A Weak Supervision Approach for Predicting Difficulty of Technical Interview Questions

Arpita Kundu, Subhasish Ghosh, Pratik Saini,
Tapas Nayak and **Indrajit Bhattacharya**
TCS Research, India
{arpita.kundu1, g.subhasish, pratik.saini, nayak.tapas, b.indrajit}@tcs.com

Abstract

Predicting difficulty of questions is crucial for technical interviews. However, such questions are long-form and more open-ended than factoid and multiple choice questions explored so far for question difficulty prediction. Existing models also require large volumes of candidate response data for training. We study weak-supervision and use unsupervised algorithms for both question generation and difficulty prediction. We create a dataset of interview questions with difficulty scores for Deep Learning and use it to evaluate SOTA models for question difficulty prediction trained using weak supervision. Our analysis brings out the task's difficulty as well as the promise of weak supervision for it.

1 Introduction

For effective technical interviewing, it is important to know the question difficulty — the probability of a student from a cohort, e.g. senior undergraduate CS students, correctly answering the question. We address the problem of predicting the difficulty of interview questions for candidate cohorts.

Predicting difficulty from the question statement, answer choices and related documents has been studied for multiple choice or factoid questions for reading comprehension and exams (Wang et al., 2014; Huang et al., 2017; Pado´, 2017; Qiu et al., 2019; Benedetto et al., 2020; Yaneva et al., 2020; Benedetto et al., 2021; Cheng et al., 2021; Byrd and Srivastava, 2022). All publicly available datasets (Benedetto et al., 2021; Cheng et al., 2021; Yaneva et al., 2020; Qiu et al., 2019) also contain multiple choice or factoid questions. The nature of technical assessment questions in interviews is different. These look to assess knowledge and understanding rather than memorization of facts and are more open-ended. Answers are long-form, typically spanning 2-5 sentences.

Existing approaches, particularly recent deep models (Xue et al., 2020; Qiu et al., 2019; Benedetto et al., 2021), require large volumes of candidate response data to train the models. This is a challenge when creating a question bank for a new domain or a subject, since field tests need to be performed with real students. In contrast, we explore training question difficulty prediction models using weak supervision based on subject textbooks and Bloom's Taxonomy. This removes dependence on candidate responses and answer assessment.

We explore various strategies of creating weakly-supervised training data. Weak supervision has been explored extensively for many NLP tasks (Lison et al., 2020; Ratner et al., 2020; Ren et al., 2020; Awasthi et al., 2020). For question difficulty, the training data requires not just difficulty scores but interview questions as well. We explore pre-trained large language models (GPT3) and template-based algorithms for generating training questions. We then assign difficulty to these questions using an unsupervised algorithm that uses subject textbooks and Bloom's Taxonomy (Bloom, 1956; Anderson and Krathwohl, 2014). While Bloom's Taxonomy has been used extensively in computer educational testing (Masapanta-Carrión and Ángel Velázquez-Iturbide, 2018; Duran et al., 2018) and for *analysis* of difficulty for short answer questions (Pado´, 2017), but not in predictive models.

For evaluation, we create a dataset of interview questions with difficulty scores from an authoritative textbook on Deep Learning. We use this to evaluate the performance of state-of-the-art QDE models (Benedetto et al., 2020, 2021) when trained using weak-supervision. Our analysis highlights both the challenges of the task as well as the promise of weak-supervision for it.

Our contributions in this paper are as follows. (a) We motivate and introduce the task of difficulty prediction for technical interview questions and curate a dataset for this task. (b) We explore various forms of weak-supervision for this task and analyze the performance of state-of-the-art mod-

Proceedings of the 29th International Conference on Computational Linguistics, pages 4537–4543
October 12–17, 2022.

els. (c) We propose an unsupervised algorithm for question difficulty prediction based on text-book structure and Bloom's Taxonomy. Aside from use in weak supervision, we show that this performs competitively on its own.

2 Dataset

We created a dataset for evaluating interview question difficulty prediction. We made this dataset publicly available[1]. We focus on Deep Learning and use the book "Deep Learning" by Courville et. al. available freely online. First, annotators familiar with technical interviewing and Deep Learning generate interview questions from different chapters of this book. A chapter was given to 2 annotators who reached agreement over validity of generated questions for use in interviews.

Next, we needed to annotate these questions for difficulty on a scale of 1-10. We define higher (lower) difficulty as indicating lower (higher) probability of getting the correct answer from a candidate who has studied this book, and does not have any other exposure to this subject. Attempts to directly annotate difficulty of individual questions led to very low inter-annotator agreement. Instead, we annotated *relative difficulty* for a pair of questions with 3 possible labels: (a) Q1 MORE DIFFICULT, (b) Q2 MORE DIFFICULT and (c) EQUALLY DIFFICULT/EASY. We introduced a *difficulty explanation* label for individual questions in a pair. Possible values were (i) lot of pre-req, (ii) little pre-req, (iii) lot of mathematics, (iv) little mathematics, (v) well-highlighted answer, (vi) hard-to-find answer, (vii) about fundamental concept(s), (viii) about niche concept(s), and (ix) other. Annotators were advised to decide the pair-wise label considering the explanations for the two questions.

The final dataset has 150 unique questions from 16/20 chapters of the book. The questions are well distributed over cognitive tasks (Sec.3.2) and templates (Sec.3.1). 360 question pairs were selected for annotation after running our unsupervised difficulty prediction algorithm (Sec.3.2) to ensure non-triviality of the pair-wise decision. There were 30 unique annotators and each pair was annotated by 5 annotators. After the first round, inter-annotator agreement was 0.23 Fleiss Kappa (fair), and 60/360 questions had a tie. These were broken by 2 additional annotators. The final distribution over labels is 100 Q1, 130 Q2 and 130 EQUAL.

[1]Dataset and Customized definition of BT are publicly available at https://github.com/kunduarpita2012/Technical_question_difficulty_prediction.git

3 Weak Supervision

In this section, we address weak supervision (WS) approaches for question difficulty prediction. WS has been extensively explored for various NLP tasks. One specific challenge is that the training dataset needs not only difficulty scores for questions, but also the questions. Generation of questions is also expertise intensive and gold-standard questions are small in volume. Therefore, WS needs to generate both questions and difficulty scores.

3.1 Question Generation

To generate questions, we explore two different unsupervised approaches: (a) a pre-trained LLM (GPT3), and (b) a template-based algorithm.

GPT3 Questions: Recently, GPT3 (Brown et al., 2020) has been used for weak supervision for many NLP tasks, including question generation from context and answers (Wang et al., 2021). We use prompting with the GPT3 Interview Question preset to generate interview questions from book contexts. In the GPT3 prompt, we provide a context (part of a section) from the book, followed by a new line and an instruction — "Generate a list of questions from the above passage". This was arrived at via experimentation. This process generates diverse questions, but questions are sometimes imprecise in different ways, such as the context not containing the answer, and incompleteness.

Template Questions: To generate more precise questions of types commonly seen in interviews, we use template-based question generation (Puzikov and Gurevych, 2018; Fabbri et al., 2020; Yu and Jiang, 2021). We use the following templates: WHAT IS X?, DEFINE X., EXPLAIN X., WHAT ARE BENEFITS/ADVANTAGES/DISADVANTAGES OF X?, COMPARE X AND Y. For each template, we use precise regular expressions with dictionaries to check its applicability for a sentence. We use a concept dictionary constructed using the book index to detect occurrences of X in sentences.

3.2 Unsupervised Q. Difficulty Prediction

We now describe our unsupervised algorithm for assigning difficulty $d(q)$ to a question q. It as-

signs *context difficulty* $d^c(q)$ considering the specific part of the book from which the question is generated. It also assigns *(cognitive) task difficulty* $d^t(q)$ involved in answering the question considering Bloom's Taxonomy. The overall difficulty of the question is obtained by combining the two: $d(q) = wd^c(q) + (1-w)d^t(q)$.

Context Difficulty: Intuitively, questions from later parts of the book, and similarly later parts of a chapter / section / subsection, are likely to have more dependencies on earlier parts, and are therefore more difficult. We use the chapter no. n^0, section no. n^1 and subsection no. n^2 of a context c to assign a context difficulty score: $d(c) = \sum_{l=0}^{2} w(l)d(n^l; l)$. $w(l)$ is the weight of level l, and we use weights 1, 0.1 and 0.01 for chapters, sections and subsections respectively. The intuition behind the level weights is that two questions generated from two different chapters which are farther apart, are likely to have a greater gap between their difficulty scores than two questions generated from two different sections within a chapter. This intuition similarly extends to subsections within sections. $d(n^l; l)$ is the difficulty associated with level number n^l for level l. So that numbers closer to the end have higher difficulty, we define $d(n^l; l) = n^l / n^l_{max}$, where n^l_{max} is the maximum n^l for a level l.

Task Difficulty: Bloom's Taxonomy (BT) is a well-known resource for determining complexity of educational and assessment tasks. The *Cognitive Process* dimension of BT has levels of cognitive ability, namely REMEMBER, UNDERSTAND, (e.g., explain, classify), APPLY, ANALYZE, EVALUATE and CREATE, and has action verb dictionaries for each level. We first manually assign difficulty scores $d(l)$ to BT levels l. Then the task difficulty $d^t(q)$ of a question q is scored as $d^t(q) = \sum_l \text{sim}(q, l)d(l)$.

To customize BT for interviews, we enrich the taxonomy levels. To each level, we add a list of WH words, and a list of question templates (Sec.3.2). We made this resource public as well[1].

For $\text{sim}(q, l)$, we embed the question q and the BT level l appropriately and compute their cosine similarity. We perform POS tagging and dependency parsing on the question using Spacy. For *verb similarity*, we embed question verbs and level verbs using word2vec (Mikolov et al., 2013) and take the max pair-wise similarity. For *template similarity*, we templatize the question by masking verbs and objects, embed the question template and level templates using pre-trained sBert (Reimers and Gurevych, 2019), and take the max similarity. For *wh similarity*, we check for existence of the question wh word in the level. These three are weighted equally to get $\text{sim}(q, l)$.

Weakly Supervised Training: While there are various weak-supervision frameworks for NLP tasks (Lison et al., 2020; Ratner et al., 2020; Ren et al., 2020; Awasthi et al., 2020), we explore a simple mechanism where we fine-tune a deep model over a weakly labeled training set generated using an unsupervised model. Recent papers have shown that deep models trained using such weak supervision are able to outperform the unsupervised algorithm on the test set (Dehghani et al., 2017; Yu et al., 2021). We plan to explore more sophisticated weak supervision frameworks in future work.

4 Experiments and Analysis

In this section, we report our experiments on the interview question difficulty dataset. We test the usefulness of the following aspects for weak-supervision (WS): (a) difficulty scores predicted by our unsupervised algorithm, (b) algorithm-generated questions, and (c) questions from a related subject.

Model	Q.Subj.	Micro F1
R2DE	DL	0.50
TrQDE[-]	DL	0.51
TrQDE[DL]	DL	**0.54**
TrQDE[DL]	DL+ML	0.53
TrQDE[DL+ML]	DL+ML	0.525
UQDP	-	0.51

Table 1: Comparison of WS types. Q.Subj. indicates subject of questions in training data: deep learning (DL), machine learning (ML). Test questions are on DL. TrQDE[X] indicates TrQDE with MLM fine-tuned on book for subject X. UQDP is unsupervised algorithm for difficulty prediction. Micro-avg F1 is the maximum over threshold θ

Ftr	All	C	T	Tt	Tv	Tw
M. F1	**0.51**	0.50	0.43	0.43	0.37	0.37

Table 2: Ablation for unsupervised difficulty prediction algo. UQDP on test data. C: w/ only context difficulty, T: w/ only all aspects of task difficulty. Tt, Tv, Tw indicate template, verb, wh similarity for task difficulty.

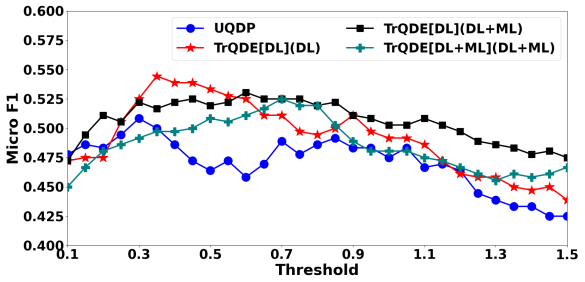

Figure 1: Performance vs threshold θ for best models

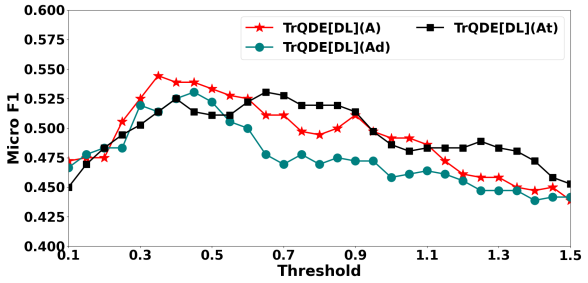

Figure 2: Performance vs threshold θ for question generation algorithms for TrQDE[DL]

All experiments were run on an A100 20GB server. We used Adam with batch-size 16, learning rate 1E-5, dropout rate 0.5 and 50 epochs.

We evaluated two state-of-the-art models for difficulty prediction for factoid / MCQ questions. R2DE (Benedetto et al., 2020) regresses on questions to predict difficulty. We consider the *ques_only* version, since we do not use answers. It uses tf-idf representation of the questions. We report performance for linear regression as the regression model, which was the best. TrQDE (Benedetto et al., 2021) uses transformers to represent the question, with a final regression layer. It fine-tunes the transformer MLM layer using the question corpus, and then further fine-tunes it for the regression task. We report performance for DistilBERT, which worked better than BERT. Here too, we used the *q only* setting. For the unsupervised difficulty prediction algorithm UQDP, we use $w = 0.8$ for combining context and task difficulty.

The primary WS training data covered all 20 chapters of the Deep Learning book, and had 2536 questions (GPT3:1647, Template:889). The secondary WS training data covered 20 chapters from "Pattern Recognition and Machine Learning" (Bishop), also available online. This had 2218 questions (GPT3:1268, Template: 950).

Since the test set has relative difficulty labels, EQUAL is predicted when the difference between a model's predicted difficulty scores for the two questions in a pair is less than or equal to a threshold θ.

The main results are shown in Tab.1. First, WS using UQDP generated difficulty scores for algorithm generated questions improves performance beyond that achieved by using UQDP alone for MLM-fine-tuned versions of TrQDE. This shows the usefulness of both aspects (a) and (b). However, R2DE and the TrQDE with just regression-layer fine-tuning cannot beat UQDP. Next, we analyze aspect (c). Note that UQDP scores difficulty of the DL questions in the training data using the DL book and those of the ML questions using the ML book. Still, including ML questions to train the regression layer does not help, even after including the ML book to fine-tune the MLM layer. The most likely explanation is that the test questions and difficulties are from DL. Including ML questions changes the train distribution, even though the subjects are quite related. Fine-tuning the MLM-layer fits the altered training distribution more closely, leading to poorer results in test.

In Fig.1, we show how micro F1 varies across threshold θ for the 3 best models. This reveals a more nuanced picture. While peak performance of DL-only training is higher, including ML questions in training results has more stable gains across θ values. However, including the ML book for MLM fine-tuning results in worse performance than both.

We investigate aspect (b) further in Fig.2 by plotting performance vs θ when training using different question generation algorithms. We see that performance is the best when using both template and GPT3 generated questions. But, interestingly, templates have better performance individually than GPT3 across θ values. This is very likely because template questions, though smaller in volume and lacking diversity, better mimic the human interview questions seen in test.

Tab.1 showed that UQDP itself has competitive performance on the test set, outperforming R2DE and TrQDE w/o MLM fine-tuning. We investigate aspect (a) further in Tab.2 by performing ablation over different UQDP features. We see that context similarity makes the most significant contribution but adding task similarity improves performance slightly. The contributions of verb and wh similarity are limited compared to template similarity.

5 Error analysis

We now report results of error analysis for our best performing model TrQDE[DL](DL).

The question pairs in the data belong to 3 groups. (A) Same-task-different-context: the questions belong to the same BT level, but are from different book contexts and are about unrelated concepts. (B) Same-context-different task: the questions are from the same context or about related concepts, but belong to different BT levels. (C) Different-context-different-task: the questions are about unrelated concepts / different context and belong to different BT levels. In our labeled data, 36%, 18% and 46% are from groups A, B and C respectively. The third group is the most challenging for relative difficulty labeling. This is so even for human annotators. The Fleiss Kappa scores for inter-annotator agreement are 0.23, 0.27 and 0.21 respectively. Note that while the tasks (question templates) are labeled by annotators when annotating question, while concepts of a question are obtained by eliminating stopwords, wh-words and prepositions using NLTK libraries.

We analyze difficulty prediction errors for each group separately. The errors are of two types. OL-1 (Ordinal Loss 1) errors occur when the predicted and true relative difficulty differ by 1, i.e. the predicted (or true) label is EQUALLY DIFFICULT/EASY and the true (or predicted) label is Q1 MORE DIFFICULT or Q2 MORE DIFFICULT. OL-2 (Ordinal Loss 2) errors occur when the predicted and true labels are the two extremes, i.e. the predicted (or true) label is Q1 MORE DIFFICULT and the true (or predicted) label is Q2 MORE DIFFICULT.

Overall, $\sim$ 32.5% of the predictions of TrQDE[DL](DL) correspond to OL-1 errors and $\sim$ 13% to OL-2. Fig.3 shows a group-wise drill-down. First, we observe that the total error is highest for group C, as expected, as is OL-2 error, demonstrating that it is the hardest group. Between groups A and B, total error is slightly higher for B, indicating that predicting task-difficulty is a bigger challenge than context difficulty.

Deeper analysis provided further insights into the prediction errors of TrQDE. One of these stems from an underlying assumption for UQDE that context difficulty is higher for later parts of a book. However, concepts introduced earlier are often revisited in later chapters in the context of related concepts. UQDE assigns context difficulty incorrectly in such cases and corrupts training data for TrQDE.

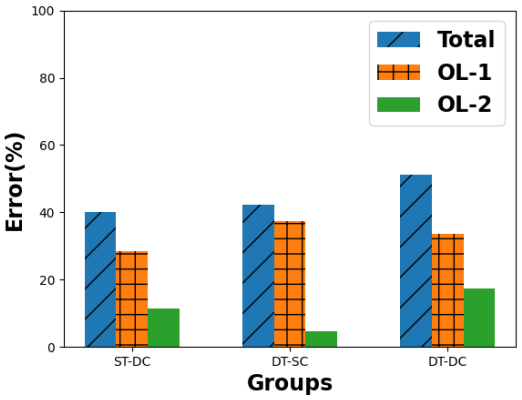

Figure 3: Prediction error of TrQDE[DL](DL) for different groups of question pairs where ST-DC, DT-SC and DT-DC represent Same-task-different-context, Same-context-different task and Different-context- different-task groups respectively.

For example, graphical models and their types are first introduced when Structured Probability Models are introduced in 3.14 under Probability and Information Theory, and discussed again when discussing Graphs for Model Structure in 16.2 Structured Probabilistic Models for Deep Learning. As a result, the question 'What are the different categories of graphical models?' from 16.2 incorrectly gets assigned a higher difficulty level. Other than this, we observe that while TrQDE learns from UQDE-assigned levels in general, sometimes it makes an incorrect prediction for test question pairs where UQDE makes the correct prediction.

6 Conclusions

In summary, we have motivated the task of difficulty prediction for technical interview questions and curated a dataset for evaluation. We have shown that weak-supervision using algorithm-generated questions and an unsupervised difficulty scoring algorithm is a promising direction for fine-tuning related state-of-the-art models for this task. The simple unsupervised algorithm itself shows competitive performance and hints at aspects that new models for this challenging problem will need to consider.

References

Lorin Anderson and David Krathwohl. 2014. *A taxonomy for learning, teaching and assessing: A revision of Bloom's*. Addison Wesley Longman.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and

Sunita Sarawagi. 2020. Learning from rules generalizing labeled exemplars. In *In International Conference on Learning Representations (ICLR)*.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the Tenth International Conference on Learning Analytics and Knowledge (LAKE)*.

Luca Benedetto, Paolo Cremonesi, Giovanni Aradelli, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications (ACL)*.

Benjamin Bloom. 1956. *The taxonomy of educational objectives, the classification of educational goals, volume Handbook I: Cognitive Domain*. David McKay.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of 34th Conference on Neural Information Processing Systems (NeurIPS)*.

Matthew Byrd and Shashank Srivastava. 2022. Predicting difficulty and discrimination of natural language questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (AACL-IJCNLP)*.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the SIGIR*.

Rodrigo Duran, Juha Sorva, and Sofia Leite. 2018. Towards an analysis of program complexity from a cognitive perspective. In *Proceedings of the ACM Conference on International Computing Education Research (ICER)*.

Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, Aliaksandr Hubin Pierre Lison, Jeremy Barnes, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Susana Masapanta-Carrión and J. Ángel Velázquez-Iturbide. 2018. A systematic review of the use of bloom's taxonomy in computer science education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of 27th Conference on Neural Information Processing Systems (NeurIPS)*.

Ulrike Pado´. 2017. Question difficulty – how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (ACL)*.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation (ICNLG)*.

Zhaopeng Qiu, Xian Wu, and Wei Fan. 2019. Question difficulty prediction for multiple choice problems in medical exams. In *Proceedings of Conference on Information and Knowledge Management (CIKM)*.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason A. Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: rapid training data creation with weak supervision. *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB) Endowment*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. 2020. Denoising multi-source weak supervision for neural text classification. In *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Quan Wang, Jing Liu, Bin Wang, and Li Guo. 2014. A regularized competition model for question difficulty estimation in community question answering services. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP*.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications at Annual Meeting of the Association for Computational Linguistics (ACL)*.

Victoria Yaneva, Le An Ha, Peter Baldwin, and Janet Mee. 2020. Predicting item survival for multiple choice questions in a high-stakes medical exam. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC)*.

Xiaojing Yu and Anxiao Jiang. 2021. Expanding, retrieving and infilling: Diversifying cross-domain question generation with flexible templates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pretrained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.