Self-Supervised Test-Time Learning for Reading Comprehension

Pratyay Banerjee **Tejas Gokhale** **Chitta Baral**
Arizona State University
pbanerj6, tgokhale, chitta@asu.edu

Abstract

Recent work on unsupervised question answering has shown that models can be trained with procedurally generated question-answer pairs and can achieve performance competitive with supervised methods. In this work, we consider the task of unsupervised reading comprehension and present a method that performs "test-time learning" (TTL) on a given context (text passage), without requiring training on large-scale human-authored datasets containing *context-question-answer* triplets. This method operates directly on a single test context, uses self-supervision to train models on synthetically generated question-answer pairs, and then infers answers to unseen human-authored questions for this context. Our method achieves accuracies competitive with fully supervised methods and significantly outperforms current unsupervised methods. TTL methods with a smaller model are also competitive with the current state-of-the-art in unsupervised reading comprehension.

1 Introduction

Reading comprehension is the task in which systems attempt to answer questions about a passage of text. Answers are typically found in the passage as text-spans or can be inferred through various forms of reasoning (Rajpurkar et al., 2016). The answer to the following question:

"Who is the President of the United States?"

depends on the timeframe and context of the passage provided, and will be different for news articles written in 2001 vs. 2021. If the context is the script of the TV series "The West Wing", the answer is "Jed Bartlet", and even in this fictional setting, it will later change to "Matt Santos".

Knowledge sources such as Wikipedia get updated when new events occur (such as the outcome of elections), or new facts about the world are revealed (such as scientific discoveries), with contributors adding new information and removing information that is no longer valid (Almeida et al., 2007). With such context-dependent answers and continual changes in knowledge, it is hard to justify training models over fixed corpora for tasks such as question answering (QA). We would like models to answer questions based on the given context and not to learn biases from datasets or historical news articles.

Moreover, supervised learning has been shown to perform poorly in QA tasks with adversarial examples (Jia and Liang, 2017), domain shift (Jia and Liang, 2017; Yogatama et al., 2019; Kamath et al., 2020), and biased or imbalanced data (Agrawal et al., 2018; McCoy et al., 2019). For example, QA systems trained on Wikipedia fail to generalize to newer domains such as Natural Questions (Rennie et al., 2020) or biomedical data (Wiese et al., 2017), and suffer a significant drop in accuracy. Even small semantics-preserving changes to input sentences, such as the substitution of words by synonyms, have been shown to degrade performance in NLP tasks (Alzantot et al., 2018; Jia et al., 2019). Continual changes in text corpora are inevitable, thus calling for the development of robust methods that can reliably perform inference without being subject to biases.

Supervised Question Answering faces challenges such as the need for large-scale (usually human-authored) training corpora to train models. Such corpora typically require significant post-processing and filtering to remove annotation artifacts (Sakaguchi et al., 2020). To address these challenges, some recent methods (Lewis et al., 2019; Li et al., 2020) approach question answering as an unsupervised learning task. A significant advantage of this approach is that it can be extended to domains and languages for which collecting a large-sized human-authored training corpus is challenging. Methods for unsupervised QA procedurally generate a large corpus of *(context, question, answer)* triples, and train large neural language

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1200–1211
June 6–11, 2021. ©2021 Association for Computational Linguistics

models, such as BERT (Devlin et al., 2019).

In this work, we focus on unsupervised reading comprehension (RC) under evolving contexts and present the "Test-Time Learning" paradigm for this task. RC – the task of answering questions about a passage of text, acts as the perfect setting for robust question-answering systems that do not overfit to training data. While large-scale language models trained on large datasets may contain global information, the answer needs to be extracted from the given context. Thus, our work seeks to learn unsupervised reading comprehension without access to human-authored training data but instead operates independently on each test context. This makes our method 'distribution-blind' where each new context is assumed to be a novel distribution. The test-time learning (TTL) framework enables smaller models to achieve improved performance with small procedurally generated question-answer pairs, and is summarized below:

- a single context (text passage) c_i is given, from which we procedurally generate QA pairs;
- these QA pairs are used to train models to answer questions about c_i;
- the inference is performed on previously unseen questions for c_i.

This framework has a simple assumption that every context comes from a distinct distribution. Hence, parameters learned for the previous context might not be useful to generalize to other contexts. This assumption holds where the contexts evolve over time, and rote memorization of answers might lead to wrong predictions. As such, the above process is repeated for each new context c_i.

For question-answer generation, we use simple methods such as cloze-translation (Lewis et al., 2019), template-based question-answer generation (Fabbri et al., 2020) and question-answer semantic role labeling (QA-SRL) (He et al., 2015). We use two neural transformer-based language models, BERT-Large (Devlin et al., 2019) and DistilBert (Sanh et al., 2019), to study the efficacy of our framework with large and small transformer models. We evaluate our method on two reading comprehension datasets, SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2017). We investigate test-time training under multiple learning settings: (1) single-context learning – the "standard" setting, (2) K-neighbor learning – by retrieving top-K multiple related contexts for each test context, (3) curriculum learning – progressively learning on question-types of increasing order of complexity, (4) online learning – sequentially finetuning models on each incoming test sample.

Our experimental findings are summarized below:

- Test-time learning methods are effective for the task of reading comprehension and surpass current state-of-the-art on two benchmarks: SQuAD and NewsQA.
- Online TTL trained over K-neighboring contexts of the test context is the best version with EM/F1 gains of $7.3\%/7.8\%$ on SQuAD 1.1 and $5.3\%/6.9\%$ on NewsQA.
- DistilBERT – which has less than $\frac{1}{5}^{th}$ of the number of model parameters of BERT-Large is competitive with current SOTA methods that use BERT-Large.

2 Test-Time Reading Comprehension

Consider a reading comprehesion test dataset $\mathcal{D}^{test}{=}\{(c_i, q_i, a_i)\}_{i=1}^{n}$ with context text passages c_i, human-authored questions q_i and true answers a_i. The QA model $g(\cdot)$ is parameterized by $\theta = (\theta_f, \theta_h)$ where θ_f are parameters for the feature extractor, and θ_h for the answering head. The answer is predicted as a text-span, given by the start and stop positions $[y_{start}, y_{stop}]$. Contemporary unsupervised RC models (Lewis, 2019; Li et al., 2020) are trained on a large dataset $\hat{\mathcal{D}}^{train}{=}\{(c_i, \hat{q}_i, \hat{a}_i)\}_{i=1}^{n}$, where the QA pairs are synthetically generated from the context.

In our setting, we do not use such large training datasets, but instead directly operate on individual test contexts $c_i \in \mathcal{D}^{test}$. Given c_i, M synthetic question-answer pairs $\{(\hat{q}_i^j, \hat{a}_i^j)\}_{j=1}^{M}$ are procedurally generated as described in Section 3. The QA model parameters θ are trained over the synthetic data to predict the span of the answer $[\hat{y}_{start}, \hat{y}_{stop}]$ by optimizing the loss ℓ_{ans}:

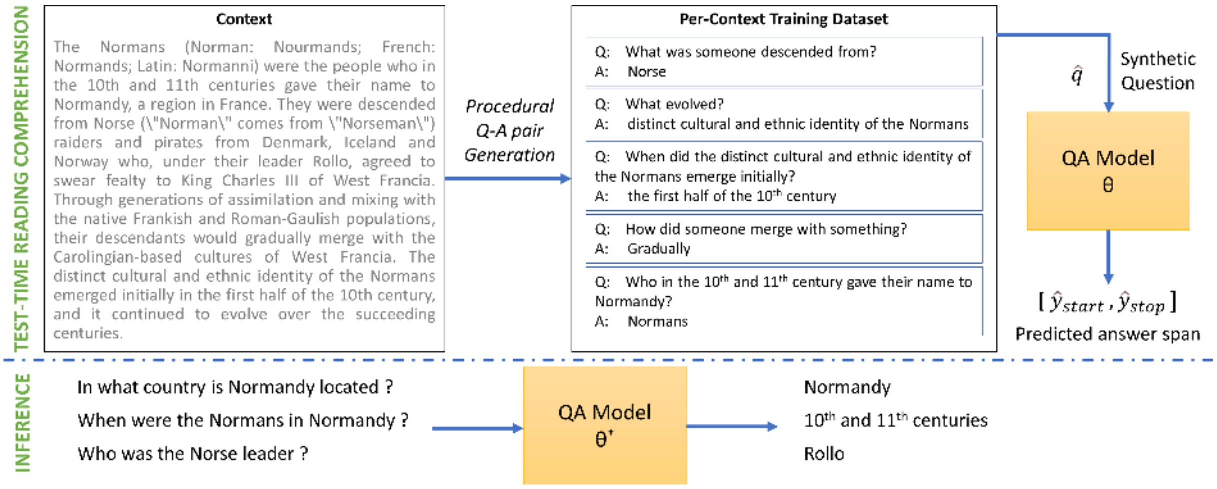
$$\underset{\theta}{\text{minimize}} \sum_{j=1}^{M} \ell_{ans}(c_i^j, \hat{q}_i^j, \theta) \tag{1}$$

$$\ell_{ans} = \ell_{CE}(\hat{y}_{start}, \hat{a}_{start}) + \ell_{CE}(\hat{y}_{stop}, \hat{a}_{stop}) \tag{2}$$

where ℓ_{CE} is cross-entropy loss. The inference is performed on human-authored questions to predict the answer spans:

$$[y_{start}, y_{stop}] = g(c, q). \tag{3}$$

Next, we describe the variants of test-time reading comprehension.


TEST-TIME READING COMPREHENSION
Context
Procedural Q-A pair Generation
Per-Context Training Dataset
$\hat{q}$
Synthetic Question
QA Model θ
$[\hat{y}_{start}, \hat{y}_{stop}]$
Predicted answer span
INFERENCE
QA Model θ*

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse (\"Norman\" comes from \"Norseman\") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Q: What was someone descended from?
A: Norse

Q: What evolved?
A: distinct cultural and ethnic identity of the Normans

Q: When did the distinct cultural and ethnic identity of the Normans emerge initially?
A: the first half of the 10th century

Q: How did someone merge with something?
A: Gradually

Q: Who in the 10th and 11th century gave their name to Normandy?
A: Normans

In what country is Normandy located ?
When were the Normans in Normandy ?
Who was the Norse leader ?

Normandy
10th and 11th centuries
Rollo

Figure 1: Overview of our self-supervised test-time learning framework for reading comprehension. Our method does not require a human-authored training dataset but operates directly on each single test context and synthetically generates question-answer pairs over which model parameters θ are optimized. The inference is performed with trained parameters θ^* on unseen human authored questions.

Single-Context Test-Time RC. This is the standard formulation of test-time learning in this paper, with Equation 1 optimizing over θ, i.e. for each context c_i, the feature extractor θ_f is re-initialized with pre-trained BERT, and the answering head θ_h is randomly initialized.

K-neighbor Test-Time RC. In this version, K contexts similar to the test-context c_i are grouped together, and Equation 1 is optimized over each set of similar contexts as opposed to single contexts in the standard setting. We index contexts in a Lucene-based information retrieval system (Gormley and Tong, 2015) and retrieve top-K similar contexts given c_i, which we call Context Expansion with IR described in Section 3.

Curriculum Test-Time RC. In the curriculum learning version, questions are ordered in increasing order of complexity. We generate different types of questions, such as, semantic role labelling, cloze-completion, template-based and dependency tree-based translation of cloze questions to natural questions. This provides an ordering of complexity and we study the effect of test-time training with such an increasing complexity.

Online Test-Time RC. In the online test-time learning (TTLO), test samples are considered to be encountered in sequence. As such, answering head parameters θ_h are updated sequentially without being randomly re-initialized like in the standard single-context setting. For each new test context c_i, θ_h is initiliazed with the optimal parameteres from the previous test context c_{i-1} to optimize Equation 1.

3 Self-Supervised QA Generation

In this section, we detail our framework for procedurally generating QA pairs from a given context. We use named-entity recognition from Spacy (Honnibal and Montani, 2017), dependency parsing from Berkeley Neural Parser (Stern et al., 2017) and semantic role labeling (He et al., 2015) as our core methods to extract plausible answers and generate natural questions. As described in our task formulation, we create a set of M question-answer pairs $\{(\hat{q}_i^j, \hat{a}_i^j)\}_{j=1}^M$ for the given context c_i.

Cloze Generation. Statements in which the answer is replaced with a mask or blank token are called cloze questions. We follow the steps provided in Lewis et al. (2019) in which answers are replaced with a special token depending on the answer category. For example, in a sentence,

> *"They were descended from Norse raiders and pirates from Denmark"*

the answer *Denmark* is replaced by [LOCATION], resulting a cloze question:

> *"They were descended from Norse raiders and pirates from* [LOCATION]*"*.

Cloze Translation is utilized to rephrase cloze questions into more natural questions by using rule-based methods from Lewis et al. (2019).

Template-based Question Generation utilizes simple template-based rules to generate questions. Given a context of format:

[FRAGMENT A][ANSWER][FRAGMENT B]

a template of the format "Wh+B+A+?" replaces the answer with a Wh-word (e.g., who,what,where) as described in Fabbri et al. (2020).

Dependency Parsing-based Question Generation. In this method, we use dependency reconstruction to translate clozes to natural questions as described in Li et al. (2020), according to the following steps:

1. Right child nodes of the answer are retained and left children are pruned.
2. For each node of the parse tree, if the child node's subtree contains the answer, the child node is moved to the first child node.
3. An in-order traversal is performed on the reconstructed tree. A rule-based mapping is applied to replace the special mask token of the cloze with an appropriate "Wh-word".

QA-Semantic Role Labeling (QA-SRL) was proposed by He et al. (2015) as a method to annotate NLP data, by using QA pairs to specify textual arguments and their roles. As seen in Figure 1, for the context sentences:

> *"They were descended from Norse raiders and pirates from Denmark.",*
> *"The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century and it continued to evolve."*

the following QA pairs were generated,

> *("What was someone descended from?", "Norse"),*
> *(What evolved?, distinct cultural and ethnic diversty)*

We can observe the questions are short and use generic descriptors and pronouns such as *"something"* and *"someone"* instead of specific references calling for the model to have greater semantic understanding of the given context.

Context Expansion using IR is used in the K-neighbor version of TTL. For Context Expansion, we index all paragraphs present in a Wikipedia dump in ElasticSearch. During test-time learning, we preprocess the context c_i by removing the most frequent stop-words, and use it as a seed query to search and retrieve top-K similar contexts. This provides us with related paragraphs that describe similar topics, and consequently more diverse and slightly larger number of QA pairs to train compared to only c_i. We then generate QA pairs using the above described methods. We study the effect of varying the number of most similar contexts (K) on the downstream QA performance.

4 Experiments

Datasets. We evaluate our learning framework on two well-known reading comprehension datasets: SQuAD 1.1 (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2017).

QA Model. We focus on training two transformer-encoder based models, BERT-Large (Devlin et al., 2019) trained with whole-word masking and DistilBERT (Sanh et al., 2019). BERT-Large is used by current state-of-the-art methods on unsupervised extractive QA tasks and has 345 million trainable parameters. On the other hand, DistilBERT is a knowledge-distilled transformer-encoder based model and only has 66 million parameters ($\sim 5\times$ smaller than BERT-Large), allowing us to study the efficacy of TTL with respect to model-size.

Metrics. We use the standard metrics for extractive QA – *macro Exact Match*, where the predicted answer span is directly matched with the ground-truth, and *macro F1*, which measures the overlap between the predicted and the ground-truth spans. For comparisons with existing unsupervised methods, since TTL operates directly on test instances, we report validation set performance only for SQuAD 1.1, as the test set is hidden.

Training Setup. For all test-time learning variants, we limit the maximum number of questions generated per context to 4000 and the maximum number of training steps to 1500. The number of training steps is linearly dependent on the selected batch size $\in [16, 64]$. For our K-neighbor TTL setup that uses Context Expansion, we limit the number of retrieved contexts to 500. In Curriculum Test-Time RC, we ensure that all variants have an equal number (1000) of generated QA-pairs per-context. We evaluate multiple learning rates within the range 1e-5 to 5e-5. We use the Adam (Kingma and Ba, 2014) optimizer and truncate the paragraphs to a maximum sequence length of 384. The number 384 was chosen by evaluating the 99^{th} percentile of the combined length of question and the contexts, to reduce training overhead and GPU memory size. Long documents are split into multiple windows with a stride of 128. All

	SQuAD 1.1		NewsQA	
Models	Dev	Test	Dev	Test
DCR (2016)	62.5 / 71.2	62.5 / 71.0	- / -	- / -
mLSTM (2016)	64.1 / 73.9	64.7 / 73.7	34.4 / 49.6*	34.9 / 50.0*
FastQAExt (2017)	70.3 / 78.5	70.8 / 78.9	43.7 / 56.1	42.8 / 56.1
R-NET (2017)	71.1 / 79.5	71.3 / 79.7	- / -	- / -
BERT-Large (2019)	84.2 / 91.1	85.1 / 91.8	- / -	- / -
SpanBERT (2020)	- / -	88.8 / 94.6	- / -	- / 73.6
DistilBERT (2019)	77.7 / 85.8	- / -	57.2 / 64.8	56.1 / 63.5

Table 1: Results (EM / F1) from supervised methods on SQuAD 1.1 and NewsQA.

	SQuAD 1.1		NewsQA	
Models	Dev	Test	Dev	Test
BERT-Large				
+ Dhingra et al.[†]	28.4 / 35.8	- / -	18.6 / 27.6	18.6 / 27.2
+ Lewis et al.[‡]	45.4 / 55.6	44.2 / 54.7	19.6 / 28.5	17.9 / 27.0
+ Li et al.	62.5 / 72.6	61.1 / 71.4	33.6 / 46.3	32.1 / 45.1
+ Fabbri et al.	46.1 / 56.8	- / -	21.2 / 29.4	- / -
+ our data	49.4 / 59.1	- / -	28.2 / 37.6	27.3 / 36.4
DistilBERT				
+ Lewis et al. data	23.4 / 29.5	- / -	14.1 / 21.6	14.7 / 20.6
+ Li et al. data	42.6 / 48.3	- / -	25.4 / 36.2	27.1 / 35.4
+ Fabbri et al. data	37.5 / 45.6	- / -	16.3 / 22.3	16.1 / 22.9
+ our data	38.9 / 46.8	- / -	23.2 / 31.9	22.4 / 31.1
BERT-Large TTL[★]	**69.8 / 80.4**	- / -	**38.9 / 53.2**	**38.2 / 52.6**
DistilBERT TTL[★]	**58.1 / 68.9**	- / -	**32.6 / 46.4**	**30.5 / 45.2**

Table 2: Comparison with previous unsupervised methods on SQuAD 1.1 and NewsQA. [★]We show the best TTL model here, and results from all TTL variants in Table 3. Metrics are EM / F1. Previous SOTA for both models are shaded in gray. *results from Trischler et al. (2017); [†] Lewis et al. (2019); [‡] Li et al. (2020).

	Default init. θ_f		Pre-trained init. θ_f	
TTL Models	SQuAD 1.1	NewsQA	SQuAD 1.1	NewsQA
BERT-Large				
Single-Context	54.9	34.9	59.8	37.5
Single-Context Online	56.1	36.3	61.8	39.1
K-neighbor	66.2	41.6	78.3	50.7
K-neighbor Online	**68.7**	46.3	**80.4**	**53.2**
Curriculum	68.3	**46.7**	79.7	52.8
All test contexts	64.7	39.8	68.2	43.5
DistilBERT				
Single-Context	37.2	23.2	49.4	34.6
Single-Context Online	38.5	25.3	55.6	39.8
K-neighbor	42.4	27.8	64.3	43.5
K-neighbor Online	**49.7**	**29.1**	**68.9**	**46.4**
Curriculum	49.3	28.7	68.7	45.8
All test contexts	42.4	28.2	47.4	38.7

Table 3: Comparison of Dev-set F1 scores for TTL variants, when θ_f are trained from default initialization for each test instance, or pre-trained on our generated data. Scores surpassing previous best, are shaded in cyan for SQuAD and red for NewsQA.

experiments were conducted on two Nvidia RTX-8000 GPUs. We use ten percent of the training data to perform three hyper-parameter trials for each variant. We train models with three random seeds, and report the mean F1 and EM scores.

Baselines. As we generate our own data using QA-SRL, we use the following strong baselines. First, we train BERT-Large with generated data from previous methods described in Section 3 and our method (which contains additional QA-SRL samples). Second, we replicate the baselines using the low parameter-count model DistilBERT (66 million vs 345 million for BERT-Large). Third, for a fair comparison to Single-Context and K-neighbor test-time learning where we train models for each context independently, we propose a baseline where we train on all the test contexts together, referred to as "All test contexts". We also evaluate all TTL variants on two initializations of feature-extractor parameters –

1. "default" initialization of BERT-Large, i.e. θ_f pre-trained on masked language modeling and next-sentence prediction tasks, and θ_h randomly initialized for each context and trained from scratch, or
2. θ_f and θ_h further pre-trained on $100K$ synthetic QA pairs generated procedurally using our methods described in Section 3 with contexts taken from the Wikipedia corpus.

5 Results and Discussion

5.1 Unsupervised Question Answering

We compare our results with current state-of-the-art supervised methods (Table 1) and unsupervised methods (Table 2) on SQuAD 1.1 and NewsQA. The previous best unsupervised method with both BERT-Large and DistilBERT is Li et al. (2020). Our best TTL method is the Online version (TTLO), with a pre-training phase and a randomly-shuffled ordering of QA pairs with an average of 3000 QA pairs per context, trained with only 100 steps. With this setup, we are able to improve the state-of-the-art for the SQuAD benchmark with BERT-Large by 7.8% exact-match accuracy and 7.3% F1 score. With DistilBERT, the best TTL method shows an improvement of 15.5% EM and 20.6% F1 over DistilBERT-based baseline, as shown in Table 2. In NewsQA, TTL improves BERT-Large performance by 5.3% EM and 6.9% F1 score, and with DistilBERT shows an improvement of 7.2% EM and 7.2% F1 score.

Training BERT-Large and DistilBERT with "our data" i.e. with a combined synthetic corpus created via all four QA-pair generation methods, marginally improves the F1 score. This shows that our QA generation methods lead to an improvement over existing unsupervised QA generation methods as shown in Table 2. However, the TTL framework leads to even larger gains (∼20%

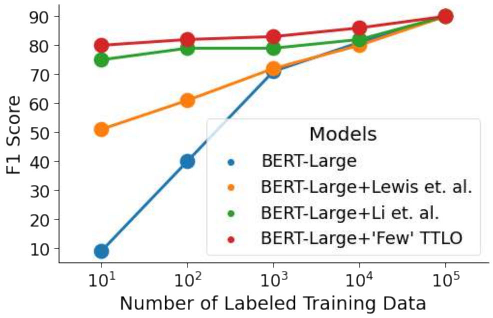


Figure 2: Comparison of F1 scores of TTL models when trained with an increasing number of labeled training samples on SQuAD. TTLO–Online TTL.

for SQuAD and $\sim$10% for NewsQA), indicating the benefits of test-time learning. This result also points to the limits of training with a large number of contexts compared to training on individual contexts. This limitation is especially profound in lower parameter models, such as DistilBERT. In Reading Comprehension, since the answer comes from the context, "understanding" the context is much more relevant. It has a higher inductive bias than learning to comprehend a significantly large number of contexts during training.

For instance, there are multiple contexts about Normans in the SQuAD dataset, one of which is shown in Figure 1. But each context may have different historical persons referred to as the leaders or rulers of the Normans. Answers to questions such as *"Who was the leader of the Normans"* are better learned for each context separately than from all contexts. Pre-training on several contexts is indeed beneficial to obtain better parameter initializations, as observed in Table 2, which can be further independently finetuned for each context during TTL.

5.2 Few-Shot Question Answering

We evaluate our best method under the few-shot setting, i.e. when models are trained with a limited number of human-authored QA pairs from the training datasets. Figure 2 shows a comparison with an increasing number of labeled training samples for SQuAD. TTL-Online is consistently better than existing methods and achieves 81.6% F1 score with just 100 labeled samples. This indicates that this learning framework can reduce the number of in-domain human-authored samples required for training. TTL-Online is also consistently better than (Li et al., 2020) which the previous best unsupervised method for SQuAD. All methods (which use BERT-Large as backbone) converge to similar performance, with an increasing number of additional human-authored samples. This indicates the saturation of the inductive bias that can be incorporated into the architecture using current human-authored annotations.

Curriculum Order	Default init. θ_f		Pre-trained θ_f	
(Left to Right)	SQuAD	NewsQA	SQuAD	NewsQA
BERT-Large				
Random Shuffled	68.7	46.3	80.4	53.2
QA-SRL > T > DP	68.3	46.7	79.7	52.8
T > QA-SRL > DP	67.6	45.4	77.6	50.0
T > DP > QA-SRL	65.8	44.3	75.3	47.2
DistilBERT				
Random Shuffled	49.7	29.1	68.9	46.4
QA-SRL > T > DP	49.3	28.7	68.7	45.8
T > QA-SRL > DP	48.8	28.1	67.2	43.9
T > DP > QA-SRL	47.1	26.5	65.3	39.2

Table 4: Dev-set F1 scores for K-neighbor Online test-time learning, for different Curriculum Learning orderings of QA-SRL (He et al., 2015), T (template-based methods), DP (dependency parsing).

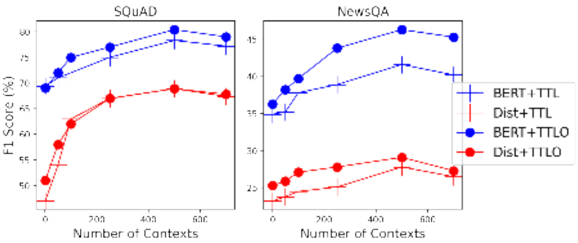


Figure 3: Comparison of F1 scores of TTL models when trained with an increasing number of contexts, on both SQuAD and NewsQA.

5.3 Analysis

We study the different variants of test-time learning and effects of hyperparameters, such as the number of training steps and the number of contexts, on the validation split for both datasets.

Single-Context vs K-neighbor Test-Time RC. In Table 3, we compare all TTL variants. We observe that training with additional contexts has a significant impact on F1 score, compared to training on only the given test context c_i. This may be simply explained as more synthetic training samples from similar contexts leading to a better generalization to human-authored samples. Although similar work in image classification (Sun et al., 2020) and super-resolution (Shocher et al., 2018) show a substantial performance improvement in a single sample learning, we observe that context expansion is beneficial for reading comprehension.

In Figure 3, we vary the number of retrieved

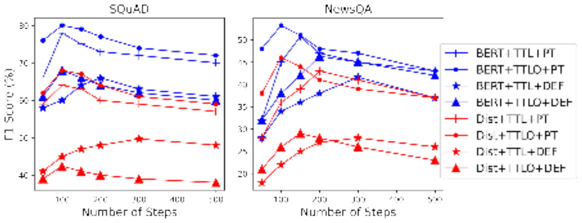


Figure 4: Effect of number of train steps on F1 scores of each TTL model on both SQuAD and NewsQA. PT–Pre-Trained θ_f, θ_h, DEF–Default θ_f, θ_h.

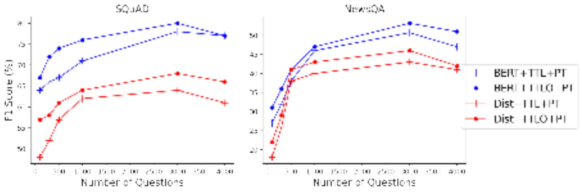


Figure 5: Effect of number of questions on F1 scores of each TTL model on both SQuAD and NewsQA. PT–Pre-Trained θ_f.

neighbors contexts, K, and observe that F1 scores continue to increase till a limit ($\sim$ 500). This is consistent in both BERT-Large and DistilBERT, as well as in the two datasets, SQuAD and NewsQA. Our hypothesis is that there exists an optimal number of QA pairs that the model benefits from, and a maximum threshold on the number of similar contexts after which, the model starts to overfit to the synthetic nature of the QA pairs.

Randomly initialized v/s Pre-trained θ_f,θ_h. We study the effect of re-initializing the question answering head and further pre-training using a set of procedurally generated QA-pairs on downstream test-time learning in Figure 4 and Table 3. While F1 scores achieved without pre-training are comparable to prior methods, pre-training leads to improved performance and also faster convergence, as shown in Figure 4. This can be attributed to better initial weights, which are further finetuned during the test-time learning phase. We studied pre-training with $50k$, $100k$, and $200k$ QA pairs and observed the best performance with $100k$ samples.

Curriculum Test-time learning. In Table 4 we study the effect of curriculum TTL, compared to the baseline of the default random-shuffled QA pairs. Interestingly, using a random ordering rather than a defined curriculum begets the best performance. Among the three curriculum ordering that we utilized, [QA-SRL, TEMPLATE-BASED (T), DP (DEPENDENCY- PARSING-BASED)] was effective but slightly lower than the performance with random ordering. However, training with QA-SRL at the end has a distinctly negative effect. We hypothesize that the model starts to overfit to the shorter vague questions from QA-SRL and “forgets" more natural questions. Hence, it loses generalizability to the human-authored questions.

Online-Test-time Learning. In online test-time learning, the model is continuously self-supervised and evaluated on a continuous stream of contexts and QA-pairs. From Table 3 and Figures 3, 4 and 5, we can observe that TTL-Online consistently outperforms the single-context variant. One key observation is that the model achieves its best performance within 100 training steps (batch size of 48), whereas the base version needs around 300 to 500 steps. This fast adaptation enables a faster inference time, compared to θ_h being trained from scratch. We studied the effect of different random orderings of the test samples and observed the deviation as $\pm 1.6\%$ in F1 scores, which indicates ordering of test samples has a minor effect.

Effect of Batch Size and Learning Rate. Batch-size and learning rate have strong effects on online test-time learning. We observe that resuming with the learning rate of the last epoch of the pre-training with synthetic QA pairs achieves the best F1 scores. We do not use any weight decay. A persistent optimizer state between contexts is critical. Similarly, we hypothesize that the batch-layer normalization statistics pre-computed in transformer encoder layers get updated in further pre-training with QA pairs, leading to a better estimation during TTL. For the base variant of TTL, a higher, fixed learning rate of 3e-5 with a batch size of 32-48 achieves the best F1 scores.

Effect of number of Training steps and QA pairs is studied in Figures 4 and 5. To limit inference time per test context, we observe TTL variants initialized with pre-trained θ achieve the top performance within 150 training steps, whereas those trained with default initialization need $200-300$ steps. In Figure 5, we can observe the variants achieve their best F1 scores around $3k$ QA pairs. This appears consistent with 100 train steps with a batch size of $24-32$. Surprisingly, DistilBERT with pre-trained θ performs equally well compared to BERT-Large with no pre-training on synthetic question-answer pairs.

Effect of TTL on inference time. TTL and its variants all increase the inference time as compared to traditional inference. For the best variant of TTL-Online with BERT-Large, we train for 100 steps with a batch size of 48 samples, which leads to an inference time of $\sim$5 minutes per context. Each context contains, on average $6-7$ questions in SQuaD 1.1 and NewsQA. The best variant of DistilBERT, although has a lower average inference time of 1.6 minutes per context, by employing several engineering tricks, such as saving models on RAM instead of the disk by using `tmpfs` (Snyder, 1990), and using mixed-precision training (Micikevicius et al., 2018). In comparison, non-TTL methods have inference times in the range $\sim$ 10K samples/sec with a GPU hardware of Nvidia V100 16GB. TTL inference time is limited by the current computation power of the GPUs but is potentially remediable. However, with an increase in CUDA cores in GPUs and RAM size, we estimate the inference time can be further improved. Moreover, with newer efficient transformer architectures such as Linformer (Wang et al., 2020) and Big Bird (Zaheer et al., 2020), it is possible for this inference time to be further reduced. It will be an interesting future work to increase TTL's efficiency further while retaining its strength of generalizing to evolving distributions.

Error Analysis. We analyzed 100 wrongly answered samples from SQuAD validation split and observed the model is biased towards answering named-entities. This is not unexpected as most of our QA-pair generation methods are focused on named-entity answers. For example, for the question *"Is it easier or harder to change EU law than stay the same?"*, the TTL DistilBERT model generates *"EU"*, whereas the ground-truth answer is "harder". Although QA-SRL generates more diverse answers, the corresponding questions are vague and much more synthetic, leaving scope for improving QA pair generation to include a variety of question and answer types in the future. Another source of errors is the alternate plausible answers generated by our models, shown in Table 5.

6 Related Work

Extractive QA. The goal for extractive question answering (EQA) is to predict a span of text in a context document as the answer to a question. Various benchmarks have been established to evaluate the capability of EQA models on corpuses from different domains such as Wikipedia-based question answering in SQuAD (Rajpurkar et al., 2016), Natural Questions dataset (Kwiatkowski et al., 2019), as well as questions requiring complex reasoning to extract answers in HotPotQA (Yang et al., 2018); questions about news' articles in NewsQA (Trischler et al., 2017); and about trivia-facts in TriviaQA (Joshi et al., 2017).

Unsupervised QA. For many of the aforementioned extractive QA benchmarks, "human-like" performance has been reached via supervised methods. Unfortunately, these methods do not transfer well to new domains, and the collection of training data in new domains and new languages may not always be feasible. To address this, unsupervised EQA has been proposed as a challenge (Lewis et al., 2019), in which aligned *(context, question, answer)* triplets are not available. Self-supervised data-synthesis methods (Lewis et al., 2019; Banerjee and Baral, 2020; Rennie et al., 2020; Fabbri et al., 2020; Li et al., 2020; Banerjee et al., 2020) have been used for question answering by procedurally generating QA pairs and training models on these synthetic data.

Self-Supervised Learning. The key idea in self-supervision is to design auxiliary tasks so as to and extract semantic features from unlabeled samples, for which input-output data samples can be created from unlabeled datasets. Self-supervision has been used to train large transformer-based language models such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) for the auxiliary task of masked token prediction, and XLNET (Yang et al., 2019) for token prediction given any combination of other tokens in the sequence. ELECTRA (Clark et al., 2019) instead of masking tokens, jointly trains a generator to substitute input tokens with plausible alternatives and a discriminator to predict the presence or absence of substitution. MARGE (Lewis et al., 2020) is trained to retrieve a set of related multi-lingual texts for a target document, and to reconstruct the target document from the retrieved documents. The goal of self-supervised pretext task design is to come up with tasks that are as close to the main task, to learn better representations. In NLP, QA format provides us such an opportunity where we can leverage NER, SRL, Cloze Completion as auxiliary tasks for complex QA.

Learning at test-time. Our work is inspired by image processing methods such as single-image

Question	**Predicted**	**GT**
What can block a legislation?	parliament	majority in parliament
Which TFEU article defines the ordinary legislative procedure that applies for majority of EU acts?	294	TFEU article 294
Who was killed in Dafur ?	Red Cross employee	Red Cross employee dead
Who does the African National Congress say should calm down ?	Archbishop Desmond Tutu	Tutu

Table 5: Error Analysis: Illustration of alternate plausible answers predicted by our models, but regarded as wrong predictions for SQuAD and NewsQA.

super-resolution (Glasner et al., 2009; Freedman and Fattal, 2011; Shocher et al., 2018) that do not require access to external training datasets but instead formulate a self-supervised task for upsampling natural image patches recurring at different scales in the image. Test-time training (TTT) (Sun et al., 2020) for image classification makes use of rotation prediction Gidaris et al. (2018) as an auxiliary task to implicitly learn image classification at test-time and shows improved robustness. While we can directly synthesize main-task data (QA pairs) from the context and do not require an auxiliary task, our work is closely related to TTT.

Domain Adaptation. Pre-training for the tasks such as masked language modeling or other synthetic tasks on unlabeled corpora for a new domain has been evaluated for commonsense reasoning (Mitra et al., 2019) and classification tasks (Gururangan et al., 2020). On the other hand, our work can be viewed as task-specific self-supervision with each new context as a new domain.

7 Conclusion

In this work, we propose test-time learning (TTL) as a new framework for unsupervised extractive question answering (EQA). We present four variants of TTL with a simple but effective context expansion method. We utilize four question-answer pair generation methods for EQA and propose using QA-SRL as an additional source of QA pairs, to supplement prior methods. We show TTL enables "understanding" of contexts at test-time, without human-authored annotations, and significantly improves EQA, including low parameter models.

We envision TTL as a framework that can direct work in reading comprehension to be viewed as a problem of ever-evolving datasets instead of a static corpus. Natural language itself undergoes continuous evolution (Gentner and France, 1988; Traugott and Dasher, 2001; Hamilton et al., 2016) via changes in preference for syntactical structures; creation of new words and phrases; and changing usage frequencies and semantics for existing words. TTL can potentially be applied to such scenarios with semantic drift or domain shift. Further improvements w.r.t. selection of similar contexts for K-neighbor TTL could be explored by leveraging hard sample selection, hard negative mining, bootstrapping, and contrastive learning, along with improved currculum strategies.

Acknowledgements

The authors acknowledge support from the DARPA SAIL-ON program W911NF2020006, ONR award N00014-20-1-2332 and NSF grant 1816039; and thank the reviewers for their feedback.

Ethical Considerations

Our test-time learning method treats every new test instance as a new distribution, and does not rely on a human-authored training dataset. We believe that this is a possible way to avoid learning spurious correlations or linguistic priors, especially when it comes to socio-cultural and historical biases that have been shown to percolate into models for various NLP tasks (Hendricks et al., 2018; Kurita et al., 2019; Sheng et al., 2019). On the other hand, if the test context itself contains biased, false, or propaganda statements, our model will use those statements to extract answers. We would not want models trained on such data to be deployed in the real world. However, because model parameters are randomly initialized for each new context in the standard version of our framework, if contexts are fact-checked by "reliable" sources, then we believe our model will be relatively bias-free, as compared to pre-trained language models for which it is hard to trace *why* a certain prediction was made. Test-time learning allows us to disentangle biases learned from single contexts, from biases learned by language models from large corpora.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980.

Rodrigo B Almeida, Barzan Mozafari, and Junghoo Cho. 2007. On the evolution of wikipedia. In *ICWSM*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162, Online. Association for Computational Linguistics.

Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2020. Self-supervised vqa: Answering visual questions using images and captions. *arXiv preprint arXiv:2012.02356*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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*, pages 4508–4513, Online. Association for Computational Linguistics.

Gilad Freedman and Raanan Fattal. 2011. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):1–11.

Dedre Gentner and Ilene M France. 1988. The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. In *Lexical ambiguity resolution*, pages 343–382. Elsevier.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*.

Daniel Glasner, Shai Bagon, and Michal Irani. 2009. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine.* " O'Reilly Media, Inc.".

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

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. 2018. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. 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 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

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.

Martha Lewis. 2019. Compositional hyponymy with positive operators. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 638–647, Varna, Bulgaria. INCOMA Ltd.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. *Advances in Neural Information Processing Systems*, 33.

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

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2018. Mixed precision training. In *International Conference on Learning Representations*.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. How additional knowledge can improve natural language commonsense question answering? *arXiv preprint arXiv:1909.08855*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Steven Rennie, Etienne Marcheret, Neil Mallinar, David Nahamoo, and Vaibhava Goel. 2020. Unsupervised adaptation of question answering systems via generative self-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1148–1157, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. 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 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Assaf Shocher, Nadav Cohen, and Michal Irani. 2018. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126.

Peter Snyder. 1990. tmpfs: A virtual memory file system. In *Proceedings of the autumn 1990 EUUG Conference*, pages 241–248.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*.

Elizabeth Closs Traugott and Richard B Dasher. 2001. *Regularity in semantic change*, volume 97. Cambridge University Press.

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*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

W Wang, N Yang, F Wei, B Chang, and M Zhou. 2017. R-net: Machine reading comprehension with self-matching networks. *Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep*, 5.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Yang Yu, Wei Zhang, Kazi Hasan, Mo Yu, Bing Xiang, and Bowen Zhou. 2016. End-to-end answer chunk extraction and ranking for reading comprehension. *arXiv preprint arXiv:1610.09996*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.