Pre-Training Methods for Question Reranking

Anonymous ACL submission

Abstract

One interesting approach to Question Answering (QA) is to search for semantically similar questions, which have been answered before. This task is different from answer retrieval as it focuses on questions rather than only on the answers, therefore it requires different model training on different data. In this work, we introduce a novel unsupervised pre-training method specialized for retrieving and ranking questions. This leverages (i) knowledge distillation from a basic question retrieval model, and (ii) new pre-training task and objective for learning to rank questions in terms of their relevance with the query. Our experiments show that (i) the proposed technique achieves state-of-the-art performance on QRC and Quora-match datasets, and (ii) the benefit of combining re-ranking and retrieval models.

1 Introduction

An effective approach for answering user questions is to find semantically identical questions, which have been previously answered. Although this method cannot be applied to completely new questions, it provides optimal solutions for applications such as Frequently Asked Questions (FAQs) (Sakata et al., 2019), Forum services (Hoogeveen et al., 2015; Lei et al., 2016), and QA caching systems (Campese et al., 2023; Lewis et al., 2021), as it provides cheaper and more efficient access to answers than the system generated them.

These Data Base-based QA systems (DBQAS) typically consist of three components: (i) a DB of questions with their answers, (ii) a retrieval model, which given a question, Q, retrieves its most similar questions, and (iii) a selection model, which can more accurately rerank the questions in terms of semantically equivalence. The answer associated with the top-ranked question is typically used as the system output. The fine-tuning of the retrieval and ranking models requires training data, labeled in a ranking fashion, i.e., given the query (target question), its top similar k questions needs to be labelled as semantically equivalent or not. While datasets, e.g., QUORA, constituted by annotated samples of question-question pairs can be used for an initial training, ranking data is essential to obtain optimal accuracy. Unfortunately, these datasets require intensive and costly annotation processes and resources to be built. For example, even an annotation workflow built using Amazon Mechanical Turk, is costly[1].

Alternative approaches to reducing the amount of data have been proposed, ranging from data augmentation (Wang and Li, 2023; Yang et al., 2019a) to specialized pre-training (PT) techniques that are aligned with the downstream task. For example, Lee et al. (2019) proposed the The Inverse Cloze Task, an unsupervised PT technique based on a discriminative objective that captures some features of answer retrieval. Di Liello et al. (2022a,b) pre-trained on Wikipedia, simulating the task of Answer Sentence Selection (AS2), by selecting sentences that belong or not to the same document or paragraphs.

These methods focus on the relation between question and answer pairs, rather than between two questions, and, most importantly, they do not model the ranking task. In this work, we propose a novel PT technique using a loss function and a data, which surrogate a question re-ranking task. We generate an unsupervised dataset consisting of 18M examples using a re-implementation of the QADBS proposed by Campese et al. (2023), where each example comprises a question and a rank of five question-answer pairs. To generate PT data, we then swap the first QA pair with another one. The PT task consists in detecting whether the order of QA pairs in the rank is correct or it has been modified. This innovative approach both exploits

[1]We estimated the cost per question with 15 ranked items to be 2-3$ with labels from expert annotators.

(i) a new loss function and (ii) knowledge distilled from the retrieval model, i.e., the initial rank.

We tested our PT techniques for question re-ranking on two different datasets: (i) QRC (Campese et al., 2023), a question ranking resource designed for DBQAS training and testing, and (ii) Quora-match (Wang et al., 2020b), a binary-classification over question pairs. The results show that our approach achieves state-of-the-art performance on these benchmarks, e.g., +2% in question selection Accuracy on QRC. Moreover, we show interesting synergies between re-ranking PT and existing retrieval models, which can be further explored.

2 Related work

Various PT techniques have been developed for Transformer-based architectures. Most of them are based on general and intuitive tasks that can be applied over plain texts. These tasks are designed to teach the model to extract actionable information from text and to learn semantic patterns. First and foremost, Masked Language Model (MLM) PT task was introduced in BERT (Devlin et al., 2019), where the objective consists of predicting a small fraction of masked tokens, The same PT was applied to various other models, including RoBERTa (Liu et al., 2019) and MiniLM (Wang et al., 2020a), showing remarkable results in various downstream applications, including QA and Semantic text similarity. Alternative PT techniques were proposed by changing the MLM objective: (i) Permutation Language Model (PLM) (Yang et al., 2019b), where the model tries to predict the next token (left-to-right) of a sentence, whose tokens were permuted; (ii) Random Token Detection (RTD), where the model is trained to find a small amount of tokens replaced with plausible alternatives, generated by a separate model (ELECTRA by (Clark et al., 2020a)); (iii) Random Token Swap (RTS) (Di Liello et al., 2021), similarly to RTD, the model discriminates the original tokens from those swapped with tokens from the vocabulary; and (iv) Text-to-text objective Kale and Rastogi (2020), where spans of texts are masked to train the model generating coherent sequences. (v) Tan et al. (2020) replace tokens according to Text Normalization substitutions. Finally, (vi) Clark et al. (2020b) improves the way ELCTRA select complex tokens in RTD.

All the above techniques target individual tokens with operations, masking, swapping, replacing them. In contrast, our approach model the entire questions, requiring their classification in the objective function. A closer work to ours are sentence-based techniques, which take multiple sentences as input and try to categorize them: (i) Next Sentence Prediction (NSP) (Devlin et al., 2019) tries to predict if two input sentences appear side by side in a text or not. (ii) DeCLUTR (Giorgi et al., 2021) uses a contrastive learning objective to predict if two sentences come from the same document. (iii) Di Liello et al. (2022a,b) define objectives aiming at replicating the AS2 downstream task. They used continuous pre-training techniques on unlabeled data, where the objective is to predict when two sentences are part of the same paragraph. We propose an objective with the same aim of Di Liello et al., i.e., learning the downstream task, but it targets learning of ranking function of a new task, question rather than answer selection.

3 Question Ranking pre-training

We create pre-training data using (i) a basic QADBs to generate query/question rank data, and (ii) modifying the rank to simulate the ranking objective.

QADBS: this consists of (i) a DB of 38M q/a pairs, including 6M q/a pairs from Campese et al. (2023) and 32M additional pairs from PAQ (Lewis et al., 2021); (ii) a dense retrieval architecture of 33M parameters we built on top of MiniLM-12L-v2 (pre-trained on a corpus of 900 million sentence pairs for semantic text similarity (Reimers and Gurevych, 2019)). We fine-tuned it using QRC (see details on Appendix A). The retrieval model is a sentence-encoder, which generates the query embedding and, then, computes the cosine-similarity with the pre-computed embeddings associated with each q/a pair stored in the DB. This means that it can efficiently sort the entire DB, and returns the top k q/a pairs.

QRP Data: We collected 18M questions from WQA (Zhang et al., 2022), GooAQ (Khashabi et al., 2021), and PAQ dataset, and used as queries for QADBS, using the top $k = 5$ question/answer pairs ranked according to their similarity with the query. Then, we randomly selected 50% of the retrieved ranks. For each of them, we swap the top ranked q/a pair with one of the remaining pairs randomly selected. Specifically, we encoded each pre-training example as concatenation of its q/a

pairs, i.e., *[CLS]* q_1/a_1 *[SEP]* q_2/a_2 *...[SEP]* q_5/a_5 *[EOS]*. In the next sections, we refer to this resource as Question Ranking Pre-training data (QRP). We show some examples of QRP data in Appendix B.

Task and rationale: Our PT task consists of determining if a given rank was modified or not. The data does not include the input query. Therefore, to derive if the rank was modified or not the model must learn to internally reconstruct the original query that generated the rank. In this reconstruction step the model learns from the relations between the different candidates, which semantic property best represent the unknown query. Recognizing this property is very important for solving the downstream, which indeed requires them to select the most semantic similar question. Interestingly, as a proof of concept, we included the query in the PT data, our development loss showed that the objective could be learned easily and did not produce any improvement in our DBQAS.

4 Experiments

We compared our PT approach with several baselines on QRC and Quora datasets.

4.1 Datasets and metrics

QRC is a question ranking dataset of 15K queries, divided in training (11.5K), development (1.5K), and test(2K). Each query is associated with 30 q/a pairs, and each resulting triplet (q/q/a) receives a 0/1 label of the query/question equivalence. The model performance is computed on the rank using standard metrics, such as Precision@1 (P@1), MAP, and MRR.

Quora-match is a large dataset of 200K q/q/a triplets, but they are not organized in rankings. The task consists of identifying whether two questions are equivalent or not (binary classification). Therefore, this task is measured with classification metrics, such as ROC-AUC, Accuracy, and F1 score. Given that the dataset is unbalanced (35% positive, 65% negative), we mostly focus on ROC-AUC optimization.

4.2 Pre-Training (PT)

We consider multiple PT baselines: (i) public checkpoint without additional training; (ii) our Question Ranking PT (QR) defined in Section 3; (iii) models pre-trained on multiple existing and general objectives, including MLM, RTS, STS, and ALL (Di Liello, 2023). These models were all pre-trained on the same QRP data, thus we can directly measure the impact of pre-training objective.

Distillation: Our PT objective is conceptual similar to knowledge distillation, where the pre-trained model learns the output of the dense retrieval used to generate ranking data. We investigated two distinct approaches: First, the standard distillation method described by Hinton et al. (2015), where the loss is defined as linear combination of (i) the CrossEntropy loss between model prediction (s_s) and label (y), and (ii) MSE between the teacher (s_t) and the student (s_s) probability scores [0,1].

$$\mathcal{L}(y, s_s, s_t) = (1-\lambda)\mathcal{L}_{CE}(y, s_s)+\lambda\mathcal{L}_{MSE}(s_s, s_t)$$

λ is a regularization hyper-parameter selected through classical model selection, with values in $\lambda \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$.

Second, we considered an alternative distillation approach from Gabburo et al. (2023), defined as

$$\mathcal{L}(y, s_s, s_t) = \mathcal{L}_{CE}(y, s_s) \times (1 - s_t)$$

Intuitively, this loss increases the weight of examples, where the teacher score is low, helping the model fixing teacher's uncertainty. Finally, we combine distillation and pre-training approaches to highlight that our pre-training task can't be substituted by distillation approach.

4.3 Training

We use two steps: First, we trained a Transformer model on our generated QRP. Second, we fine-tune the model on QRC or Quora-match and measure performance. All of the models used in our experiments start from a Deberta-v3-base (He et al., 2021) public checkpoint[2]. To pre-train our baselines, we adopted a learning rate of $5e^{-6}$, a batch size of 1024, cross-entropy loss, while we fine-tune the models for 2 epochs. In the case of distillation approaches, we skip the first step (pre-training) and we distill the model on the target task directly. The teacher model is the same we used to generate QRP data, which is MiniLM-v2-12L. The teacher model was pre-trained on 900M sentence pairs and fine-tuned on QRC. Thus, in both cases, PT and distillation, we ingest question ranking knowledge into our models.

We fine-tuned the trained model on the two target datasets separately. In this step, we encoded

[2]Available at https://huggingface.co/microsoft/deberta-v3-base

Setting	P@1	MAP	MRR
Public ckp	$50.82_{\pm 0.38}$	$48.44_{\pm 0.07}$	$60.23_{\pm 0.23}$
PRE-TRAINING TECHNIQUES			
QR (our)	$\mathbf{51.87_{\pm 0.17}}$	$\mathbf{48.87_{\pm 0.06}}$	$\mathbf{60.98_{\pm 0.10}}$
QQR	$51.04_{\pm 0.44}$	$48.87_{\pm 0.18}$	$60.63_{\pm 0.20}$
MLM	$50.23_{\pm 0.42}$	$48.25_{\pm 0.18}$	$59.90_{\pm 0.23}$
RTS	$50.95_{\pm 0.42}$	$48.63_{\pm 0.08}$	$60.38_{\pm 0.24}$
STS	$50.97_{\pm 0.49}$	$48.60_{\pm 0.25}$	$60.36_{\pm 0.41}$
ALL	$50.85_{\pm 0.45}$	$48.68_{\pm 0.23}$	$60.23_{\pm 0.33}$
DISTILLATION APPROACHES			
Hinton et al. (2015)	$51.57_{\pm 0.51}$	$48.95_{\pm 0.15}$	$60.86_{\pm 0.24}$
+QR	$51.28_{\pm 0.44}$	$48.97_{\pm 0.13}$	$60.63_{\pm 0.30}$
Gabburo et al. (2023)	$50.96_{\pm 0.41}$	$48.84_{\pm 0.24}$	$60.48_{\pm 0.32}$
+QR	$\mathbf{52.01_{\pm 0.34}}$	$\mathbf{49.14_{\pm 0.11}}$	$\mathbf{61.02_{\pm 0.30}}$

Table 1: Results on QRC test set.

Setting	ROC AUC	Accuracy	F1
Public ckp	$96.92_{\pm 0.05}$	$\mathbf{91.56_{\pm 0.28}}$	$87.81_{\pm 0.28}$
PRE-TRAINING TECHNIQUES			
QR (our)	$\mathbf{97.05_{\pm 0.03}}$	$91.37_{\pm 0.11}$	$\mathbf{87.86_{\pm 0.25}}$
QQR	$96.63_{\pm 0.07}$	$91.55_{\pm 0.16}$	$87.76_{\pm 0.27}$
MLM	$96.78_{\pm 0.06}$	$91.06_{\pm 0.14}$	$87.05_{\pm 0.20}$
RTS	$96.81_{\pm 0.04}$	$91.22_{\pm 0.14}$	$87.42_{\pm 0.16}$
STS	$94.42_{\pm 0.22}$	$87.61_{\pm 0.38}$	$82.43_{\pm 0.32}$
ALL	$97.00_{\pm 0.09}$	$91.35_{\pm 0.60}$	$87.20_{\pm 0.12}$
DISTILLATION APPROACHES			
Hinton et al. (2015)	$92.14_{\pm 0.65}$	$90.74_{\pm 0.69}$	$86.59_{\pm 1.15}$
+QR	$92.94_{\pm 0.65}$	$90.52_{\pm 0.43}$	$86.59_{\pm 0.61}$
Gabburo et al. (2023)	$97.01_{\pm 0.07}$	$91.67_{\pm 0.12}$	$87.95_{\pm 0.05}$
+QR	$\mathbf{97.20_{\pm 0.20}}$	$\mathbf{91.77_{\pm 0.12}}$	$\mathbf{88.05_{\pm 0.05}}$

Table 2: Results on Quora-match test set.

q/q/a triplets as *[CLS] query [SEP] answer [SEP] question [EOS]*. Based on preliminary experiments, we observed that encoding triplets with this structure is the most effective way to train the model for question ranking. This strategy was also confirmed by Campese et al. (2023). The learning rate ($\{1, 2\}e^{-\{5,6\}}$) and batch size ($2^{\{5,6,7,8\}}$) were selected through grid search by monitoring the loss on the validation set. All fine-tuning experiments were repeated 5 times, results were averaged across different runs.

4.4 Results

Tables 1 and 2 show the performance of our proposed solution and other baselines on QRC and Quora-match respectively.

The QRC table shows that previous pre-training techniques, such as MLM, RTS, STS, and ALL do not improve the performance of the Public checkpoint (ckp) first row, which is fine-tuned on QRC. In contrast, our QR PT improves P@1 by +1.05% (statistically significant through t-test, p-value=0.0005) and halved the standard deviation computed across multiple runs, leading to better model stability. Query Question Rank (**QQR**) is a PT approach using the original query together the top 5 q/a pairs from the retrieval. The drops of 0.83% in P@1 suggest that the query reduces the complexity of the pre-training task, preventing the model to learn meaningful concepts shared by the different question candidates. The two distillation approach by Hinton et al. (2015) improves P@1 by 0.75% (statistically significant, p-value=0.0299). This indicates ranking knowledge can improve the performance on the downstream task. Finally, the retrieval knowledge only works when combined with a weighting approach with QR, producing the best performance (+1.19% P@1), suggesting that distillation from retrieval is less accurate than our PT task. Regarding Quora-match, the Table 2 shows a similar trend: First, other PT tasks do not significantly affect the downstream performance. Second, the combination of distillation (Gabburo et al., 2023) and QR PT achieves the best performance, +0.28% ROC-AUC (statistically significant, p-value=0.0161). The improvements are lower because our approach is specific for question ranking, while Quora is a classification task. Also the baseline models already achieve ceiling performance (e.g., ∼97%).

5 Conclusion

We introduced a novel PT technique to improve models for question ranking tasks. Thus consists in distilling knowledge from a question retrieval model through unsupervised data generation. Our experiments show a clear improvement on two different benchmarks. We will release our code, generated data, and models[3] to support future research on this topic.

6 Limitations

We have proposed a task-specific PT approach that helps improving the performance on question ranking tasks. However, the same approach can hardly be adapted to other different tasks, limiting possible applications.

In our experiments, we generated a ranking data to pre-train models by using a dense retrieval which consists of 33M parameters only, and we distill this knowledge into models of 110M parameters. In other words, the teacher model is 3 times bigger than the student. Although larger teacher models can intuitively boost the performance further, their training can be quite challenging. The training of the MiniLM to generate the ranking data required

[3]We will make our repository available after review

18 days on an AWS EC2 p4dn instance, with a cost of 32$ per hour, making the entire approach expensive. Larger models can increase significantly the cost. As alternative, we could generate ranking data through available LLM directly instead of training a specialized model. However, we estimated that generating the same amount of data we used in our experiments, i.e. 18M queries with 5 ranked q/a pairs each, through Mistral 7B (Jiang et al., 2023) or Falcon 7B (Penedo et al., 2023) required approximately 1500 hours on the same machine, making the entire process infeasible.

References

Stefano Campese, Ivano Lauriola, and Alessandro Moschitti. 2023. Quadro: Dataset and models for question-answer database retrieval. *arXiv preprint arXiv:2304.01003*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020a. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020b. Pre-training transformers as energy-based cloze models. *arXiv preprint arXiv:2012.08561*.

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.

Luca Di Liello. 2023. Structural self-supervised objectives for transformers. *arXiv preprint arXiv:2309.08272*.

Luca Di Liello, Matteo Gabburo, and Alessandro Moschitti. 2021. Efficient pre-training objectives for transformers. *arXiv preprint arXiv:2104.09694*.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022a. Paragraph-based transformer pre-training for multi-sentence inference. *arXiv preprint arXiv:2205.01228*.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022b. Pre-training transformer models with sentence-level objectives for answer sentence selection. *arXiv preprint arXiv:2205.10455*.

Matteo Gabburo, Siddhant Garg, Rik Koncel-Kedziorski, and Alessandro Moschitti. 2023. Learning answer generation using supervision from automatic question answering evaluators. *arXiv preprint arXiv:2305.15344*.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium (ADCS)*, ADCS '15, pages 3:1–3:8, New York, NY, USA. ACM.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. Gooaq: Open question answering with diverse answer types. *arXiv preprint arXiv:2104.08727*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the*

Association for Computational Linguistics, 7:453–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

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, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

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*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116.

Fei Tan, Yifan Hu, Changwei Hu, Keqian Li, and Kevin Yen. 2020. TNT: Text normalization based pre-training of transformers for content moderation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4735–4741, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

YuChen Wang and Li Li. 2023. Generating question-answer pairs for few-shot learning. In *International Conference on Artificial Neural Networks*, pages 414–425. Springer.

Zizhen Wang, Yixing Fan, Jiafeng Guo, Liu Yang, Ruqing Zhang, Yanyan Lan, Xueqi Cheng, Hui Jiang, and Xiaozhao Wang. 2020b. Match²: A matching over matching model for similar question identification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 559–568, New York, NY, USA. Association for Computing Machinery.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. Data augmentation for BERT fine-tuning in open-domain question answering. *CoRR*, abs/1904.06652.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zeyu Zhang, Thuy Vu, Sunil Gandhi, Ankit Chadha, and Alessandro Moschitti. 2022. Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4707–4711.

A Dense retrieval training

Starting from a public checkpoint of MiniLM-v2-12L, 33M parameters, we continuously pre-trained it on a plethora of datasets for unsupervised Sentence Text Similarity (STS) tasks, including paraphrasing, sentence similarity, question answering, and summarization to name a few. Some of these datasets are MSMARCO (Nguyen et al., 2016), Natural Questions (Kwiatkowski et al., 2019), The Semantic Scholar Open Research Corpus (Lo et al., 2020), PAQ (Lewis et al., 2021), AmazonQA (Gupta et al., 2019), WikiHow (Koupaee and Wang, 2018), and many others. A comprehensive list can be found on the web[4]. Overall, these resources contain more than ≈ 0.9B semantically related text pairs.

Similarly to previous work on dense retrieval training, e.g. SentenceBERT (Reimers and Gurevych, 2019), we consider a simple pre-training

[4] https://www.sbert.net

task where the model predicts if two texts are semantically equivalent or not. The model was pre-trained with mixed precision (FP16), Symmetric MultipleNegativesRanking loss (Henderson et al., 2017), learning rate of 2e-5, batch size of 1536, and max sequence length of 128 tokens.

After pre-training, the model is fine-tuned on QRC. Our best configuration, selected through model selection, is based on MultipleNegativesRanking and Online Contrastive losses, learning rate of 5e-6 and batch size of 32.

B Examples of generated data

Table 3 shows some examples of data generated by our dense retrieval model to build the pre-training task. For each of the 4 query examples, we show the top k=5 retrieved similar questions. Intuitively, a human can understand most of the generated ranks. Typcally, the top ranked question is very similar to the input query, whereas questions back in the rank, although still equivalent to the input query, can have a different shape or minor modifications. For instance, *"How old is the Sun?"* is equivalent, as it expresses the same intent, to *"Who long has the sun existed?"*, but the latter adds extra complexity to the original query. The same concept holds for *"What is a cucumber?"* compared to *"What is the definition of cucumber?"*. Other cases have wider discrepancy. For instance *"How many calories in a pineapple?"* is not equivalent to *"How many calories are in a serving of pineapple?"* as the latter asks for a serving, not the entire fruit.

By swapping the top ranked with other associated questions, we can create virtually infinite amount of challenging training examples that can help the training of question-ranking models. Note that our pre-training task does not consider the query as input. Thus, the model sees the rank only and tries to infer the original query before understanding the correct rank.

	How many calories in a pineapple?
1	How many calories are in an pineapple?
2	How many calories in a whole pineapple?
3	How many calories does a pineapple have?
4	How many calories are in a serving of p.?
5	How many calories are in a piece of a p.?
	How many calories in a banana?
1	How many calories in a banana?
2	How many calories are in a banana?
3	How many calories are are in a banana?
4	How many calories does a banana have?
5	How many calories does a banana contain?
	How old is the sun?
1	How old is the Sun?
2	How old is sun?
3	How old can the Sun be?
4	What is the approximate age of the sun?
5	How long has the sun existed?
	What is a cucumber?
1	What are cucumbers?
2	What is cucumber mean?
3	Tell me what is cucumbers?
4	What does cucumber mean?
5	What is the definition of cucumber?

Table 3: Examples of generated data