Does QA-based intermediate training help fine-tuning language models for text classification?

Shiwei Zhang
School of Computing Technologies
RMIT University, Australia
dr.shiwei.zhang@gmail.com

Xiuzhen Zhang*
School of Computing Technologies
RMIT University, Australia
xiuzhen.zhang@rmit.edu.au

Abstract

Fine-tuning pre-trained language models for downstream tasks has become a norm for NLP. Recently it is found that intermediate training based on high-level inference tasks such as Question Answering (QA) can improve the performance of some language models for target tasks. However it is not clear if intermediate training generally benefits various language models. In this paper, using the SQuAD-2.0 QA task for intermediate training for target text classification tasks, we experimented on eight tasks for single-sequence classification and eight tasks for sequence-pair classification using two base and two compact language models. Our experiments show that QA-based intermediate training generates varying transfer performance across different language models, except for similar QA tasks.

1 Introduction

The framework of fine-tuning pre-trained Language models (LMs), especially transformer-based LMs, for downstream tasks has shown state-of-the-art performance on many natural language processing (NLP) tasks (Devlin et al., 2019; Raffel et al., 2020). It is believed that the pre-training stage leads LMs to develop general-purpose abilities and knowledge that can then be transferred to downstream tasks (Raffel et al., 2020).

To further improve the performance of pre-trained LMs on target tasks, two novel training approaches have been recently researched, namely further pre-training and intermediate training. A further pre-training stage for LMs (Gururangan et al., 2020) is a stage between pre-training and fine-tuning, which further pre-trains LMs on an extra dataset using unsupervised objectives. It has been found that further pre-training LM on the target domain (domain-adaptive pre-training) leads to

*Corresponding author.

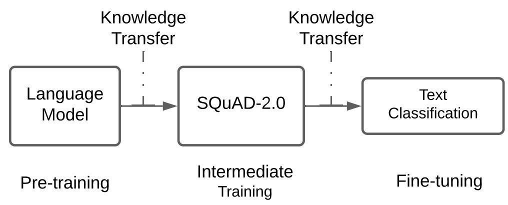

Figure 1: We experiment SQuAD-2.0 as the intermediate training task for text classification tasks.

improved performance on target tasks (Gururangan et al., 2020). Another effective transfer learning approach named intermediate training that chooses to train a LM model on an intermediate task via supervised manner and then fine-tune it on target tasks. This also leads to promising results across various NLP tasks including text classification, QA and sequence labeling (Phang et al., 2018; Vu et al., 2020; Pruksachatkun et al., 2020).

Text classification is the problem of classifying text into categories or classes which has been widely studied. In terms of input, there are mainly two forms of text classification problems: single-sequence classification tasks (e.g., sentiment classification and topic classification) and pairwise tasks (e.g., NLI and IR-related QA). In recent years, a common approach to tackle text classification problems is to fine-tune a pre-trained LM on target text classification tasks. Recently, advanced transfer learning-based approaches have been proposed to further improve the performance. For example, a recent work (Sun et al., 2019) has studied how to fine-tune BERT for text classification. They found that further pre-training LM using data within-task or in-domian can improve the performance of BERT for text classification tasks.

More recently, cross-task transfer learning technique for text classification has been investigated (Vu et al., 2020), and it is found that tasks that require high-level inference and reasoning abilities, such as natural language inference and question answering (QA) (Rajpurkar et al., 2018), are often the best intermediate tasks for text classification tasks. In a recent study (Pruksachatkun et al., 2020), it is found that natural language inference and QA tasks are generally helpful as intermediate tasks. Vu et al. (2020) showed that SQuAD-2.0 is the most favourable intermediate task for text classification. There are only a few text classifications tasks and only one language model (BERT) in their experiments, making it hard to conclude that SQuAD-2.0 as the intermediate task can generally improve the performance of all types of text classification tasks.

In this paper, we investigate the effectiveness of intermediate training for four different LMs – ELECTRA, RoBERTa, MobileBERT, and SqueezeBERT)– using the most popular QA resource SQuAD-2.0 as the intermediate task for eight target text classification tasks. We found that intermediate training shows varying transfer performance across different language models. Particularly contrary to previous thoughts, intermediate training with high-level inference QA tasks does not generally show positive transfer for low-level inference text classification tasks.

2 Related Work

As a large quantity of labeled data is not always available for training deep learning models, transfer learning becomes quite important for many of NLP problems. With transfer learning, widely available unlabeled text corpora containing rich semantic and syntactic information can be leveraged for learning language models, such as BERT (Devlin et al., 2019), GPT (Brown et al., 2020), and T5 (Raffel et al., 2020). Then, these language models are fine-tuned on downstream tasks, which is the dominant transfer learning method adopted in NLP at the moment. The second way of using transfer learning in NLP is to further pre-train pre-trained language models in domain data before fine-tuning on downstream tasks (Gururangan et al., 2020; Sun et al., 2019). The third approach, which is the method we investigate in our work, is to transfer models fine-tuned on an intermediate task for a target task (Pruksachatkun et al., 2020).

A recent work (Pruksachatkun et al., 2020) investigated when and why intermediate-task training is beneficial for a given target task. They experimented with 11 intermediate tasks and 10 target tasks, and find that intermediate tasks requiring high-level inference and reasoning abilities tend to work best, such as natural language inference tasks and QA tasks. Another recent work (Vu et al., 2020) has explored transferability across three types of tasks, namely text classification/regression, question answering and sequence labeling. They found that transfer learning is more beneficial for low-data source tasks and also found that data size, task and domain similarity, and task complexity all can affect transferability.

3 Methods

To find out whether using SQuAD-2.0 as the intermediate training task is generally helpful for text classification tasks for different language models, we experiment with 8 single-sequence text classification tasks and 8 sequence-pair text classification tasks, across four language models.

In SQuAD-2.0, each question is given a context from which to infer the answer. A QA system is expected to extract a span of text from that given context. More specifically, given a context C which consists of n tokens ($[t_1, t_2, ...t_n]$) and a question Q, a QA model is expected to predict the position of the start and end tokens of the answer in the context C. To correctly extract the answer span, on one hand an SQuAD-2.0 model needs to learn word-level dependencies between two sequences (semantic similarity); on the other hand it learns how to infer an answer from the context given a question. Training a transformer-based LM for SQuAD-2.0 intuitively enforces model's ability on inference and measuring semantic similarity, which is shown in previous studies (Pruksachatkun et al., 2020; Vu et al., 2020) to benefit text classification target tasks at the lower, sequence-level, either classification of single sequences or classification of the inference or similarity for sequence pairs.

When using transformer-based models for pairwise text classification, often a special token (e.g., [SEP]) is added between two sequences, similar to the QA input. We are interested in whether such a similarity between QA tasks and sequence-pair text classification tasks can make a difference. In terms of training procedure, we follow previous works (Phang et al., 2018; Vu et al., 2020). Specifically, we first fine-tune a pre-trained LM on SQuAD-2.0 (intermediate training stage) and then fine-tune it on each text classification tasks.

When adopting transformer-based language models (LM) for span extraction, we first load a

Table 1: Dataset Statistics

	Task	#DataSize (Training/Testing)	#Classes	Metric	Source
AGNEWS (Zhang et al., 2015)	Topic Classification	120000/7600	0: 31900, 1: 31900, 2: 31900, 3: 31900	Accuracy	News
SST2 (Wang et al., 2018)	Sentiment Classification	67349/872	0: 30208, 1: 38013	Accuracy	Movie Reviews
LIAR (Wang, 2017)	Fake News Detection	10269/1283	0: 2248, 1: 2390, 2: 2215, 3: 1894, 4: 1871, 5: 934	F1	POLITIFACT.COM
OFFENSIVE (Barbieri et al., 2020)	Offensive Speech Detection	11916/1324	0: 8595, 1: 4181	F1	Twitter
HATE (Barbieri et al., 2020)	Hate Speech Detection	9000/2970	0: 6935, 1: 5035	F1	Twitter
COLA (Wang et al., 2018)	Linguistic Acceptability	8551/1043	0: 2850, 1: 6744	Matthews Correlation	Books and Journal
EMOTION (Barbieri et al., 2020)	Emotion Detection	3257/1421	0: 1958, 1: 1066, 2: 417, 3: 1237	F1	Twitter
IRONY (Barbieri et al., 2020)	Irony Detection	2862/784	0: 1890, 1: 1756	F1	Twitter
MNLI (Wang et al., 2018)	Natural Language Inference	392702/9815	0: 134378, 1: 134023, 2: 134116	Accuracy	Multiple Text Corpus
QQP (Wang et al., 2018)	Quora Question Pairs	363846/40430	0: 255013, 1: 149263	F1	Quora
QNLI (Wang et al., 2018)	Question Answering	104743/5463	0: 55079, 1: 55127	Accuracy	Wikipedia
WIKIQA (Yang et al., 2015)	Question Answering	20360/2733	0: 25192, 1: 1333	F1	Wikipedia
BOOLQ (Wang et al., 2019)	Boolean Questions	9427/3270	0: 4790, 1: 7907	F1	Google search
MRPC (Wang et al., 2018)	Semantic Equivalence	3668/408	0: 1323, 1: 2753	F1	News
RTE (Wang et al., 2018)	Recognizing Textual Entailment	2490/277	0: 1395, 1: 1372	Accuracy	News and Wikipedia
WNLI (Wang et al., 2018)	Natural Language Inference	635/71	0: 363, 1: 343	Accuracy	Winograd Schema Challenge

pre-trained LM and then add a span classification head on top of it (a linear layer on top of the hidden-states output). A span classification head eventually generates two logits for each token, namely a logit for the start token and a logit for the end token. Learning a SQuAD-2.0 model performs classification at the token-level – classify a token either the start token or the end token. At inference stage, predictions are made based on logits (taking the token with the largest start logits as a start token and the token with largest end logits as an end token).

After we train a SQuAD-2.0 model, the next step is to transfer it for text classification tasks. When transferring a SQuAD-2.0 model, we only need to change a span classification head to a sequence classification head. The transferred transformer with a new sequence classification head will then be fine-tuned on text classification tasks. The weights of both the transferred SQuAD-2.0 model and the classification head will be updated during the fine-tuning stage. Therefore, the training process consists of three training stages, namely pre-training stage (pre-training a LM), intermediate training stage (fine-tuning on SQuAD-2.0), and fine-tuning stage (fine-tuning on each text classification tasks).

4 Experiments

4.1 Data and models

The dataset statistics and evaluation metrics for each task are shown in Table 1. We selected 8 single-sequence text classification tasks and 8 sequence-pair text classification tasks, covering binary and multi-class classification problems, balanced and imbalanced datasets, data-rich and data-scarce tasks, and different data sources. We select four pre-trained transformer-based LMs, namely ELECTRA (Clark et al., 2019), RoBERTa (Liu et al., 2019), MobileBERT (Sun et al., 2020), SqueezeBERT (Iandola et al., 2020).

4.2 Results

Experiment results (averaged over three runs) are reported in Table 2 and Table 3. Note that QQP, QNLI, MNLI, MRPC, WNLI, RTE, and COLA are sub-tasks of language understanding benchmark GLUE (Wang et al., 2018) widely used for LM evaluation. Our results are slightly different from (lower than) those reported in their paper, as we used the same setting of hyper-parameters (e.g., epoch, learning rate, input length, and batch size) for all LMs rather than tuning hyper-parameters, for fair comparison across all LMs.

According to Table 2, we can see that SQuAD2-tuned models for single-sequence text classification tasks have mixed results. On data-rich tasks, such as AGNEWS and SST2, the performance of SQuAD2-tuned models are slightly worse, except for RoBERTa(T) and MobileBERT(T) which have slightly better performance on SST2. On data-poor tasks, such as IRONY and EMOTION, transferred SQuAD2 models also tend to perform worse. In case of multi-class problems, such as AGNEWS and LIAR, the performance of models with SQuAD2 fine-tuning are not consistent. For example, ELECTRA(T), MobileBERT(T) and SqueezeBERT(T) improved the performance on LIAR, while RoBERTa(T) did not. Overall, we can see that SQuAD2-tuned models show varying transfer performance across four language models for single-sequence classification.

The results of sequence-pair text classification are reported in Table 3. Sequence-pair tasks can be roughly categorized into two groups, namely similarity tasks (e.g., QQP, MPRC) and inference tasks. Similarity tasks measure the semantic similarity between two sequences, while inference tasks measure the semantic relations between two sequences. Inference tasks have two sub-groups: natural language inference (e.g., WNLI, MNLI and RTE)

	AGNEWS	SST2	LIAR	OFFENSIVE	HATE	COLA	EMOTION	IRONY
ELECTRA	94.46	94.61	26.63	83.48	48.01	67.65	82.59	71.96
ELECTRA(T)	94.59^{+}	94.26^{-}	27.76^{+}	82.91^{-}	44.90^{-}	67.01^{-}	81.86^{-}	70.96^{-}
RoBERTa	94.84	93.00	27.65	83.18	44.19	58.84	82.75	71.41
RoBERTa(T)	$94.82^{=}$	94.15^{+}	27.35^{-}	83.45^{+}	46.62^{+}	57.17^{-}	81.79^{-}	69.35^{-}
MobileBERT	94.57	90.13	26.07	84.71	43.66	49.99	78.23	63.08
MobileBERT(T)	94.32^{-}	91.05^{+}	26.27^{+}	85.01^{+}	45.57^{+}	50.25^{+}	79.72^{+}	62.36^{-}
SqueezeBERT	94.68	89.90	27.26	84.09	41.97	44.50	78.72	66.07
SqueezeBERT(T)	94.09^{-}	89.10^{-}	27.72^{+}	83.61^{-}	40.54^{-}	35.37^{-}	77.73^{-}	66.44^{+}

Table 2: Performance(%) for single-sequence text classification tasks. Models with SQuAD2.0 intermediate tuning are denoted with T, $+$, $=$ and $-$ denote increase, equal and decrease in performance for SQuAD-tuned models.

	QQP	**QNLI**	WNLI	MNLI	**WIKIQA**	**BOOLQ**	MRPC	RTE
ELECTRA	91.69	92.09	47.88	88.52	46.04	84.16	88.60	77.61
ELECTRA(T)	91.45^{-}	$\mathbf{92.44}^{+}$	52.58^{+}	88.77^{+}	$\mathbf{50.43}^{+}$	$\mathbf{86.34}^{+}$	87.78^{-}	78.34^{+}
RoBERTa	91.24	92.04	56.34	87.69	43.41	84.22	89.56	75.33
RoBERTa(T)	91.14^{-}	$\mathbf{92.42}^{+}$	$56.34^{=}$	$87.65^{=}$	$\mathbf{52.45}^{+}$	$\mathbf{84.54}^{+}$	88.31^{-}	79.18^{+}
MobileBERT	89.09	89.18	46.48	82.63	40.18	77.65	83.69	56.68
MobileBERT(T)	88.94^{-}	$\mathbf{90.88}^{+}$	35.21^{-}	82.45^{-}	$\mathbf{52.60}^{+}$	$\mathbf{81.63}^{+}$	86.87^{+}	67.75^{+}
SqueezeBERT	89.32	89.16	52.11	80.49	41.70	79.45	83.62	68.11
SqueezeBERT(T)	89.07^{-}	$\mathbf{90.13}^{+}$	39.90^{-}	80.05^{-}	$\mathbf{50.89}^{+}$	$\mathbf{79.98}^{+}$	85.31^{+}	66.79^{-}

Table 3: Performance(%) for pairwise classification tasks. Models with SQuAD2.0 intermediate tuning are denoted with T, where $+$, $=$ and $-$ denote increase, equal and decrease in performance for SQuAD-tuned models. Note the positive transfer results on QA tasks QNLI, WIKIQA and BOOLQ.

and QA-related tasks (e.g., QNLI, WIKIQA and BOOLQ). We can see that SQuAD2-tuned models have consistently better performance for QA tasks QNLI, WIKIQA and BOOLQ. A possible explanation is when trained on SQuAD-2.0, if a question is unanswerable, the index of [CLS] token is usually set as the answer, which means that the representation of [CLS] token contains information about whether a question has the answer in the given context. On similarity tasks, SQuAD2-tuned models have worse performance on QQP (data-rich), but on MRPC (data-poor) SQuAD2-tuned models tend to have mixed performance. On natural language inference tasks, MNLI (data-rich) seems not benefit from SQuAD2 fine-tuning, but the performance on WNLI (data-poor) has shown some improvements. Our experiments show that SQuAD2-tuned models have seen consistent success on QA tasks, but generally sequence-pair tasks do not always benefit from this intermediate training, whether data rich or data-poor. Consequently, it is still hard to conclude that using SQuAD-2.0 as the intermediate training task is generally helpful for text classification.

5 Conclusion

We studied using the SQuAD-2.0 QA intermediate task for target text classification across different language models. Our experiments on eight classification target tasks and four language models show that SQuAD2-tuned models do not generally have better performance, whether single-sequence or sequence-pair, or data-rich or data-poor settings. This result highlights that high-level inference intermediate tasks may not generally produce positive transfer as previously thought. On the other hand, SQuAD-tuned models always have positive transfer results for QA tasks, which suggests further research is needed to investigate if task similarity rather than task complexity plays a significant role for intermediate training.

Acknowledgements

This initiative was funded by the Australian government Department of Defence and the Office of National Intelligence under the AI for Decision Making Program, delivered in partnership with the Defence Science Institute in Victoria.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1644–1650.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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*, pages 4171–4186.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Forrest Iandola, Albert Shaw, Ravi Krishna, and Kurt Keutzer. 2020. Squeezebert: What can computer vision teach nlp about efficient neural networks? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 124–135.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Xiang Zhang, Junbo Zhao, and Yann Lecun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 2015:649–657.