to zero and standard deviation of 0.02. This guarantees that, during the initial training steps, adapter layers behave similarly to an identity function.

3.2. Personalized Models

We propose three extensions of the previously described adapter layers to incorporate user-related information into BERT: i) Topic User Adapter (TUA), ii) Concatenation Topic User Adapter (CTUA), and iii) Gate-based Topic User Adapter (GTUA). In our implementation, each method receives an external vector that represents the user who wrote the utterance in input; we describe how to encode the user vector in Section 3.3.

Once we have obtained the user vector, we give it as input to a new down projection, making it comparable to the dimensions of the BERT projected vector. Then, in TUA, we apply a sum operation between the user and BERT vectors, resulting in a new vector that captures the combined information. Then a RELU function is applied to obtain the final output. This architecture is shown in Figure 1. TUA requires training only 0.1% more parameters than adapters without personalization. In CTUA, instead, we combine the two vectors by appending one to the end of the other, resulting in a vector whose length is twice the length of the projected vectors. Finally, in GTUA, we use a gating function to weigh the user vector representation. More formally, the output of the Gated Adapter is defined as $Relu(\sigma(Wb + W'E)W'E + Wb)$ where $b \in \mathbb{R}^d$ is the input to the adapter layer, $d = 768$ the intermediate dimension of BERT, $E \in \mathbb{R}^t$ our user embedding with t that corresponds to the number of topics, $W \in \mathbb{R}^{dxd'}$ and $W' \in \mathbb{R}^{txd'}$ are matrices of BERT and user down projections respectively, d' is the intermediate dimension of the adapter and σ is a sigmoid function. Since σ takes values between 0 and 1, if its value is high and near to one, user embeddings are contributing to the final predictions, otherwise, if the value is near to zero, user embeddings have a small impact.

[1] https://huggingface.co/bert-base-uncased

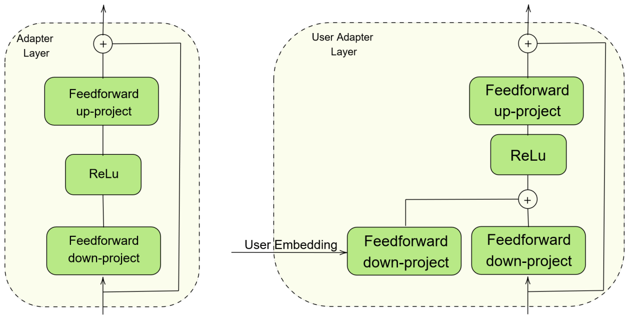

Figure 1: Different kinds of adapters: *left* adapter as defined in [6], *right* Our new User Adapter

3.3. User Embeddings

We employ topic modelling to generate user vectors, specifically, we use the Latent Dirichlet Allocation (LDA) [9] implementation from `scikit-learn` [10]. Let's denote by D_U the document containing all dialogues in which a single user U engages. For each document D_U and topic T, LDA returns a value $V(D_U, T)$ between zero and one such that $\sum_{T\in\{1,...,n\}} V(D_U, T) = 1$, where n is the number of topics. $V(D_U, T)$ represents how much the topic T characterizes dialogues in which the user U participates. To optimize the performances of the LDA model, we test different hyperparameters configurations in a small development set to find the best number of topics between $10, 15, 20, 25, 30$. Our analysis reveals that the best number of topics is 10, which results in the best perplexity score. Our method produces a user embedding that represents the user's level of interest in different topics: each entry $i \in \{1, ..., n\}$ of the user U vector corresponds to the value $V(D_U, i)$ of the user's association with the topic i. For users with a limited past conversation history, topic modelling assigns a vector whose entries are all equal to $1/n$. We apply the same strategy to users in validation and test sets with no prior context, as it is impossible to obtain their user embeddings without any previous conversations.

4. Evaluation

We use the same evaluation metrics as in previous works [4, 5]: during testing, each model has to select the k best answers from a set of $n = 10$ candidates with the same history context. There is only one correct answer in the set of candidates. Then we calculate the Recall values $R@k$ of the true positive utterances among the k selected responses. The value of k can be 1, 2 or 5. In Table 2, we report the performances of our extended models. Even without personalization, fine-tuned remains the best model. This result is not unexpected: using adapters for transfer learning can decrease the performance of fine-tuning. To our knowledge, no-one has tried before to employ adapters in the next utterance classification task: we learn that our best Task Adapter module decreases the performance of fine-tuning by 4% on $R@1$ and $R@2$, and by 1% on $R@5$. Compared to the GLUE [11] benchmark dataset, as reported by Houlsby et al. [6], adapter layers attain within 0.4% of the performances of the fine-tuned model. Thus, adapters perform worse on a next utterance classification task than on a Natural Language Understanding

Model	**Train Emb.**	**Trained Params.**	**R@1**	**R@2**	**R@5**
		Baselines			
Fine-Tuning	Yes	110 M	**0.761**	**0.877**	**0.975**
Task Adapter$_{128}$	Yes	28 M	0.667	0.814	0.952
Task Adapter$_{256}$	Yes	32 M	0.707	0.841	0.961
Task Adapter$_{128}$	No	4.7 M	0.724	0.845	0.961
Task Adapter$_{256}$	No	9.5 M	0.729	0.842	0.963
		Personalized			
Topic User Adapter$_{256}$	No	9.56 M	**0.751**	**0.868**	**0.970**
Concatenation Topic User Adapter$_{256}$	No	14.28 M	0.708	0.836	0.960
Gate-based Topic User Adapter$_{256}$	No	9.55 M	0.716	0.850	0.966

Table 2

Results for all the proposed models. Subscripts are the dimensions of the intermediate layer of adapters.

task. From our results, the best intermediate size for the adapter is 256, which outperforms dimension 128 by 1.7% on average. Training the word embeddings layer performs worse than not training it, likely due to overfitting on train set. The performances of the personalized models are comparable, the best model is TUA, which decreases performances of fine-tuning by 1.3%, 1.0% and 0.51% on $R@1$, $R@2$ and $R@5$ respectively, which is better than using only Task Adapters. By using sum operation, when the user has no prior information, the output vector of each adapter layer is like a non-personalized one. This approach allows for the integration of personalized and non-personalized models, resulting in superior performances. Personalization does not surpass fine-tuning performances, but it can still obtain better results than training adapter without personalization by adding only 0.1% of total BERT weights.

5. Conclusions and Future Work

In this paper, we achieve promising results in the Next Utterance Classification task by adding personalization layers, surpassing the performances of using only task-specific adapters training only 0.1% more parameters. However, this approach does not outperform fine-tuning, which remains the most effective but also the most computationally expensive method. Our objective is to apply personalized adapters to various tasks, including information retrieval and news recommendation, which can benefit from personalization. Therefore, we aim to propose new personalization techniques based on the attention mechanism rather than topic modelling. By incorporating an attention mechanism in the adapter layer, we can represent the entire history of user conversations through a Long Transformer embedding and give it as input to the attention mechanism, allowing the model to determine the impact of past dialogues for classification. Our goal is to improve fine-tuning performances with personalization without having to retrain all parameters of a model, which is computationally expensive with Transformer architectures.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[2] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training. (2018).

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer., The Journal of Machine Learning Research (2020) 5485–5551.

[4] J. Han, T. Hong, B. Kim, Y. Ko, J. Seo, Fine-grained post-training for improving retrieval-based dialogue systems., Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2021) 1549–1558.

[5] J.-C. Gu, T. Li, Q. Liu, Z.-H. Ling, Z. Su, S. Wei, X. Zhu, Speaker-aware bert for multi-turn response selection in retrieval-based chatbots, Proceedings of the 29th ACM International Conference on Information and Knowledge Management (2020). URL: http://doi.acm.org/10.1145/3340531.3412330.

[6] N. Houlsby, A. Giurgiu, S. Jastrzebski, Q. d. L. Bruna Morrone, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp., International Conference on Machine Learning (2019) 2790–2799.

[7] R. Lowe, N. Pow, I. Serban, J. Pineau, 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, Association for Computational Linguistics, Prague, Czech Republic, 2015, pp. 285–294. URL: https://aclanthology.org/W15-4640. doi:10.18653/v1/W15-4640.

[8] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks., Advances in neural information processing systems, 27 (2014).

[9] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation., Journal of machine Learning research (2003) 993–1022.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Glue: A multi-task benchmark and analysis platform for natural language understanding., Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (2018).