Know your tools well: Better *and* faster QA with synthetic examples

Anonymous ACL submission

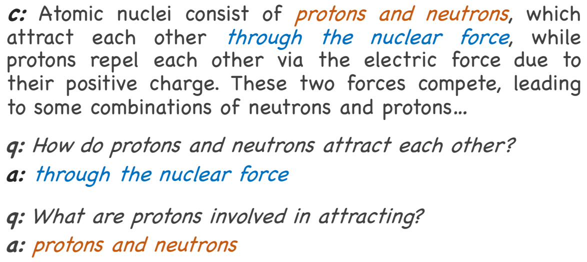
Abstract

Synthetic training data—commonly used to augment human-labeled examples in supervised learning—are often noisy, but can be generated in very large quantities and diversity. This paper proposes to leverage these unique attributes in a targeted manner to maximize the utility of synthetic examples. Via two novel applications that utilize synthetic data for targeted pre-training and knowledge distillation, we demonstrate the feasibility of this idea for machine reading comprehension (MRC). Using our proposed methods, we are able to train simultaneously ***smaller***, ***faster*** and ***more accurate*** MRC models than existing synthetic augmentation methods. Our methods are generic in nature and can be applied to any task for which synthetic data can be generated.

1 Introduction

With the proliferation of large, data-hungry deep neural models, synthetic training data are an increasingly useful utility in machine learning (Nikolenko, 2019, 2021). In real-world NLP applications like question answering (QA), synthetic data can augment training examples created by human domain experts (Dong et al., 2019; Zhang and Bansal, 2019; Sultan et al., 2020) or even train useful models on their own in new domains (Reddy et al., 2021; Puri et al., 2020).

The process of creating synthetic examples for tasks like QA often relies on sequence-to-sequence generation methods (Du et al., 2017; Zhang and Bansal, 2019; Shakeri et al., 2020), and has been greatly simplified by modern unified approaches to language generation (Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020). Figure 1 shows two question-answer pairs generated by BART (Lewis et al., 2020) when fine-tuned on labeled machine reading comprehension (MRC) data.



c: Atomic nuclei consist of *protons and neutrons*, which attract each other *through the nuclear force*, while protons repel each other via the electric force due to their positive charge. These two forces compete, leading to some combinations of neutrons and protons...

q: *How do protons and neutrons attract each other?*
a: *through the nuclear force*

q: *What are protons involved in attracting?*
a: *protons and neutrons*

Figure 1: Synthetic MRC examples generated by our generator from a Wikipedia paragraph. The first example is accurate, but the second one is not.

Not much has been reported in the literature, however, on efforts to understand the data that are thus generated, their properties, and best utilization. In supervised learning settings where human-annotated 'gold' training instances are also available, synthetic data are most commonly used to *pre-train* models, which are then fine-tuned with the gold examples (Dong et al., 2019; Alberti et al., 2019; Sultan et al., 2020). This is a reasonable approach, where noisy synthetic pre-training effectively provides gold fine-tuning with a strong initialization of model parameters. We believe, however, that a more careful consideration of the unique set of properties synthetic data possess and attempts to directly capitalize on those can further increase their utility.

Consider for example the fact that synthetic examples can be generated in numbers that are orders of magnitude larger than gold examples. This creates a large pool of novel test cases on which the behavior of an existing model can be studied, potentially informing the design of a better model. Based on this general idea, we propose a novel synthetic data augmentation method in this paper. Termed *targeted synthetic pre-training*, our method first identifies cycle-consistent examples (Alberti et al., 2019) T in a large synthetic corpus S on which an existing gold-trained model m exhibits high prediction loss. Intuitively, T embodies the weaknesses of m and should complement its gold training well. On two public MRC benchmarks

SQuAD2.0 (Rajpurkar et al., 2018) and NewsQA (Trischler et al., 2017), we empirically show that pre-training with a much smaller subset T of such "hard" synthetic examples indeed yields better models than the original larger corpus S, also drastically reducing training time.

Being able to elicit the detailed behavior of an existing model can be key also for a second machine learning framework: knowledge distillation (KD). We posit that *large amounts* of *diverse* synthetic data—generated using top-p top-k sampling (Holtzman et al., 2020; Sultan et al., 2020), for example—can reveal the knowledge of a teacher model in greater detail than typically limited amounts of gold data. By relying on the teacher's soft predictions as the only targets for supervision, KD can also completely bypass the label noise typically present in synthetic data (see Figure 1). Our evaluation on SQuAD2.0 and NewsQA shows that large synthetic corpora can in fact distill better students than the original gold training sets. Impressively, distilling with both synthetic and gold examples yields students (BERT-Base, 110M parameters) that perform at least as well as their teachers (BERT-Large, 340M parameters) on both datasets.

An advantage of our two proposed methods is that their combination is quite straightforward: targeted synthetic pre-training can be used to first train a strong teacher model, from which a student model can then be distilled using synthetic and gold examples. On SQuAD2.0 and NewsQA, this combination yields BERT-Base MRC models that outperform the best BERT-Large models we train using existing synthetic training methods by 0.6–1.4 points. These results represent a **68% reduction in model size** enabling **3x faster inference**, along with **significant improvements in accuracy** on two separate benchmarks.

The above findings have clear and major implications for real-world applications of QA. Moreover, our proposed approaches are generic in nature, with broad applicability to other NLP tasks.

2 Related Work

From early rule-based approaches that relied on syntactic transformations or handcrafted semantic templates (Heilman and Smith, 2010; Lindberg et al., 2013; Mazidi and Nielsen, 2014), automatic question generation from text has gradually transitioned to using neural sequence-to-sequence methods (Du et al., 2017; Harrison and Walker, 2018; Zhu et al., 2019; Gu et al., 2021). Most state-of-the-art generators also benefit from large-scale language model (LM) pre-training (Dong et al., 2019; Scialom et al., 2019; Shakeri et al., 2020).

Synthetic training data have already been applied with great success to MRC (Duan et al., 2017; Sachan and Xing, 2018; Shakeri et al., 2020; Pan et al., 2021; Bartolo et al., 2021). Most prior work has focused on improving the quality of generation, measured by metrics such as generation accuracy (Liu et al., 2020; Dong et al., 2019) and diversity (Sultan et al., 2020; Yue et al., 2020). While a few strategies including pre-training (Dhingra et al., 2018), cycle consistency filtering (Alberti et al., 2019) and mixing with gold examples in training mini-batches (Zhang and Bansal, 2019) have been proposed to deal with noise in synthetic training data, little or no effort has been made to exploit their unique strengths. Here we intend to explore new training strategies that make better use of *abundant* and *diverse* synthetic examples.

In active learning, a key procedural step is the identification of informative unlabeled examples that are used to query an oracle (Konyushkova et al., 2017; Zhang and Chaudhuri, 2015; Lee et al., 2020). Among different existing *query strategies*, uncertainty sampling (Lewis and Gale, 1994; Scheffer et al., 2001; Wang et al., 2017; Gal et al., 2017) selects examples for which the model being trained is the least certain about what the label should be. In targeted synthetic pre-training, we share the goal of identifying the most useful training examples, but instead of querying an oracle based on model *uncertainty*, we sample from a pool of synthetic examples based on model *error*.

A second related approach to our work is coreset selection, which attempts to find a representative subset of examples that adequately approximates a larger dataset (Har-Peled and Kushal, 2005; Huggins et al., 2016; Coleman et al., 2020; Ju et al., 2021). While our goal is also to identify a useful subset of examples, instead of approximating the entire synthetic dataset, we intend to find a subset that augments human-annotated examples well.

Lastly, knowledge distillation (KD) (Hinton et al., 2015) has proven to be an effective approach to constructing small yet high-performance models by training them to imitate larger teacher models (Liu et al., 2019a; Sun et al., 2019; Yang et al., 2020; Boreshban et al., 2021). For pre-trained LMs, previous work has shown that KD during LM pre-training (Sanh et al., 2019; Dong et al., 2021) or target task supervision (Turc et al., 2019; Li et al., 2021) or both (Jiao et al., 2020) can improve performance. Yang et al. (2020) propose a two-stage KD framework where large amounts of question-passage pairs are derived from a commercial web search engine to create a binary relevance judgment task. Student models are first distilled on this auxiliary task and then with target task gold examples. Unlike all these approaches, in synthetic KD, we exploit synthetic target task (MRC) examples for improved distillation.

3 Methods

In this section, we first discuss fine-tuning of BERT for MRC (§3.1). Then we detail the generation of synthetic examples (§3.2) as well as our proposed methods of targeted synthetic pre-training (§3.3) and synthetic knowledge distillation (§3.4).

3.1 MRC Training

Following Devlin et al. (2019), we fine-tune a transformer-based pre-trained masked language model (MLM) for MRC. This section provides an overview of the procedure; we refer the reader to (Devlin et al., 2019) for further details.

Let c be a context, q a question, and a its answer in c. Let a_{start} and a_{end} be the start and end offsets of a in c. The input to the MRC system is the concatenation of q and c, separated by a special separator token. The MRC network consists of two fully connected feed-forward networks atop shared MLM transformer layers, which learn to predict the start and end probabilities $p_s(a_{start}|q, c)$ and $p_e(a_{end}|q, c)$, respectively.

Given a training dataset D, the parameters θ of the MRC model are learned using standard maximum likelihood estimation:

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \sum_{i=1}^{|D|} \log p_\theta(a_i \mid q_i, c_i) \\ &= \arg\max_{\theta} \sum_{i=1}^{|D|} \big\{ \log p_{s,\theta}(a_{start,i}|q_i, c_i) \\ &\qquad\qquad + \log p_{e,\theta}(a_{end,i}|q_i, c_i) \big\}\end{aligned}$$

At inference time, the model outputs the answer span $[j : k]$ such that:

$$\begin{aligned} j &= \arg\max_{j'} p_{s,\theta}(j' \mid q, c) \\ k &= \arg\max_{k' \geq j} p_{e,\theta}(k' \mid q, c)\end{aligned}$$

3.2 Generating Synthetic Examples

We fine-tune an encoder-decoder language model (Lewis et al., 2020) with examples of answerable questions from existing MRC datasets. Let c be a paragraph in a given document d, q a question, and a its answer in c. Let s be the sentence in c that contains a. Our generator is trained to output the sequence $s; a; q$ given c, where (i) special tokens separate the three texts, and (ii) instead of the full sentence s, only its first and last words are generated for efficiency. In essence, this training procedure teaches the generator to identify an appropriate answer sentence s in c, find a candidate answer phrase a in s, and generate a question q, all in a single autoregressive episode.

Given a dataset D of answerable MRC examples, the parameters ϕ of the generator are learned using standard maximum likelihood estimation:

$$\phi^* = \arg\max_{\phi} \sum_{i=1}^{|D|} \log p_\phi(s_i, a_i, q_i \mid c_i)$$

At inference time, given a paragraph c from a document d, we first generate a triple (s, a, q) using top-p top-k sampling (Holtzman et al., 2020; Sultan et al., 2020). An answerable example (c, q, a) is then created from this output. To create an unanswerable example for q, we simply pair it up with a different paragraph c' in d (Alberti et al., 2019), which results in the triple $(c', q,$ *"No Answer"*$)$. We show some generated examples in Appendix A.1 of the supplementary material.

3.3 Targeted Synthetic Pre-Training

Synthetic Pre-Training To remove noisy examples from the generated MRC dataset, we apply a cycle consistency filter (Alberti et al., 2019) that utilizes an existing MRC model trained on human annotated examples. This filter removes any example from the synthetic dataset for which the MRC model predicts a different answer than the one in the example. Let S be the set of cycle-consistent synthetic examples and G be a given set of gold-standard (*i.e.*, human annotated) examples. Similar to prior work (Dong et al., 2019; Sultan et al., 2020), we follow a two-step process for the application of S in conjunction with G: (1) pre-train an MRC model on S, and (2) fine-tune on G. In the rest of this section, we denote this model by $\theta_{S \rightarrow G}$

and an identical network fine-tuned only on gold data by θ_G.

Targeted Synthetic Pre-Training Given cycle-consistent synthetic training examples S, standard pre-training uses the entire set S to maximize the amount of supervision. In this section, we propose an alternative approach, where a subset $S' \subseteq S$ that explicitly encodes the weaknesses of θ_G is identified, facilitating targeted supervision of a model $\theta_{S' \to G}$ that is superior to $\theta_{S \to G}$.

Concretely, we propose *highest error* synthetic pre-training (HE), where S' consists of those examples in S for which model θ_G has the highest prediction errors; we empirically show in §5 that this S' provides more effective pre-training than S. To systematically study this effect, we first define an example difficulty function $H\colon S \to \mathbb{R}_{\geq 0}$, which computes the conditional negative log-likelihood of example $s = (c, q, a) \in S$ given θ_G:

$$H(s) = -\log p_{\theta_G}(a \mid q, c)$$

Let S_t be the examples in S sorted in decreasing order of their H values:

$$S_t = sort(S, \text{key=}H, \text{order=decreasing})$$

To examine the relationship between the difficulty of synthetic examples and their pre-training effectiveness, we partition S_t into consecutive bins of equal size b. Let n be the number of bins so that $b = \frac{|S_t|}{n}$. For $i \in \{1, 2, ..., n\}$, the i-th bin B_i consists of examples $S_t[(i-1) \times b : i \times b]$ in a randomized order. For each bin $B_i \subset S_t$, we train an MRC model $\theta_{B_i \to G}$ and evaluate the pre-training effectiveness of subset B_i based on the performance of $\theta_{B_i \to G}$ on a test set.

3.4 Synthetic Knowledge Distillation

Synthetic data are prone to label noise, and any erroneous sample in a synthetic training corpus can be detrimental for standard MLE training (§3.1), especially if validation measures such as cycle consistency (CC) check are not taken. Knowledge distillation (KD) (Hinton et al., 2015), on the other hand, can ignore such noisy labels altogether, instead relying on the soft predictions (probability distributions over possible answers) of a stronger teacher model. In the context of synthetic training, KD can thus have a useful denoising effect without requiring measures such as CC.

Here we further posit that KD can synergistically benefit from the use of synthetic training data, which can be generated in large quantities and diversity when an appropriate decoding algorithm such as top-p top-k sampling is used (Sultan et al., 2020). KD aims to uncover the knowledge of a teacher model across a range of input scenarios; we believe that large amounts of diverse synthetic examples, despite their noisy nature, can achieve this objective more effectively than limited amounts of human annotated data.

To test this hypothesis, we perform synthetic KD as follows. Given a training example (c, q, a) in synthetic dataset S, let L be the number of tokens in the corresponding MRC input (q, c) (see §3.1 for details). Let z^t_{start} and z^t_{end} be the answer start and end probability distributions over all L positions of the input sequence, respectively, as predicted by the teacher model. Similarly, let z_{start} and z_{end} be the distributions predicted by the student. We compute a distillation loss $\mathcal{L}_{distill}$ based on the Kullback-Leibler divergence from z to z^t as follows:

$$\mathcal{L}_{distill,start} = \sum_{i=1}^{|S|} D_{KL}(z^t_{start,i} \parallel z_{start,i})$$

$$\mathcal{L}_{distill,end} = \sum_{i=1}^{|S|} D_{KL}(z^t_{end,i} \parallel z_{end,i})$$

$$\mathcal{L}_{distill} = \frac{1}{2}(\mathcal{L}_{distill,start} + \mathcal{L}_{distill,end})$$

We train the student model by minimizing $\mathcal{L}_{distill}$:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{distill}$$

At inference time, prediction follows the same procedure as in §3.1.

4 Experimental Setup

Here we describe our general experimental setup. Additional details specific to individual experiments are provided in §5.

4.1 Datasets

We use two public MRC benchmark datasets: SQuAD2.0 (Rajpurkar et al., 2018) and NewsQA (Trischler et al., 2017). The documents in SQuAD2.0 are Wikipedia pages, while NewsQA contains CNN news articles. The official Test set for SQuAD2.0 is not publicly available; thus, we use the official Dev set as our Test set and a random split of the original training set as Train and Dev. For NewsQA, we use the official Train-Dev-Test split. Selected key statistics for the two datasets are provided in Table 1.

	Train	Dev	Test
SQuAD2.0			
# of Documents	397	45	35
# of Paragraphs	17,081	1,954	1,204
# of Examples	117,159	13,160	11,873
NewsQA			
# of Documents	11,469	638	637
# of Examples	107,669	5,988	5,971

Table 1: Dataset statistics. Examples are aligned to paragraphs in SQuAD2.0, but not in NewsQA.

	SQuAD2.0	NewsQA
Total Answerable	7.6M	12.4M
Total Unanswerable	1.9M	3.1M
CC Answerable	5.0M	4.8M
CC Unanswerable	1.8M	3.0M
CC Answerable in SYN	4.0M	4.0M
CC Unanswerable in SYN	1.0M	1.0M

Table 3: Counts of synthetic examples. CC: cycle-consistent; SYN: final set of synthetic pre-training examples used in our experiments.

4.2 Models

We fine-tune BART-Large (Lewis et al., 2020) for synthetic example generation. In our synthetic pre-training experiments, BERT-Large models (340M parameters) (Devlin et al., 2019) are fine-tuned for MRC. For knowledge distillation, we use BERT-Large teachers and BERT-Base (110M parameters) students. All model implementations are based on Hugging Face (Wolf et al., 2019).

4.3 Synthetic Example Generation

We train separate generators for SQuAD2.0 and NewsQA on the corresponding answerable training examples. For generation, we use top-p top-k sampling with $p = 0.9$ and $k = 10$. Given a generated answerable example (c, q, a), an unanswerable example is created by pairing up q with a randomly sampled context $c' \neq c$ from the same document (Alberti et al., 2019).

We generate synthetic examples (§3.2) for both datasets from in-domain documents. For SQuAD2.0, these are Wikipedia pages taken from the Natural Questions dataset (Kwiatkowski et al., 2019). For NewsQA, we use two different sources: (1) CNN articles (Hermann et al., 2015) that are not in NewsQA, and (2) New York Times (NYT) articles in the Gigaword corpus (Graff et al., 2005). See Table 2 for detailed statistics.

Many CNN and NYT paragraphs are relatively short; we merge such paragraphs to create contexts that are around 320 word pieces long. Longer paragraphs are used as is. For SQuAD2.0, individual paragraphs are used as contexts. We generate five examples per context for SQuAD2.0 and three per context for NewsQA, and remove all duplicates. We also create one-fourth as many unanswerable examples as answerable ones for each dataset.

	Wikipedia	CNN+NYT
# of Documents	307,373	1,333,316
# of Paragraphs	1,812,843	4,841,721

Table 2: Sizes of unlabeled corpora from which we generate synthetic examples. Wikipedia pages are used to generate examples for SQuAD2.0; CNN and New York Times (NYT) articles are used for NewsQA.

For cycle consistency check, instead of using the MRC model we are trying to improve, we propose to utilize a different, more powerful model. While other options exist, such as an ensemble of different models, we simply train RoBERTa-Large (Liu et al., 2019b) MRC models on the respective gold datasets as our cycle consistency checkers. For each dataset, we finally retain a random sample of 4M answerable and 1M unanswerable cycle-consistent examples to use in our experiments. Table 3 shows the statistics.

		SQuAD2.0	NewsQA
Batch Size		12	24
LR	BERT-Base	$3{\times}10^{-5}$	$3{\times}10^{-5}$
	BERT-Large	$3{\times}10^{-5}$	$2{\times}10^{-5}$
# of Epochs	GOLD HL	2	1
	SYN → GOLD HL	1 → 2	1→ 1
	GOLD DT	6	4
	SYN → GOLD DT	3 → 6	3 → 2

Table 4: MRC training configurations. LR: learning rate; HL: hard label training (§3.1 and §3.3); DT: soft distillation training (§3.4).

4.4 MRC Training

Training configurations for both SQuAD2.0 and NewsQA are shown in Table 4. These hyperparameter values were derived using a grid search; we choose the set of values that yield the best Dev results.

4.5 Knowledge Distillation

For each dataset, we select as our teacher model the respective best performing BERT-Large model from §3.3, trained using highest error synthetic pre-training.

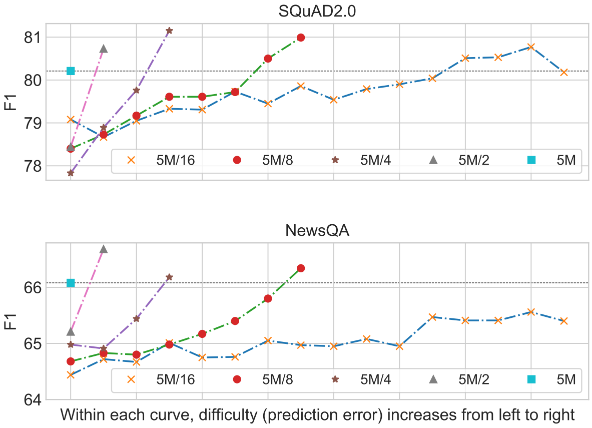


Figure 2: Dev set performance of synthetic pre-training bins. Along the x-axis, each curve places all bins of a specific size equidistantly and in increasing order of difficulty. Bin sizes across curves range from $\frac{5\text{M}}{16}$ to $\frac{5\text{M}}{2}$ examples. The square represents all 5M examples. Effectiveness of the bins (measured by F_1 score) generally increase with their difficulty, and several high-error bins outperform all 5M examples.

	SQuAD2.0		NewsQA	
	Dev	Test	Dev	Test
GOLD	79.2	80.9	64.3	63.2
SYN 5M → GOLD	80.2	82.4	66.1	65.1
SYN HE → GOLD	**81.1**	**83.3**	**66.7**	**65.2**

Table 5: Performance (F_1 scores) of BERT-Large using gold-only training, pre-training with all 5M synthetic examples, and with subsets of highest-error synthetic examples (HE) that have the best Dev results.

4.6 Evaluation

Following Rajpurkar et al. (2018), we use macro-averaged F_1 score as our evaluation metric, which measures the average lexical overlap between system predictions and ground truth answers. Each model is first evaluated on Dev; the answerability threshold that maximizes the model's Dev F_1 score is then used to obtain its Test predictions for the final evaluation on Test (Devlin et al., 2019).

5 Results and Analysis

All results reported in this section are average scores over three random seeds.

5.1 Targeted Synthetic Pre-Training

Following the procedure of §3.3, we first partition the 4M cycle-consistent *answerable* examples of Table 3 into difficulty bins of 2M (2 bins), 1M (4 bins), 500K (8 bins) and 250K (16 bins) examples. The 1M *unanswerable* examples are also hierarchically partitioned into the same number of bins. Answerable and unanswerable bins of the same rank (*e.g.*, 2nd of 8) are then merged to construct our final pre-training bins. For each bin B, we train a BERT-Large MRC model $\theta_{B \rightarrow G}$ (§3.3).

Figure 2 illustrates how these models perform on the the SQuAD2.0 and NewsQA Dev sets. For bin sizes between $\frac{5\text{M}}{8}$ and $\frac{5\text{M}}{2}$ examples (inclusive), performance consistently improves from easier bins (examples with smaller prediction losses) to harder bins (examples with larger losses). The trend is similar for the smallest $\frac{5\text{M}}{16}$-example bins, but slightly weaker. Crucially, several high-error bins of different sizes outperform all 5M examples. These results provide empirical evidence that harder synthetic examples generally yield better pre-training than easier examples.

In addition to the difficulty of the pre-training bins, a second key independent variable in the above experiment is their size. The best-performing bins for SQuAD2.0 and NewsQA contain $\frac{5\text{M}}{4}$ and $\frac{5\text{M}}{2}$ examples, respectively, which implies that even though the hardest examples are the most useful, enough of them must still be included in pre-training for sufficient sample diversity.

Table 5 compares the two best Dev models with baselines on the respective Test sets. While synthetic pre-training with all cycle-consistent examples does improve results over gold-only training, *targeted* synthetic pre-training with only a high-error subset yields the best results across the board. In a one-tailed Wilcoxon signed-rank test of the difference between the two pre-training methods, we observe $P{<}.05$ for SQuAD2.0 (Dev and Test) and $P{<}.1$ for NewsQA Dev. On NewsQA Test, targeted pre-training outscores standard all-example pre-training in each of the three independently randomized runs, even though we observe $P{>}.1$ in the significance test.

We also compare with a curriculum learning (CL) baseline (Bengio et al., 2009), which trains models in an easy-to-hard order so that the hardest examples are used at the end. CL is similar to our proposal in that it also aims to exploit differences in the difficulty of examples, but unlike our method, it uses all available training examples. Additionally, we randomize the order of examples within our individual difficulty bins before training (§3.3)—a decision that was made based on our initial experimental results on the Dev sets—which is not a feature of CL.

	SQuAD2.0		NewsQA	
	Dev	Test	Dev	Test
Plain CL	78.2	80.0	65.7	64.6
CL with 5% Switch	78.8	80.8	65.8	64.4
Random Training Order	80.2	82.4	66.1	65.1
SYN HE → GOLD	**81.1**	**83.3**	**66.7**	**65.2**

Table 6: Performance (F_1 score) of the curriculum learning (CL) baseline is worse than both a random training order and our targeted pre-training method.

Table 6 summarizes the CL results. First, we observe that CL actually underperforms a random difficulty order of the training examples, which is our primary baseline in Table 5 (row 2). According to some prior studies, introducing some harder examples early in CL can be useful (Platanios et al., 2019; Penha and Hauff, 2019); hence we also examine a variation of CL where positions of 5% of the examples are randomly switched. This model performs slightly better than plain CL overall, but is still considerably worse than the random order baseline. One possible interpretation of these results is that in CL, training with only easy examples in the beginning might lead the models' weights to a region in the parameter space that subsequently makes their generalization hard.

5.2 Synthetic Knowledge Distillation

In our knowledge distillation (KD) experiments, we use the best BERT-Large models of §5.1 as the teachers: SYN $\frac{5\text{M}}{4}$ HE → GOLD for SQuAD2.0 and SYN $\frac{5\text{M}}{2}$ HE → GOLD for NewsQA (see Table 5). All students are BERT-Base models.

Given a training dataset D, KD can generally use a combination of a distillation loss such as $\mathcal{L}_{distill}$ from §3.4 and a standard negative log-likelihood loss based on the labels from D:

$$\mathcal{L}_{labels} = -\sum_{i=1}^{|D|} \log p_\theta(a_i \mid q_i, c_i)$$

To find out the best combination for our models, we first train students on the gold training data for $\lambda \in \{0, .3, .5, .7, .9, 1\}$ in the following joint loss:

$$\mathcal{L} = \lambda \mathcal{L}_{distill} + (1 - \lambda)\mathcal{L}_{labels}$$

SQuAD2.0 models are trained for six epochs and NewsQA models for four epochs (tuned on Dev).

The best F_1 scores on Dev are achieved with $\lambda = 1$ (SQuAD2.0: 77.4, NewsQA: 66.0), where training discards the dataset labels entirely, relying only on the teacher's predictions. On the other hand, MLE training on the dataset labels ($\lambda = 0$) has the lowest F_1 scores (SQuAD2.0: 74.7, NewsQA: 62.2), indicating that any amount of KD is useful. Based on these results, we use only $\mathcal{L}_{distill}$ in all later experiments.

	SQuAD2.0		NewsQA	
	Dev	Test	Dev	Test
Teacher	**81.1**	**83.3**	**66.7**	**65.2**
$\lambda = 1$	77.4	78.7	66.0	64.8
$\lambda = 0$	74.7	75.8	62.2	60.6

Table 7: F_1 scores from gold-only distillation baselines. Pure distillation ($\lambda = 1$) is clearly better than MLE on the the dataset labels ($\lambda = 0$), but none of the students are as good as their teachers.

To establish a gold-only distillation baseline, we further evaluate the best Dev students ($\lambda = 1$) on the respective Test sets. The results are shown in Table 7. There is a clear performance gap between the teachers and their students in these results, which can either mean that the students have reached their full capacity and cannot perform any better, or that a limited amount of gold data cannot expose the teachers' knowledge in enough detail to train better students.

# of Examples	SQuAD2.0	NewsQA
1.25M	79.0	65.0
2.5M	80.0	65.7
5M	80.8	66.1
7.5M	**81.1**	**66.1**

Table 8: Synthetic distillation F_1 score on Dev consistently improves with number of training examples.

To find out which of these two explanations is correct, next we distill students with synthetic examples, which, unlike human annotations, can be produced in very large numbers. We randomly sample subsets of 1.25M, 2.5M, 5M and 7.5M synthetic examples and distill a separate student with each for one epoch. Table 8 shows the evaluation results on Dev, where we observe a clear improvement in performance as the number of training examples increases. Crucially, the 5M synthetic distillation results are already better after a single epoch of training than the gold distillation results of Table 7. These results support the hypothesis that larger training datasets yield better KD due to increased sample diversity.

Our best student models on Dev are obtained by distilling with 7.5M synthetic examples for two more epochs and then with gold examples (SQuAD2.0: six epochs, NewsQA: two epochs).

	SQuAD2.0		NewsQA	
	Dev	Test	Dev	Test
Teacher	81.1	83.3	66.7	65.2
DST^*_G	77.4	78.7	66.0	64.8
DST^*_S	81.5	83.0	66.5	65.5
$\text{DST}^*_{S\to G}$	**81.7**	**83.8**	**66.7**	**65.7**

Table 9: Final synthetic distillation F_1 scores. DST^*_G: top student with gold-only distillation; DST^*_S: top student with synthetic-only distillation; $\text{DST}^*_{S\to G}$: top student with synthetic followed by gold distillation. $\text{DST}^*_{S\to G}$ students (BERT-Base, 110M parameters) perform at least as well as their teachers (BERT-Large, 340M parameters) on both datasets.

Table 9 summarizes the performance of these models on both Dev and Test. DST^*_S, the student model distilled only with synthetic examples, already matches the teacher's performance on SQuAD2.0 Dev and NewsQA Test. With further gold distillation, $\text{DST}^*_{S\to G}$ actually outscores the teacher in all four test conditions (P<.05 for SQuAD2.0 Dev and Test, P<.1 for NewsQA Test). This small but nevertheless interesting outperformance of the teacher models by their students could be a result of lower student model variances, but further experiments are necessary to validate this hypothesis. In Appendix A.2 of the supplementary material, we demonstrate the effectiveness of synthetic distillation with an even smaller DistilBERT (Sanh et al., 2019) student.

5.3 Better and Faster MRC

The experimental results presented in this section thus far demonstrate the individual utilities of our two proposed methods. This concluding subsection addresses the overarching question of this paper: Do our methods train *better and faster* MRC models than existing synthetic training methods?

To answer the question, we take a closer look at the Test set performance of two systems. The first system is the BERT-Large SYN 5M → GOLD model (340M parameters) of §5.1, which is a state-of-the-art (SOTA) synthetic training baseline that first pre-trains an MRC model with all 5M cycle-consistent synthetic examples and then fine-tunes it with gold examples. The second system is the best student model $\text{DST}^*_{S\to G}$ of §5.2, a 110M-parameter BERT-Base model that is trained by applying our two proposed methods in succession: targeted synthetic pre-training to first train the teacher, followed by both synthetic and gold distillation to train the student.

Method	# Params	SQuAD2.0	NewsQA
SYN 5M → GOLD	340M	$F_1 = 82.4$ (46.2 QPS)	$F_1 = 65.1$ (44.0 QPS)
$\text{DST}^*_{S\to G}$	110M	$F_1 = \mathbf{83.8}$ (**145.3** QPS)	$F_1 = \mathbf{65.7}$ (**139.2** QPS)

Table 10: End-to-end Test set results. The proposed synthetic training methods, when combined, yield better and faster MRC models (bottom) than synthetic pre-training with all cycle-consistent examples (top). QPS stands for questions answered per second (on an NVIDIA Tesla V100 GPU with a batch size of 128).

We report both the accuracy and the inference speed of the two systems in Table 10. These results clearly show that when applied together, our proposed methods train better (significant at P<.05 for SQuAD2.0, P<.1 for NewsQA) and simultaneously 3x faster MRC models than the existing SOTA approach for synthetic training.

It should be noted here that both the teacher and the baseline model above are pre-trained only on cycle-consistent (CC) samples, as examples that are not CC are known to be detrimental for synthetic pre-training (Alberti et al., 2019). Synthetic distillation, on the other hand, can still use the filtered out samples, since the training objective $\mathcal{L}_{distill}$ (§3.4 and §5.2) only looks at the teacher's soft predictions, and not at the noisy synthetic labels. Given a pool of synthetic data, a larger subset can thus be used in practice for distillation (7.5M in our experiments) than for pre-training (5M).

6 Conclusion

This paper poses and explores the question of how synthetic data attributes such as abundance and diversity can be better exploited to improve model supervision. Empirical results with our two proposed applications, namely targeted synthetic pre-training and synthetic knowledge distillation, show that these attributes can indeed be leveraged in new and more targeted ways to: (i) greatly reduce the memory footprint of large transformer-based reading comprehension models, (ii) enabling faster inference, while (iii) also improving their accuracy. Future work will test the limits of the proposed methods, *e.g.*, with ensembles of large models as teachers for distillation. A second important direction is the application of the proposed ideas to other NLP tasks, as they are generic in nature and applicable in principle to any scenario where synthetic training data are available.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. *arXiv:2104.08678*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Yasaman Boreshban, Seyed Morteza Mirbostani, Gholamreza Ghassem-Sani, Seyed Abolghasem Mirroshandel, and Shahin Amiriparian. 2021. Improving question answering performance using knowledge distillation and active learning. *arXiv:2109.12662*.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. 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 *NAACL-HLT*.

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.

Chenhe Dong, Guangrun Wang, Hang Xu, Jiefeng Peng, Xiaozhe Ren, and Xiaodan Liang. 2021. Efficientbert: Progressively searching multilayer perceptron via warm-up knowledge distillation. *arXiv:2109.07222*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *NeurIPS*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. *arXiv:1703.02910*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2005. English gigaword second edition. *Linguistic Data Consortium, Philadelphia*.

Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. 2021. Chaincqg: Flow-aware conversational question generation. *arXiv:2102.02864*.

Sariel Har-Peled and Akash Kushal. 2005. Smaller coresets for k-median and k-means clustering. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, SCG '05, page 126–134, New York, NY, USA. Association for Computing Machinery.

Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306, Tilburg University, The Netherlands. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *ICLR*.

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. 2016. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association* *for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jeongwoo Ju, Heechul Jung, Yoonju Oh, and Junmo Kim. 2021. Extending contrastive learning to unsupervised coreset selection. *arXiv:2103.03574*.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2017. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Ji-Ung Lee, Christian M. Meyer, and Iryna Gurevych. 2020. Empowering Active Learning to Jointly Optimize System and User Demands. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4233–4247, Online. Association for Computational Linguistics.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *SIGIR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lei Li, Yankai Lin, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021. Dynamic knowledge distillation for pre-trained language models. *arXiv:2109.11295*.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria. Association for Computational Linguistics.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus. In *WWW*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding. *arXiv:1904.09482*.

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

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 321–326, Baltimore, Maryland. Association for Computational Linguistics.

Sergey I. Nikolenko. 2019. Synthetic data for deep learning. *arXiv:1909.11512*.

Sergey I. Nikolenko. 2021. *Synthetic Data for Deep Learning*. Springer.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Online.

Gustavo Penha and Claudia Hauff. 2019. Curriculum learning strategies for ir: An empirical study on conversation response ranking. *arXiv:1912.08555*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv:1903.09848*.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training Question Answering Models From Synthetic Data. In *EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *Unpublished manuscript*.

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*.

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, Melbourne, Australia. Association for Computational Linguistics.

Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avirup Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. 2021. Synthetic Target Domain Supervision for Open Retrieval QA. In *SIGIR*.

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. 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 629–640, New Orleans, Louisiana. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *EMC2 at NeurIPS*.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, IDA '01, page 309–318, Berlin, Heidelberg. Springer-Verlag.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6027–6032, Florence, Italy. Association for Computational Linguistics.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. 2020. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. 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 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv:1908.08962*.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. 2017. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. Model Compression with Two-stage Multi-teacher Knowledge Distillation for Web Question Answering System. In *WSDM*.

Xiang Yue, Xinliang Frederick Zhang, Ziyu Yao, Simon Lin, and Huan Sun. 2020. CliniQG4QA: Generating Diverse Questions for Domain Adaptation of Clinical Question Answering. *arXiv:2010.16021*.

Chicheng Zhang and Kamalika Chaudhuri. 2015. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. 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 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. 2019. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4238–4248, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Synthetic MRC Training Examples

In Tables 11 and 12, we show some answerable and unanswerable synthetic MRC examples generated by our SQuAD-trained generator.

Auggie and Me is not a sequel but a companion book to Wonder (although " The Julian Chapter " serves as one). It contains ***three*** stories, each telling the events of Wonder from different perspectives. The first story, called " The Julian Chapter ", is from the point of view of school bully Julian where he explains why he mistreats Auggie and if he will change. The second – called ***Pluto*** – focuses on August Pullman's life before Beecher Prep and is set in the point of view of Christopher, Auggie's oldest friend. The third...

How many stories does the book contain?
What is the name of the second story in the book?

Table 11: Two synthetic answerable questions generated by our SQuAD-trained generator. Answers are highlighted in the context.

Within higher - income families that are sending more children to universities and colleges, women make up a greater percentage (15 % compared to 7 %) of this growth. While the largest gap of educational attainment between men and women is seen in the highest income group, women are attaining higher levels of education than men in every income group. This observation poses a unique and confusing problem : if educational attainment has a positive correlation to familial income, why are more women entering and completing college than men? Bailey and Dynarski proposed that the observed educational gap by gender may be due to differing incentives to accumulate human capital. Men and women may participate in what they term " segregated labor markets " and " asymmetric marriage markets, " and perhaps, to make up for those perceived market differences, females are more motivated to obtain higher levels of education.

What percent did the number of Latinos in K-12 expand between 1999 and 2016?

Table 12: A synthetic unanswerable question used in SQuAD pre-training.

A.2 Experiments with a DistilBERT Student

DistilBERT (Sanh et al., 2019) is a 66M-parameter masked language model (MLM) distilled from BERT, which has shown strong performance relative to its size. To test the effectiveness of our proposed two-stage synthetic training on a smaller student model, we distill a DistilBERT MRC model from the BERT-Large teacher of §5.2 using synthetic and gold MRC examples.

As shown in Table 13, the accuracy of this student is comparable to the BERT-Large model of §5.1, trained using the existing synthetic training method of pre-training with cycle-consistent examples (SYN 5M → GOLD). Crucially, however, the DistilBERT model is about five times smaller, providing a 6x speedup in inference over the BERT-Large model.

Method	**# Params**	**SQuAD2.0**	**NewsQA**
SYN 5M → GOLD	340M	$F_1 = 82.4$ (46.2 QPS)	$F_1 = \mathbf{65.1}$ (44.0 QPS)
$\text{DST}^*_{S\rightarrow G}$	66M	$F_1 = \mathbf{82.8}$ (**270.7** QPS)	$F_1 = 65.0$ (**274.9** QPS)

Table 13: Test set performance of our DistilBERT models compared to the best synthetically pre-trained BERT-Large baselines. The accuracies of the two models are comparable even though the DistilBERT models are about 5x smaller and 6x faster. QPS stands for questions answered per second (on an NVIDIA Tesla V100 GPU with a batch size of 128).

The F_1 scores in Table 13 are average scores over three random seeds. For synthetic → gold distillation of the DistilBERT student, we train for 4 → 2 and 3 → 4 epochs for SQuAD2.0 and NewsQA, respectively. These values were tuned on the respective Dev sets. Batch size and learning rate are the same as in Table 4.