Multi-Stage Pre-training for Low-Resource Domain Adaptation

Rong Zhang[*], **Revanth Gangi Reddy**[*†], **Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avirup Sil**[‡], **Todd Ward**
IBM Research AI
{zhangr,avi,raduf,roukos,toddward,vittorio}@us.ibm.com,
g.revanthreddy111@gmail.com, {arafat.sultan,aferritto}@ibm.com,
efsun@gwu.edu

Abstract

Transfer learning techniques are particularly useful in NLP tasks where a sizable amount of high-quality annotated data is difficult to obtain. Current approaches directly adapt a pre-trained language model (LM) on in-domain text before fine-tuning to downstream tasks. We show that extending the vocabulary of the LM with domain-specific terms leads to further gains. To a bigger effect, we utilize structure in the unlabeled data to create auxiliary synthetic tasks, which helps the LM transfer to downstream tasks. We apply these approaches incrementally on a pre-trained Roberta-large LM and show considerable performance gain on three tasks in the IT domain: Extractive Reading Comprehension, Document Ranking and Duplicate Question Detection.

1 Introduction

Pre-trained language models (Radford et al., 2019; Devlin et al., 2019; Liu et al., 2019) have pushed performance in many natural language processing tasks to new heights. The process of model construction has effectively been reduced to extending the pre-trained LM architecture with simpler task-specific layers, while fine-tuning on labeled target data. In cases where the target task has limited labeled data, prior work has also employed transfer learning by pre-training on a source dataset with abundant labeled data before fine-tuning on the target task dataset (Min et al., 2017; Chung et al., 2018; Wiese et al., 2017). However, directly fine-tuning to a task in a new domain may not be optimal when the domain is distant in content and terminology from the pre-training corpora.

To address this language mismatch problem, recent work (Alsentzer et al., 2019; Lee et al., 2019; Beltagy et al., 2019; Gururangan et al., 2020) has adapted pre-trained LMs to specific domains by continuing to train the same LM on target domain text. Similar approaches are also used in multilingual adaptation, where the representations learned from multilingual pre-training are further optimized for a particular target language (Liu et al., 2020; Bapna and Firat, 2019). However, many specialized domains contain their own specific terms that are not part of the pre-trained LM vocabulary. Furthermore, in many such domains, large enough corpora may not be available to support LM training from scratch. To resolve this out-of-vocabulary issue, in this work, we *extend* the open-domain vocabulary with in-domain terms while adapting the LM, and show that it helps improve performance on downstream tasks.

While language modeling can help the model better encode the domain language, it might not be sufficient to gain the domain knowledge necessary for the downstream task. We remark, however, that such unlabeled data in many domains can have implicit structure which can be taken advantage of. For example, in the IT domain, technical documents are often created using predefined templates, and support forums have data in the form of questions and accepted answers. In this work, we propose to make use of the structure in such unlabeled domain data to create synthetic data that can provide additional domain knowledge to the model. Augmenting training data with generated synthetic examples has been found to be effective in improving performance on low-resource tasks. Golub et al. (2017), Yang et al. (2017), Lewis et al. (2019) and Dhingra et al. (2018) develop approaches to generate natural questions that can aid downstream question answering tasks. However, when it is not possible to obtain synthetic data that exactly fits the target task description, we show that creating auxiliary tasks from such unlabeled data can be

[*] Both authors contributed equally.
[†] Work done during AI Residency at IBM Research.
[‡] Corresponding author.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5461–5468,
November 16–20, 2020.

useful to the downstream task in a transfer learning setting.

For preliminary experiments in this short paper, we select the IT domain, partly because of the impact such domain adaptation approaches can have in the technical support industry. The main contributions of this paper are as follows: **(1)** We show that it is beneficial to extend the vocabulary of a pre-trained language model while adapting it to the target domain. **(2)** We propose to use the inherent structure in unlabeled data to formulate synthetic tasks that can transfer to downstream tasks in a low-resource setting. **(3)** In our experiments, we show considerable improvements in performance over directly fine-tuning an underlying RoBERTa-large LM (Liu et al., 2019) on multiple tasks in the IT domain: extractive reading comprehension (RC), document ranking (DR) and duplicate question detection (DQD).[1]

2 Datasets

We use two publicly available IT domain datasets. Table 1 shows their size statistics.

TechQA (Castelli et al., 2019) is an extractive reading comprehension (Rajpurkar et al., 2016) dataset developed from real user questions in the customer support domain. Each question is accompanied by 50 documents, at most one of which has the answer. A companion collection of 801K unlabeled Technotes is provided to support LM training. In addition to the primary reading comprehension task (TechQA-RC), we also evaluate on a new document ranking task (TechQA-DR). Given the question, the task is to find the document that contains the answer.

AskUbuntu[2] (Lei et al., 2016) is a dataset containing user-marked pairs of similar questions from Stack Exchange[3], which was developed for a duplicate question detection task (AskUbuntu-DQD). A static offline dump of AskUbuntu, which is organized as a set of forum posts[4], is also available and can be used for LM training.

[1] Scripts are available here.
[2] *askubuntu.com*
[3] *stackexchange.com*
[4] *archive.org/download/stackexchange/askubuntu.com.7z*

Dataset	Train	Dev	Test	Unlabeled
TechQA	600	310	490	306M
AskUbuntu	12,724	200	200	126M

Table 1: Size statistics for two IT domain datasets. Train/Dev/Test: # examples, Unlabeled: # tokens.

3 Vocabulary Extension for LM Adaptation

Texts in specialized fields including technical support in the IT domain may contain numerous technical terms which are not found in open domain corpora and are therefore not well captured by the vocabulary of out-of-the-box LMs. These terms are often over-segmented into small pieces (sub-word tokens) by the segmenter rules, which are learned from the statistics of open domain language.

As an example, the token out-of-vocabulary (OOV) rate of the standard RoBERTa vocabulary in the TechQA Technotes data is 19.8% and the BPE/TOK ratio is 1.32. Contrast this with the analogous figures for 1M randomly selected Wikipedia sentences, where the OOV rate is only 8.1% and the BPE/TOK ratio is 1.12. While transformer-based pre-trained language models (Devlin et al., 2019; Liu et al., 2019) yield better representations of previously unseen tokens than traditional n-gram models, over-segmentation can still cause degradation in downstream task performance.

We address this challenge by augmenting the vocabulary of the pre-trained LM with frequent in-domain words. Specifically, the most frequent OOV tokens after tokenization are recorded and used to bypass the BPE segmentation stage. This prevents the segmenter from splitting these terms into smaller pieces. New entries in the LM vocabulary and corresponding word embeddings are created for these tokens. In our experiments, the number of such protected tokens is decided using an empirical criterion: we require that 95% of the in-domain data be covered by the extended vocabulary. We add 10k new items to the vocabulary for the Technotes corpus and 5k for the AskUbuntu corpus. The variation in coverage due to different numbers of new vocabulary entries is shown in the appendix. The pre-trained LM is then adapted to the domain-specific corpus via masked LM (MLM) training. The embeddings of the new vocabulary are randomly initialized and then learned during the MLM training. The embeddings of existing vocabulary are also fine-tuned in this phase.

4 Task-Specific Synthetic Pre-training

While in-domain LM pre-training reveals novel linguistic patterns in target domain text, in many domains including technical documents, structure present in unlabeled text can contain useful information closer to actual end tasks. In this section, we propose to utilize such structure in unlabeled data to create auxiliary pre-training tasks and associated synthetic training data, which in turn can help target tasks via transfer learning.

TechQA. The TechQA dataset release contains a companion Technotes collection with 801K human written documents with titled sections. We observe that certain sections in these documents (e.g., *Abstract*, *Error Description* and *Question*) correspond to a problem description, while others (e.g., *Cause* and *Resolving the Problem*) describe the solution[5]. We create an auxiliary reading comprehension (RC) task from these documents. Specifically, if a document contains both problem and solution sections, a synthetic example is created where the problem description section is the query, the solution section is the target answer, and the entire document excluding the query section is the context. Additionally, ten other documents are sampled from the Technotes corpus as negatives to simulate *unanswerable* examples. This auxiliary task trains an intermediate RC model which predicts the start and end positions of the solution section as the answer given the document and the problem description. While our main goal here is to generate long-answer examples common in TechQA, the general idea of utilizing the document structure can be applicable in other scenarios including in scientific domains like Bio/Medical (G. Tsatsaronis, G. Balikas, P. Malakasiotis, et al., 2015; Lee et al., 2019) where structured text is relatively common.

AskUbuntu. The AskUbuntu dataset contains a web dump of forum posts, each containing a question and multiple answers, with one answer possibly labeled by users as "Accepted". Motivated by (Qiu and Huang, 2015; Lei et al., 2016; Rücklé et al., 2019), we create an auxiliary answer selection task from this structure. Each instance in the synthetic data for this task contains a question, its accepted answer as the positive class, and an answer randomly sampled from other question posts as the negative class. An intermediate classification model is learned from these annotations, whose weights are used to initialize the target duplicate question detection (DQD) model. Even though this auxiliary task adopts a different question-answer classification objective than the DQD task's objective of question-question classification, our experimental results show that the former still serves a good initialization for the latter.

[5]Here is a sample technote: Link

5 Experiments

5.1 Setup

Our experiments build on top of the RoBERTa-large LM. We adopt the standard methodology of using the pre-trained LM as the encoder and processing the contextualized representations it produces using task-specific layers. For the TechQA-RC task, we follow (Devlin et al., 2019) and predict the start and end position of the answer span with two separate classifiers, trained using cross entropy loss. For the TechQA-DR and AskUbuntu-DQD tasks, we follow (Adhikari et al., 2019) and classify the [CLS] token representation at the final layer with a binary classifier trained using the binary cross entropy loss; during inference, we rank the documents or questions according to their classification score. For all the tasks, during finetuning, we train the entire model end-to-end. We refer the reader to the appendix for details on hyperparameter values for all the experiments.

For the TechQA-RC task, we report both the main metric, F1, and the ancillary F1 for answerable questions, HA_F1, to capture the effects of our approach both on the end-to-end pipeline (F1) and on the answer extraction component (HA_F1). For TechQA-DR, models are evaluated by Match@1 and Match@5. For AskUbuntu-DQD, we report MAP, MRR, Precision@1 and Precision@5 following (Lei et al., 2016).

5.2 Synthetic Pre-training Corpus and Labeled Data Augmentation

Using the method described in section 4, we use the 801K Technotes to construct a synthetic corpus for the TechQA tasks. The synthetic data contains 115K positive examples, each of which has 10 randomly selected documents as negatives. For the AskUbuntu-DQD, a 210K-example synthetic corpus is constructed from the web dump data, with a positive:negative example ratio of 1:1.

Since TechQA is a very-low resource dataset

Model	Dev		Test	
	HA_F1	F1	HA_F1	F1
BERT	34.7	55.3	25.4	53.0
RoBERTa	35.0 (1.6)	58.2 (1.1)	29.3	54.4
+ Domain LM	35.2 (1.5)	58.3 (1.7)	-	-
+ 10k Vocab Ext.	36.9 (1.7)	58.5 (0.7)	-	-
+ RC Pre-training	39.7 (1.2)	59.0 (0.7)	-	-
+ Data Augmentation	**40.6** (1.4)	**59.9** (1.0)	**32.1**	**56.7**

Table 2: Results on TechQA-RC task. Each row with a + adds a step to the previous row. HA_F1 refers to F1 for answerable questions. Numbers in parentheses show standard deviation.

Model	Dev		Test	
	M@1	M@5	M@1	M@5
IR	0.437	0.637	-	-
RoBERTa	0.576 (0.017)	0.770 (0.027)	0.512	0.748
+ Domain LM	0.593 (0.020)	0.808 (0.021)	-	-
+ 10k Vocab Ext.	0.596 (0.013)	0.790 (0.024)	-	-
+ RC Pre-training	0.625 (0.014)	0.826 (0.023)	-	-
+ Data Augmentation	**0.638** (0.029)	**0.850** (0.012)	**0.536**	**0.808**

Table 3: Experimental results on TechQA-DR task. Each row with a + adds a step to the previous row. M@1 is short for Match@1 and M@5 for Match@5. Numbers in parentheses show standard deviation.

with only 600 training examples, we additionally apply data augmentation techniques to increase the size of the training set. We use simple data perturbation strategies, such as adding examples with only parts of the original query, randomly dropping words in query and passage, duplicating positive examples, removing stop words, dropping document title in the input sequence etc., to increase the size of the training set by 10 times. This augmented training set is only used under the data augmentation setting while fine-tuning on the TechQA tasks.

5.3 Results and Analysis

For each of our approaches, we show performance of the model when fine-tuned on the downstream tasks in TechQA and AskUbuntu datasets. All the numbers reported are averages over 5 seeds, unless otherwise stated. Standard deviation numbers are shown in parentheses.

TechQA-RC Table 2 describes the performance on the RC task in the TechQA dataset. The BERT baseline numbers are from (Castelli et al., 2019). Here, model performance is compared on the dev set and we report the blind test set numbers[6] for our single-best baseline and final models.

[6]Obtained by submitting to the *TechQA leaderboard*.

Adapting the LM without extending the vocabulary yields just 0.2 points over the RoBERTa-large baseline. Augmenting the vocabulary by 10k word pieces improves the HA_F1 score by 1.7 points. Furthermore, our RC-style synthetic pre-training yields a considerable improvement of 2.8 points on HA_F1 and 0.5 points on F1. Finally, data augmentation further boosts performance by about a point on both HA_F1 and F1, suggesting that data augmentation via simple perturbations can be effective in a very-low resource setting.

TechQA-DR Table 3 shows results from our experiments on the auxillary document ranking task over the TechQA dataset [7]. We use BM25 (Robertson and Zaragoza, 2009) as our IR baseline. We see that the RoBERTa models substantially outperform the IR system. Although vocabulary expansion only helps by 0.3 points in Match@1, we see considerable improvements in performance from our other approaches. The "RC Pre-training" entry shows a Match@1 improvement of 2.9 points over the language modelling. This demonstrates the effectiveness of pre-training on an ancillary task in a

[7]Since this is not the official task in the TechQA dataset, numbers on the test set were obtained by the TechQA leaderboard manager who agreed to run our scoring script on an output file produced by our submission

Model	Dev				Test			
	MAP	**MRR**	**P@1**	**P@5**	**MAP**	**MRR**	**P@1**	**P@5**
RoBERTa	0.634	0.733	0.588	0.514	0.663	0.778	0.654	0.510
	(0.009)	(0.014)	(0.022)	(0.010)	(0.014)	(0.024)	(0.038)	(0.009)
+ Domain LM	0.647	0.753	0.622	0.523	0.677	0.799	0.676	0.515
	(0.007)	(0.021)	(0.029)	(0.009)	(0.012)	(0.019)	(0.028)	(0.008)
+ 5k Vocab Ext.	0.653	0.750	0.608	0.532	0.686	0.817	0.704	0.517
	(0.016)	(0.024)	(0.038)	(0.012)	(0.017)	(0.020)	(0.033)	(0.004)
+ DQD Pre-training	**0.672**	**0.775**	**0.647**	**0.548**	**0.704**	**0.825**	**0.714**	**0.532**
	(0.008)	(0.012)	(0.023)	(0.007)	(0.012)	(0.015)	(0.028)	(0.008)

Table 4: Experimental results on AskUbuntu-DQD task. Each row with a + adds a step to the previous row. P@1 and P@5 refer to Precision@1 and Precision@5, respectively. Numbers in parentheses show standard deviation.

transfer-learning setting for the document ranking task. We further see an improvement of 1.3 points from data augmentation.

AskUbuntu-DQD Table 4 shows results for the DQD task on the AskUbuntu dataset. We see that our methods give incremental improvements in performance. Our final model is considerably better than the RoBERTa-large baseline on all four metrics. We see the biggest gain in performance from the synthetic pre-training task demonstrating its relevance to the DQD task. For this dataset, we didn't explore data augmentation strategies because it had a considerable number of training instances (see Table 1) compared to the TechQA dataset.

6 Conclusion

In this work, we show that it is beneficial to extend the vocabulary of the LM while fine-tuning it on the target domain language. We show that extending the pre-training with task-specific synthetic data is an effective domain adaptation strategy. We empirically demonstrate that structure in the unsupervised domain data can be used to formulate auxillary pre-training tasks that can help downstream low-resource tasks like question answering and document ranking. In our preliminary experiments, we empirically show considerable improvements in performance over a standard RoBERTa-large LM on multiple tasks. In future work, we aim to extend our approach to more domains and explore more generalizable approaches for unsupervised domain adaptation.

Acknowledgments

We thank Cezar Pendus for his help with submitting to the TechQA leaderbaord. We would also like to thank the multilingual NLP team at IBM Research AI and the anonymous reviewers for their helpful suggestions and feedback.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. DocBERT: BERT for document classification. arXiv 1904.08398.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. 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 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Mike McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avirup Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2019. The TechQA dataset. arXiv 1911.02984; To appear in Proc. ACL 2020.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. 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.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

G. Tsatsaronis, G. Balikas, P. Malakasiotis, et al. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(138).

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 835–844, Copenhagen, Denmark. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovi, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. arXiv 2004.10964.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289, San Diego, California. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

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 1907.11692.

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, pages 1305–1311. AAAI Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019. Neural duplicate question detection without labeled training data. 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 1607–1617, Hong Kong, China. Association for Computational Linguistics.

Georg Wiese, Dirk Weissenborn, and Mariana Neves. 2017. Neural domain adaptation for biomedical question answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 281–289, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised QA with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

A.1 Implementation Details

In our experiments, we used the Fairseq toolkit (Ott et al., 2019) for language modelling and the Transformers library (Wolf et al., 2019) for downstream tasks. For all of our target models, when fine-tuning on the downstream task, we choose the hyperparameters by grid search and pick the best models on the dev set according to the evaluation metrics for the corresponding task. For TechQA-RC task, we pick the best model according to (HA_F1 + F1) and for TechQA-DR, we choose based on Match@1. For the AskUbuntu-DQD, we pick the best model based on MAP. The best hyperparamters for each of the tasks are shown in the Tables 5 to 8 below:

Hyperparameter	Setting
WARMUP UPDATES	10000
PEAK LR	0.00015
TOKENS PER SAMPLE	512
MAX POSITIONS	512
MAX SENTENCES	8
UPDATE FREQ	64
OPTIMIZER	adam
DROPOUT	0.1
ATTENTION DROPOUT	0.1
WEIGHT DECAY	0.01
MAX Epochs	5
CRITERION	mask-whole-words

Table 5: Hyperparameters for the LM training.

Hyperparameter	Setting
Learning Rate	5.5e-6
Max Epochs	15
Batch Size	32
Max Sequence Length	512
Document Stride	192
Sampling Rate for Unanswerable Spans	0.15
Maximum Query Length	110
Maximun Answer Length	200

Table 6: Hyperparameters for the TechQA-RC task.

Hyperparameter	Setting
Learning Rate	2.5e-6
Max Epochs	20
Batch Size	32
Max Sequence Length	512
Document Stride	192
Sampling Rate for Negative Documents	0.1
Maximum Query Length	110

Table 7: Hyperparameters for the TechQA-DR task.

Hyperparameter	Setting
Learning Rate	5.5e-6
Max Epochs	5
Batch Size	32
Max Sequence Length	512
Maximum Question Length	256
Maximun Answer Length	256

Table 8: Hyperparameters for the AskUbuntu-DQD task.

A.2 Extension of Vocabulary

The Table 9 below shows the variation of coverage and BPE/TOK ratio with the number of word pieces added to the vocabulary for the Technotes Collection.

# of Added Word Pieces	Coverage	BPE/TOK
+0k	80.2%	1.32
+5k	94.4%	1.13
+10k	95.4%	1.11
+15k	95.8%	1.10

Table 9: Coverage and BPE/TOK ratio vs the number of word pieces added to the vocabulary for the Technotes collection.