Integrating Task Specific Information into Pretrained Language Models for Low Resource Fine Tuning

Rui Wang[1]* **Shijing Si**[1]* **Guoyin Wang**[1,2] **Lei Zhang**[3] **Lawrence Carin**[1] **Ricardo Henao**[1]
[1]Duke University [2]Amazon Alexa AI [3]Fidelity Investments
rui.wang16@duke.edu

Abstract

Pretrained Language Models (PLMs) have improved the performance of natural language understanding in recent years. Such models are pretrained on large corpora, which encode the general prior knowledge of natural languages but are agnostic to information characteristic of downstream tasks. This often results in overfitting when fine-tuned with low resource datasets where task-specific information is limited. In this paper, we integrate label information as a task-specific prior into the self-attention component of pretrained BERT models. Experiments on several benchmarks and real-word datasets suggest that the proposed approach can largely improve the performance of pretrained models when fine-tuning with small datasets. The code repository is released in https://github.com/RayWangWR/BERT_label_embedding.

1 Introduction

Recently, Pretrained Language Models (PLMs) (Devlin et al., 2018; Radford et al., 2019) have yield significant progress on various natural language processing (NLP) tasks, *e.g.*, neural language understanding, text generation, *etc*. Existing PLMs are usually pretrained in a task-agnostic manner, in which the model is expected to capture the general knowledge of natural language from a large corpus, independent of downstream-specific information. This is not a problem when data is abundant in the downstream dataset, in which case, the model can effectively extract task-specific information during fine-tuning. However, in real scenarios, data may be difficult to collect and labeling is usually expensive. We show that PLMs pretrained with general knowledge can overfit without enough guidance from the task-specific information, resulting in degraded performance during testing.

A clear-cut solution to this problem is to focus more on samples that are more relevant to the target task during pretraining. However, this requires a task-specific pretraining, which in most cases is computational or time prohibitive. Another approach is to pretrain on an auxiliary dataset before fine-tuning on the target task (Phang et al., 2018). Such method requires the availability of an appropriate auxiliary datasets. Unfortunately, in some cases it may negatively impact the downstream transfer (Wang et al., 2018a). Label embeddings (Akata et al., 2015) can be regarded as a feature-based definition of a classification task, in which detailed information of the task is encoded. One natural question is whether we can combine the general knowledge in a PLM and the task-specific characterization contained within label embeddings for better fine-tuning on low-resource tasks.

In this paper, we propose to utilize the label embeddings as a task-specific prior, complementary to the general prior already encoded during pretraining. We learn and integrate these label embeddings into BERT models (Devlin et al., 2018) to regularize its self-attention modules, so the task-irrelevant tokens or patterns can be readily filtered out, while the task-specific information can be enhanced during fine-tuning. Such a modification is compatible with any PLM built upon self-attention and will not degrade the original pretrained structure.

In order to validate the performance of our approach in a real-world setting, we collected two text classification datasets from the online patient portal of a large academic health system, each with a few thousand sequences. These are the first datasets for automatic patient message triage, which constitute an important problem in the field of clinical data analysis. Experimental results show that our approach significantly improves the performance of fine-tuning on low-resource datasets, *e.g.*, those consisting of only several thousand data samples.

*These authors contributed equally to this work

2 Related Work

Label embeddings have been previously leveraged for image classification (Akata et al., 2015), multi-modal learning between images and text (Kiros et al., 2014), text recognition in images (Rodriguez-Serrano and Perronnin, 2015), zero-shot learning (Li et al., 2015; Ma et al., 2016) and text classification (Zhang et al., 2017). Notably, LEAM (Wang et al., 2018b) jointly embeds words (tokens) and labels in a common latent space as a means to improve the performance on general text classification tasks. Further, Moreo et al. (2019) concatenates label embedding with word embeddings. However, this approach cannot be directly implemented into PLMs since the new (concatenated) embedding is not compatible with the pretrained parameters. We integrate label embeddings into the self-attention of BERT models, so the attention can be regularized to better focus on task-relevant information.

3 Methods

3.1 The BERT Model

The encoder of BERT and other popular PLMs are built upon the transformer architecture, which is composed of multiple layers of multi-head self-attention and position-wise feed-forward layers.

Multi-head Self-attention The multi-head self-attention is an ensemble of multiple single-head self-attention modules. Let $X \in \mathbb{R}^{L\times D}$ be the embedding matrix of the input sequence with length L. For each single head, the input sequence is first mapped into the key, query and value triplet, denoted as,

$$K = XW_K,\ \ Q = XW_Q,\ \ V = XW_V, \quad (1)$$

where $\{W_K, W_Q, W_V\} \in \mathbb{R}^{D\times d}$ are projection matrices. The self-attention can be formalized as

$$A = \frac{QK^T}{\sqrt{d}} \in \mathbb{R}^{L\times L}, \quad (2)$$

$$H_i = \text{softmax}(A)V \in \mathbb{R}^{L\times D}, \quad (3)$$

where $i = 1, \ldots, h$, h is the number of heads, $\text{softmax}(\cdot)$ is the row-wise softmax function and d is the head dimension. A is the attention score matrix representing the compatibility between Q and K. The multi-head self-attention is defined by concatenating and projecting $\{H_i\}_{i=1}^h$, the representation of each head, into $\hat{H} \in \mathbb{R}^{L\times D}$.

Positional-wise Feed Forward Layer After self-attention, a fully connected network is applied on each token representation x using

$$\text{FFN}(x) = \max(0, \max(0, xW_1 + b_1)W_2 + b_2),$$

which consists of two linear transformations and ReLU activations.

In BERT, the input sequence starts with a $[CLS]$ token, whose hidden state will be extracted as the sequence representation for classification. Let $CE(\cdot,\cdot)$ be the cross-entropy loss, $C(\cdot)$ be the final classifier and $\text{enc}(\cdot)$ be the encoder consisting of a stack of transformer layers. The classification loss can be written as,

$$L_c = \mathbb{E}_{(X,y)\sim\mathcal{D}}[CE(C(\text{enc}(X)_{[CLS]}), y)] \quad (4)$$

where $\text{enc}(X)_{[CLS]}$ is the representation of $[CLS]$ after encoding, y is the classification label and $\mathcal{D}$ is a dataset.

In the context of graph embeddings (Kipf and Welling, 2016), the $[CLS]$ token acts as a super node that connects to all other tokens (nodes) and aggregates global information during self-attention (convolution). After training, the embedding of the $[CLS]$ token should contain the task-specific information, so that it can mostly attend to task relevant information in self-attention during inference. However, embeddings of the PLMs are pretrained agnostic to downstream tasks. When fine-tuning with low-resource datasets where label information is scarce, a single $[CLS]$ token may not capture enough task specific information, resulting in model overfitting to task irrelevant tokens or patterns in the input sequences.

3.2 Integrating Label Embedding into Self-Attention

In this paper, we propose to leverage label embeddings to optimize the self-attention modules, so the model can better focus on task-relevant information when fine-tuned with small datasets.

We reformulate the representations in (1) as $\{K_w, Q_w, V_w\}$ by replacing X with block matrix $X_w = [X_{CLS}, X]$, where $X_{CLS} \in R^{1\times D}$ and $X \in R^{(L-1)\times D}$ represent the embeddings of $[CLS]$ and the other tokens in the sequence, respectively. The attention score matrix can be rewritten as,

$$A = \frac{1}{\sqrt{d}} \begin{bmatrix} Q_{[CLS]}K_{[CLS]}^T & Q_{[CLS]}K^T \\ QK_{[CLS]}^T & QK^T \end{bmatrix}. \quad (5)$$

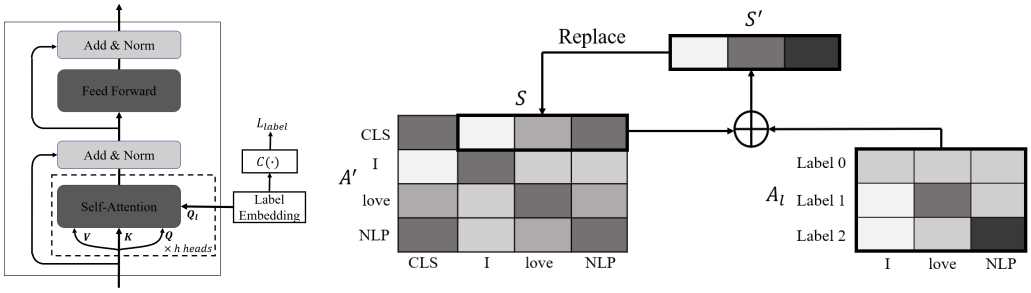


Figure 1: (a) Incorporating label embeddings into multi-head self-attention. $C(\cdot)$ is the classifier for the BERT model. (b) Modifying self-attention scores with label embeddings. $\bigoplus$ indicates row concatenation.

We denote the cross-attention between the $[CLS]$ token and all the other input tokens as $S \triangleq Q^T_{[CLS]}K^T \in \mathbb{R}^{1\times(L-1)}$. Let $X_l \in \mathbb{R}^{M\times D}$ be the label embedding matrix, where M is the number of classes. We first compute the cross attention between X_l and X as

$$A_l = \frac{Q_l K^T}{\sqrt{d}},\ Q_l = X_l W_Q, \tag{6}$$

where X_l is encoded in to Q_l with the same mapping matrix W_Q as in (1). Then, we compute a modified cross-attention row vector S' by concatenating S and A_l by row and keeping the maximum value of each column,

$$S' = \max([S; A_l]) \in \mathbb{R}^{1\times L}. \tag{7}$$

As a result, S' represents the maximum attention score of a input token with both $[CLS]$ and the label embeddings. A new attention score matrix A' can be obtained by replacing S with S' in (5),

$$A' = \frac{1}{\sqrt{d}} \begin{bmatrix} Q_{[CLS]}K^T_{[CLS]} & S' \\ QK^T_{[CLS]} & QK^T \end{bmatrix}. \tag{8}$$

In (8), when a token is highly relevant to one of the labels, it will result in a larger attention score in S', thus the $[CLS]$ embedding will be less affected by irrelevant information in the sequence, unlike (2) where only attention from the current $[CLS]$ embedding is considered. The proposed attention layer is shown in Figure 1(b). The attention score matrix A in (2) is replaced as A' in (8). All other components are the same as the original layers in BERT as in (1)–(3.1).

We share the same label embedding X_l for all the layers. The label embedding is adapted on each layer via W_Q in the multi-head attention module. As shown in Figure 1a, we also feed X_l into the final classifier $C(\cdot)$, so the label embeddings can be classified into their corresponding classes. The final loss for classification is then

$$L_{label} = \sum_{i=1}^{M} CE(C(X^i_l), i), \tag{9}$$

$$L_{final} = L_c + \lambda L_{label}. \tag{10}$$

where X^i_l is the i-th label embedding, λ is a trade-off parameter between the regularization on label embeddings and the original classification loss.

The label embeddings can be initialized randomly or by the pretrained embeddings of relevant keywords. When the label is not identified by keywords, *e.g.*, in sentence entailment tasks, their embeddings can be initialized with the representations of $[CLS]$, averaged over samples from the same class. All other parameters can be initialized from the pretrained BERT. This modification can be adapted to any PLM with self-attention modules.

4 Experiments

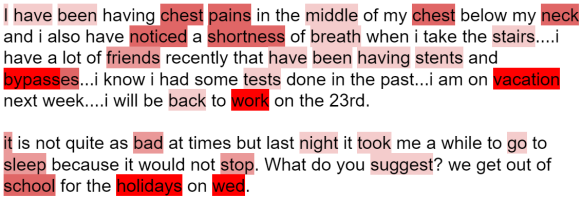
We focus on fine-tuning with small datasets. We integrate label embeddings into the pretrained (Bio)BERT models, and fine-tune on various classification benchmarks as well as two real-world clinical datasets that we collected from the online patient portal of a large academic health system.

4.1 Public Benchmarks

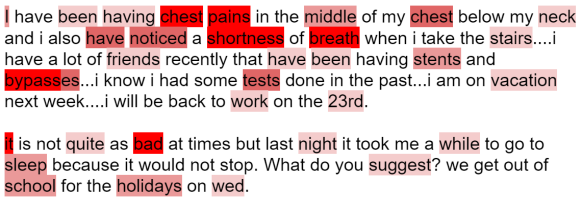
Table 1 shows the results of integrating label embedding into the pretrained bert-based-uncased model on 9 public classification benchmarks of various sizes. We find that our method improves the results from BERT on small datasets, *e.g*, WNLI, MRPC, CoLA, *etc*, which typically have only several thousand data samples available for fine-tuning. This shows that the BERT model, which is pretrained with task-agnostic objectives, is more likely

Table 1: Results on public benchmarks.

Method	TREC (5.5k)	WNLI (0.6k)	RTE (2.5k)	MRPC (3.7k)	CoLA (8.5k)	IMDB (25k)	SST-2 (67k)	MNLI-M/MM (393k)	QQP (364k)	Avg
BERT (Devlin et al., 2018)	97.00	55.11	63.90	87.29	54.47	92.36	92.32	**84.38/ 84.87**	87.53	79.92
Our Method	**97.40**	**57.75**	**66.43**	**89.48**	**56.26**	**92.43**	**92.58**	84.12/ 84.62	**87.84**	**80.89**



(a) Attention from the BioBERT.



(b) Attention from our method.

Figure 2: Examples of the attention from the $[CLS]$ token in the final attention layer. The sequences are sampled from the Message-urgency dataset. Red color indicates higher attention score. It can be shown that our method can better focus on keywords, *e.g.*, 'chest', 'bad' and 'stairs', which are more likely to ocurr on urgent requests. Alternatively, BioBERT fine-tuned on such a small dataset tends to overfit to task-irrelevant words, such as 'holiday', 'school', 'tests', *etc.*

to overfit when there is limited task-specific information during fine-tuning. However, our method produces comparable results on larger datasets such as MNLI and QQP. This is consistent with the study in Lazar (2003) where additional priors are less useful when the size of dataset grows larger. These results suggest that our method is more suitable for fine-tuning with smaller amounts of data, and that our approach to injecting the label information is at least not detrimental to the original pretrained model. This supports the intuition of combining the pretrained general knowledge and the task-specific information for better fine-tuning with small datasets.

We note that label information can improve the results on many tasks of neural language inference, *e.g.*, WMLI and QQP, where classes are not identified by keywords, but rather certain patterns in the input sentence pair. This may be because the self-attention will encode these input patterns into intermediate tokens, which act as pseudo keywords that can be emphasized by the attention from label embeddings.

Table 2: Results on our healthcare datasets. Values are shown as F1/Precision/Recall.

Dataset	Message-urgency (1.7k)	Acknowledgment (1.6k)
BERT (Devlin et al., 2018)	0.761/0.762/0.761	0.980/0.976/0.984
BioBERT (Lee et al., 2020)	0.764/0.774/0.758	0.985/0.990/0.980
Our Method	**0.789**/0.784/0.797	**0.990**/0.993/0.987

4.2 Patient Message Triage

We further evaluate the proposed approach in real-world scenarios of patient message classification. This is a task motivated by the increasing popularity of online patient portals. Most of the patient messages generated from the portal are non-urgent, while the doctors are expected to focus on the urgent requests, which amount to only a small portion (about 10%) of all messages. As a result, the heath providers will have to spend considerable time just identifying urgent messages, thus being less efficient at emergency responses. We obtain two healthcare datasets –*Message-urgency* and *Acknowledgment*– from a large academic health system online portal. Detailed description of these two datasets can be found in Appendix A.

We employ our method on the BioBERT pretrained model (Lee et al., 2020), which has the same architecture as BERT but further pretrained on the clinical corpora. Results are shown in Table 2. Our model improves on all the baselines in terms of F1 score, which validates the usefulness of the proposed method for low-resource fine-tuning in the real scenarios.

5 Conclusion

We propose to integrate task specific information into PLMs that are pretrained with task-agnostic objectives. To do this, we leverage label embeddings to regularize the self-attention in PLMs. Results on public benchmarks and real-world datasets suggest that our method can effectively improve the results for low resource fine-tuning.

References

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2015. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. 2014. Multimodal neural language models. In *International conference on machine learning*, pages 595–603.

Nicole A Lazar. 2003. Bayesian empirical likelihood. *Biometrika*, 90(2):319–326.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Xirong Li, Shuai Liao, Weiyu Lan, Xiaoyong Du, and Gang Yang. 2015. Zero-shot image tagging by hierarchical semantic embedding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 879–882.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180.

Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2019. Word-class embeddings for multiclass text classification. *arXiv preprint arXiv:1911.11506*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jose Antonio Rodriguez-Serrano and Florent C Perronnin. 2015. Label-embedding for text recognition. US Patent 9,008,429.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, et al. 2018a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. *arXiv preprint arXiv:1812.10860*.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018b. Joint embedding of words and labels for text classification. In *ACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2017. Multi-task label embedding for text classification. *arXiv preprint arXiv:1710.07210*.

A Description of healthcare datasets

In this work, we utilized 1,756 web portal messages generated from 10/2014 to 08/2018 by adult patients ($>$ 18 years old) of a large academic medical center. The Electronic Health Record (EHR) system (Epic Verona, WI, USA) with associated patient portal (MyChart) was the source of all patient messages. A custom-built Application Programming Interface (API) securely made available the portal messages from the EHR enterprise data warehouse into a highly protected virtual network space offered by the medical center. Approved users were allowed access to work with the identifiable

Label	Count	Typical Example
Non-urgent	631	That would be awesome... thank you.
Medium	955	Dr. [name]. All seems well now. I am at home resting. My wife and I have a trip planned to Maryland this week beginning on Wednesday. We can fly, drive or stay home if I should not travel. Are there any reasons that I should not fly.
Urgent	170	I have continued having chest pain shortness of breath since waking. Please tell me what to do. I have tried in hailers am going to try nebulizers. I just feel extremely tight in my chest.

Table 3: Typical examples of patient messages to providers grouped by urgency. These are examples of the message urgency dataset used in the experiments.

Label	Count	Typical Example
1	1123	Thank you. Have a good day.
0	566	I have continued having chest pain shortness of breath since waking. Please let me know what to do.

Table 4: Typical examples of patient messages to providers. Label 1 for messages being pure acknowledgment, while 0 for non-trivial messages.

protected health information. These messages included free, unstructured plain text sent by patients to their healthcare team. Responses and messages sent from the clinician or health system to the patient were excluded from the analysis.

A.1 Message-urgency dataset

In message-urgency dataset, portal messages were manually labeled by experienced sub-specialty (cardiology) clinicians into three levels of priority: non-urgent, medium and urgent. Non-urgent labels include notes of appreciation (*e.g.*, thank you). The Medium urgency class contains messages that could be reasonably responded to in 1-3 days. Urgent messages are those requiring an immediate phone call to the patient by the clinician. Conditions suggesting acute myocardial infarction, exacerbation of heart failure respiratory distress or possible stroke were labeled as urgent and would be inappropriate for an asynchronous patient portal.

A.2 Acknowledgment dataset

This acknowledgment dataset is randomly selected from patient's responses to the hospital. A significant portion of these messages is purely acknowledgment, like 'Thank you'. It would be helpful if this type of messages can be filtered out, so that hospital staff can focus on non-trivial messages. A doctor and a nurse labelled and validated this dataset.

B Implementation Details

For all the experiments, we use finetune the pretrained model for 3 epoches with learning rate 2e-5 and batch size 32. We use the Adam training algorithm. λ is generally set to 3. We set warm up steps as 10 percent of the total training steps. We do not apply weight decay and the norm of all the gradients are clipped by 1. Experiments on the public benchmarks are run on a TITAN X (Pascal) 1080 gpu. The healthcare experiment are run on the CPU in a secured virtual machine system.