Leveraging QA Datasets to Improve Generative Data Augmentation

Dheeraj Mekala[◇] **Tu Vu**[♠] **Timo Schick**[♣] **Jingbo Shang**[◇,♡,*]

[◇] University of California San Diego

[♠] University of Massachusetts Amherst

[♣] Meta AI Research

[♡] Halıcıoğlu Data Science Institute, University of California San Diego

[◇] {dmekala, jshang}@ucsd.edu [♠] tuvu@cs.umass.edu [♣] schick@fb.com

Abstract

The ability of generative language models (GLMs) to generate text has improved considerably in the last few years, enabling their use for *generative data augmentation*. In this work, we propose CONDA, an approach to further improve GLMs' ability to generate synthetic data by reformulating data generation as context generation for a given question-answer (QA) pair and leveraging QA datasets for training context generators. Then, we cast downstream tasks into the same question answering format and adapt the fine-tuned context generators to the target task domain. Finally, we use the fine-tuned GLM to generate relevant contexts, which are in turn used as synthetic training data for their corresponding tasks. We perform extensive experiments on multiple classification datasets and demonstrate substantial improvements in performance for both few- and zero-shot settings. Our analysis reveals that QA datasets that require high-level reasoning abilities (e.g., abstractive and common-sense QA datasets) tend to give the best boost in performance in both few-shot and zero-shot settings.

1 Introduction

Recent advances in NLP have substantially improved the capability of pretrained language models to generate high-quality text (Radford and Narasimhan, 2018; Radford et al., 2019; Lewis et al., 2020; Brown et al., 2020). Various approaches (e.g., Kumar et al., 2020; Anaby-Tavor et al., 2020; Mekala et al., 2021) leverage this capability for *generative data augmentation*. This process usually involves first fine-tuning the GLM on training samples prepended with their target label and then generating synthetic data by prompting the GLM with a given target label. However, it is not evident that the model parameters learnt during pretraining or fine-tuning should support data generation using such unintuitive formulations with label tokens as prompts: In low data regimes, fine-tuning can be unstable (Devlin et al., 2019) and relies on the pretrained parameters to be reasonably well-suited for the target task (Phang et al., 2018). Therefore, for target domains that are different from the pretraining domain, such formulations may result in poor quality generation (Feng et al., 2020).

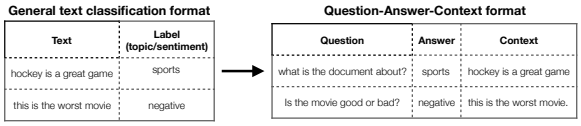


Figure 1: Examples of converting topic classification and sentiment analysis data into question-answer-context format.

To address this challenge, we propose **CONDA**, an approach to leverage existing QA datasets for training ***Con****text generators* to improve generative **D**ata **A**ugmentation. We propose to use a question answering (QA) formulation as a consistent format to prompt GLMs for synthetic data: We use QA datasets for training GLMs to be *context generators* for a given question and answer.

As illustrated in Figure 2, our method consists of two steps. The first step is QAC fine-tuning, where we fine-tune a pretrained language model on a QA dataset to obtain a general context generator that is capable of generating contexts for given questions and answers. To this end, we view the QA dataset in question-answer-context format instead of the context-question-answer format used to solve QA tasks (Radford and Narasimhan, 2018; Radford et al., 2019; Raffel et al., 2020). Then, we adapt the general context generator to the target domain by further training it on available few-shot data, resulting in a target-domain context generator. Inspired from recent work in converting several NLP tasks into a common format (McCann et al., 2018; Raffel et al., 2020), we format the target tasks into a

* Jingbo Shang is the corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9737 - 9750
December 7-11, 2022

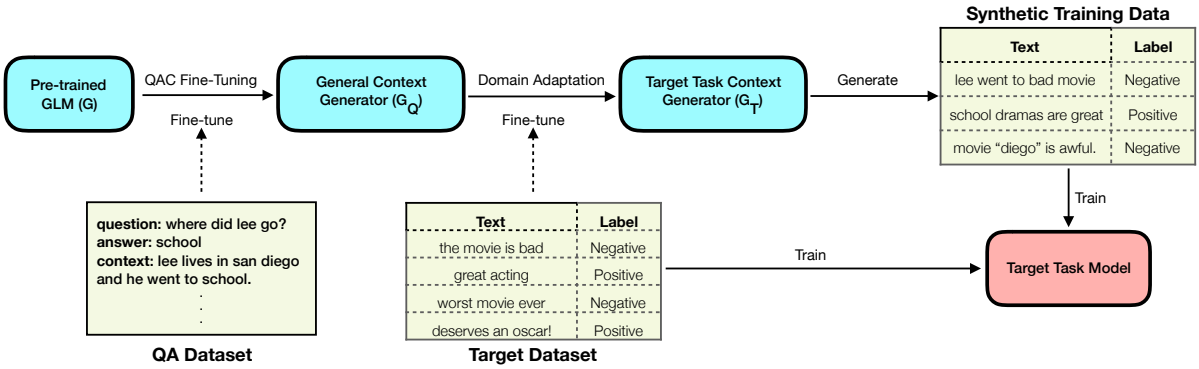


Figure 2: We propose to use QA datasets for transforming pre-trained generative language models into high-quality target task data generators. We view QA datasets in question-answer-context format and fine-tune a pre-trained GLM (G) to obtain a general context generator (G_Q). Then, we adapt it to the target domain by training it further on few-shot target dataset supervision, resulting in G_T. Finally, using G_T, we generate synthetic training data for the target task, use it to augment the few-shot target dataset and train the target task model on the augmented data.

question-answer schema. For example, as shown in Figure 1, topic classification and sentiment analysis data can be cast into the question-answer-context format with its respective *label* as *answer*, and *text* as *context*. We adapt the context generator to the target task domain by further training on target task few-shot supervision, resulting in target task context generator. Finally, we generate synthetic training data for the target task by generating contexts for questions and answers pertaining to the respective dataset. Then, we add the generated samples to the few-shot supervision and train a target task model on the augmented data.

We perform extensive experiments on multiple sentiment analysis and topic classification datasets with several abstractive, extractive, and common-sense reasoning QA datasets. Through rigorous experiments and thorough analysis, we observe that QA datasets that require high-level reasoning abilities such as abstractive and common-sense QA datasets suit the best for generating high-quality data.

Our contributions are summarized as follows:

- We propose to use QA datasets for training generative language models to be context generators for a given question and answer.
- We formulate various classification tasks into a QA format and model synthetic training data generation for these tasks as context generation.
- We perform experiments on multiple sentiment analysis and topic classification datasets to demonstrate the effectiveness of our method in zero- and few-shot settings.
- We release the code on Github[1].

2 Related Work

Data Augmentation Wei and Zou (2019) propose a simple data augmentation method using synonym replacement, random insertion, random swap, and random deletion. Sennrich et al. (2016) augment samples by translating them into foreign language and then back to English. Du et al. (2021) compute task-specific query embeddings to retrieve sentences from unlabeled documents from the Internet. After a rise in pretrained generative language models, the generation capabilities of these models have been explored to generate synthetic data. Anaby-Tavor et al. (2020); Kumar et al. (2020); Schick and Schütze (2021b); Mekala et al. (2021) generate labeled documents using the GLMs and (Yang et al., 2020) do so specifically for common-sense reasoning. Puri et al. (2020) use GLMs to synthesize questions and answers and improve performance on question answering. Vu et al. (2021) generate data for NLI tasks.

Few-shot Learning Our work is closely related to few-shot learning as we take a few annotated samples as supervision. The idea of formulating classification as a prompting task is getting increasingly popular. Brown et al. (2020) introduce a new paradigm called in-context learning to infer from large language models using few annotated samples. Schick and Schütze (2021a) formulate input samples as cloze-style phrases and assign pseudo-labels that are used for training the classifier and Tam et al. (2021) improves their approach

[1] https://github.com/dheeraj7596/CONDA

further without using any task-specific unlabeled data. (McCann et al., 2018; Raffel et al., 2020) format several NLP tasks into a question-answer and text-to-text schema. Lin et al. (2021) train multilingual autoregressive language models to enable few-shot learning in multiple languages. Gao et al. (2021) propose to generate prompts and convert smaller pretrained language models to few-shot learners. Other work proposes to pre-train prompts by adding soft prompts into the pre-training stage (Gu et al., 2022; Vu et al., 2022b,a).

Language Model Fine-Tuning Pre-trained language models are applied to downstream tasks by fine-tuning them using task-specific objectives (Howard and Ruder, 2018). However, this process requires significant annotated downstream task data (Yogatama et al., 2019). Many methods have been proposed to address this challenge. Gururangan et al. (2020) propose to continue training on unlabeled data from the target task domain. Aghajanyan et al. (2021) propose pre-finetuning, a large-scale multi-task learning stage between language model pre-training and fine-tuning. Phang et al. (2018) introduce intermediate task fine-tuning which involves fine-tuning a language model on an auxiliary task before continuously training on the target task. Pruksachatkun et al. (2020) observe that the tasks requiring high-level inference and reasoning abilities are the best choice as intermediate tasks. Vu et al. (2020) identify the best auxiliary tasks for high performance on downstream tasks. Vu et al. (2021) use NLI as auxiliary task to generate synthetic NLI data for intermediate fine-tuning. Our method differs from (Phang et al., 2018) in two fronts: (1) we use QA datasets for training context generators instead of answering the question , and (2) we use the fine-tuned GLM to generate synthetic data instead of training directly for the downstream tasks. It also differs from (Vu et al., 2021) in terms of the generated data, where they consider NLI as an auxiliary task and generate synthetic samples in target-domain for the NLI task irrespective of the target task and perform intermediate task fine-tuning. CONDA formats target tasks into question-answer format and directly generates samples relevant for target task.

3 CONDA: QA Datasets for Generative Data Augmentation

In this section, we describe the problem statement, and explain our method including QAC fine-tuning, target-domain adaptation, and synthetic training data generation.

3.1 Problem Formulation

For a given task $\mathcal{T}$, the input in a few-shot setting contains a very small labeled dataset $\mathcal{L}_\mathcal{T} = \{(\mathcal{D}_1, l_1), (\mathcal{D}_2, l_2), \ldots, (\mathcal{D}_{|\mathcal{L}_\mathcal{T}|}, l_{|\mathcal{L}_\mathcal{T}|})\}$ and m target classes $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$. Our method requires users to provide a question per dataset that is representative of the task to be solved. Our aim is to build a model for the task $\mathcal{T}$ that assigns a label $\mathcal{C}_j \in \mathcal{C}$ to each document $\mathcal{D}$.

3.2 QAC Fine-tuning

We consider question-answering datasets Q containing triplets (q, a, c) of a question q, the corresponding answer a, and a context c required to derive the correct answer. Question-answering datasets can roughly be divided into *extractive* (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Reddy et al., 2019) and *abstractive* datasets (Kočiský et al., 2018; Huang et al., 2019; Xiong et al., 2019; Sap et al., 2019). For extractive QA datasets, the answer can be found as a contiguous span in the context, whereas in abstractive QA datasets, the answer needs to be generated in natural language without being able to rely on the vocabulary of the question or context.

We transform the QA dataset Q into training data D_{QAC} for fine-tuning GLM. To this end, each triplet (q, a, c) is converted into a single text by prepending *"question:"*, *"answer:"* and *"context:"*, respectively, and concatenating q, a and c separated by newlines. For example, a preprocessed training document in D_{QAC} from an extractive QA dataset might look as follows:

> question: when did battle of plassey happen?
> answer: 23 june 1757
> context: the battle of plassey was a decisive victory of the british east india company over the nawab of bengal and his french allies on 23 june 1757.

We fine-tune a pretrained GLM G on D_{QAC} to obtain a *general context generator* G_Q using a language modeling objective to maximize the log-likelihood of the (q, a, c) triplet. The general context generator G_Q is capable of generating contexts for given questions and answers.

3.3 Domain Adaptation and Synthetic Training Data Generation

We adopt G_Q to the target domain by fine-tuning it further on available few-shot data. To preserve its context generating ability, we perform QAC fine-tuning instead of regular language model fine-tuning. This is enabled by transforming the few-shot supervision into our question-answer-context format. First, we manually design one question per dataset that is representative of the task and the dataset. Furthermore, following Schick and Schütze (2021a), we define a verbalizer as a mapping v: $\mathcal{C} \rightarrow \mathcal{V}$ that maps each label in $\mathcal{C}$ to a word from G_Q's vocabulary $\mathcal{V}$. Finally, for every document $\mathcal{D}_i$ and its respective label l_i in our few-shot data, we consider the verbalizer mapping of the label, $v(l_i)$, as answer and the text $\mathcal{D}_i$ as context. For example, a *negative* review *"I hate this movie"* from the IMDb dataset (Maas et al., 2011) is transformed as follows:

```
question: is the movie good or bad?
answer: bad
context: i hate this movie.
```

We fine-tune G_Q on the converted few-shot data to obtain a target task context generator $G_{\mathcal{T}}$.

Synthetic Training Data Generation Recall that our method requires a question q for every dataset that is representative of the task to be solved. To obtain synthetic training data, for every distinct label $\mathcal{C}_j$, we create a question-answer prompt with q as question, $v(\mathcal{C}_j)$ as answer and let $G_{\mathcal{T}}$ generate the context c_{gen}. The generated context c_{gen} and label $\mathcal{C}_j$ are considered as a synthetic training sample. We repeat this process multiple times to generate n samples that we collect in a synthetic training dataset denoted by $\mathcal{D}_{gen}$.

As a final step, we train the target task model on the combination of $\mathcal{D}_{gen}$ and our original few-shot dataset $\mathcal{L}_{\mathcal{T}}$. We use this trained target-task model for inference.

4 Experiments

In this section, we evaluate our method against several data augmentation and few-shot methods on sentiment analysis and text classification tasks.

4.1 QA Datasets

We consider several extractive, abstractive, and common-sense QA datasets. Common-sense QA datasets are also abstractive datasets that require common-sense reasoning to answer the questions. The QA dataset statistics are provided in Table 1. The details of these datasets are as follows:

Dataset	Type	# Samples	Training Context
SQuAD	Extractive	87,600	Wikipedia
NewsQA	Extractive	76,560	News
TweetQA	Abstractive	10,692	News Tweets
SocialIQA	Commonsense	33,410	Crowdsourcing
CosmosQA	Commonsense	21,448	Blogs

Table 1: Relevant statistics of the QA dataset used in our experiments.

Dataset	Question	Verbalized Labels
	Sentiment	
IMDb	*is this movie good or bad?*	good, bad
Yelp	*how is the service?*	awful, bad, fine, good, excellent
SST-2	*is this sentence positive or negative?*	positive, negative
	Topic	
Yahoo	*what is this document about?*	sports, society, science, health, politics, education, computer, business, entertainment, relationship
NYT	*what is this document about?*	arts, business, politics, sports
AGNews	*what is this document about?*	sports, business, technology, politics

Table 2: Questions and Verbalized labels of the target task datasets considered in our experiments.

- **SQuAD** (Rajpurkar et al., 2016, 2018) is a collection of questions and answers based on Wikipedia articles.
- **NewsQA** (Trischler et al., 2017) is a challenging QA dataset in the News domain where crowd-workers were shown a news article's headline and summary, and asked to formulate a question about the article without accessing its content.
- **TweetQA** (Xiong et al., 2019) is a QA dataset made from a collection of tweets sampled from two major news websites (CNN and NBC).
- **SocialIQA** (Sap et al., 2019) is a QA dataset that tests social common-sense intelligence. The data is made of common phrases from stories and books.
- **CosmosQA** (Huang et al., 2019) is a commonsense-based reading comprehension task formulated as multiple-choice questions. Answering questions requires reasoning not only based on the exact text spans in the context, but also abstractive commonsense reasoning.

4.2 Target Task Datasets

We evaluate our method on six English text classification datasets. In particular, we consider the three sentiment analysis datasets: IMDb reviews (Maas

et al., 2011), Yelp[2], and SST-2 (Socher et al., 2013), as well as three topic classification datasets: Yahoo (Zhang et al., 2015), The New York Times[3] (NYT), and AGNews (Zhang et al., 2015). The dataset-representative questions, and their respective verbalized labels of target task datasets are mentioned in Table 2. We follow and adapt McCann et al. (2018) for questions in sentiment analysis datasets. The question for topic classification is intuitive and straightforward. More details about the datasets can be found in Appendix A.1.

4.3 Compared Methods

We compare with a wide range of data augmentation and intermediate-task fine-tuning (ITFT) methods described below:

- **BERT-FT** trains the `BERT-base-uncased` classifier (Devlin et al., 2019) on the few-shot supervision.
- **ITFT-X** (Phang et al., 2018) first trains a model on dataset X and fine-tunes it further on the target task. We compare with ITFT-MNLI and ITFT-SQuAD fine-tuned intermediately on MNLI (Williams et al., 2018) and SQuAD datasets respectively.
- **BackTranslation** (Sennrich et al., 2016) augments samples by translating them into a non-English language and translating them back to English. We translate them to French, Spanish, and Portuguese thereby augmenting three synthetic samples for every sample.
- **PEGASUS** (Zhang et al., 2019) is a state-of-the-art paraphrasing model. We paraphrase the input text and consider it as a synthetic sample and augment it to the training set.
- **EDA** (Wei and Zou, 2019) generates synthetic samples by synonym replacement, random insertion, random swap, and random deletion and augment them to the training set.
- **LAMBADA** (Anaby-Tavor et al., 2020) fine-tunes a GLM on few-shot supervision prepended with their target labels and then generates synthetic data by prompting the GLM with a given target label.

We denote our method as CONDA, which includes QAC fine-tuning, domain adaptation, synthetic samples generation, and training the target task classifier. CONDA-X represents that the QAC fine-tuning of GLM is performed on QA dataset X. We also compare with CONDA$\backslash QA$ where we perform no QAC fine-tuning and directly fine-tune the GLM on target dataset.

4.4 Experiment Settings

We consider two low-data regimes: few-shot and zero-shot. We consider 8 annotated samples per label in the few-shot setting. In the zero-shot setting, we skip the domain adaptation step and use G_Q directly for synthetic training data generation and train the target task model only on the generated synthetic training data. We use GPT2-Medium (Radford et al., 2019) as our GLM and fine-tune it for 3 epochs in QAC-fine-tuning and domain adaptation steps. While generating synthetic training samples, we use top-k sampling with $k = 20$, a maximum length of 200 tokens, and generate $n = 450$ synthetic samples per label. We use `BERT-base-uncased` (Devlin et al., 2019) as target task classifier. We feed `[CLS]` representation into the classification head and train all the parameters on the downstream target tasks. Following (Devlin et al., 2019), we fix the number of epochs of target task BERT classifier training to 4 unless mentioned otherwise. We perform 3 random restarts and report the mean and standard deviation.[4] We use the Transformers library (Wolf et al., 2020) and NVIDIA RTX A6000 GPUs for our experiments.

To enable a fair comparison, we generate the same number of samples per label as CONDA (i.e., 450) for all data augmentation baselines. We use `BERT-base-uncased` as the target task classifier for all baselines. CONDA$\backslash QA$ for zero-shot setting implies a pre-trained GPT2. While training the target task classifier, since the number of training samples for baselines like BERT-FT, ITFT are different than data augmentation baselines and our method CONDA, we set the number of epochs for all baselines such that the number of update steps remain the same for a fair comparison.

4.5 Results and Discussion

Results for few- and zero-shot settings are shown in Table 3 and Table 4, respectively, using Micro- and Macro-F1 as evaluation metrics. We discuss the effectiveness of our method below.

CONDA vs Baselines. In the few-shot setting, CONDA with abstractive and common-sense based datasets outperforms all baselines for most of the datasets, beating the best baseline in five out of

[2] `https://www.yelp.com/dataset/`

[3] `http://developer.nytimes.com/`

[4] For each restart, we resample the few-shot training set.

	Sentiment						Topic					
	IMDb		Yelp		SST-2		NYT		Yahoo		AGNews	
Method	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1
BERT-FT	$69.1_{4.9}$	$69.1_{4.9}$	$39.8_{2.3}$	$38.9_{3.4}$	$62.0_{4.7}$	$61.8_{4.8}$	$94.4_{1.1}$	$88.1_{1.6}$	$55.4_{2.1}$	$55.2_{1.6}$	$78.4_{1.8}$	$78.3_{1.8}$
ITFT-MNLI	$\underline{73.9_{4.6}}$	$\underline{73.5_{4.8}}$	$40.4_{2.6}$	$40.0_{2.9}$	$\underline{66.5_{6.9}}$	$\underline{65.5_{6.9}}$	$90.1_{1.2}$	$80.8_{1.1}$	$38.7_{5.8}$	$37.6_{5.2}$	$71.1_{1.1}$	$70.6_{1.1}$
ITFT-SQuAD	$65.5_{4.4}$	$64.6_{4.3}$	$38.4_{2.5}$	$36.8_{2.5}$	$61.9_{2.2}$	$61.5_{2.2}$	$93.0_{0.5}$	$85.3_{1.3}$	$45.3_{3.4}$	$45.0_{3.0}$	$72.0_{1.9}$	$71.4_{2.1}$
BackTranslation	$68.0_{4.6}$	$67.1_{5.2}$	$\underline{41.6_{2.4}}$	$\underline{40.3_{3.0}}$	$60.6_{4.5}$	$60.0_{4.9}$	$95.4_{0.6}$	$90.0_{1.3}$	$57.4_{1.4}$	$57.1_{1.2}$	$80.0_{2.2}$	$79.8_{2.3}$
PEGASUS	$66.8_{5.0}$	$65.9_{5.6}$	$35.9_{3.7}$	$34.4_{3.1}$	$61.1_{5.3}$	$60.9_{5.3}$	$93.2_{0.5}$	$87.2_{0.6}$	$58.1_{1.9}$	$57.2_{1.7}$	$\underline{81.1_{1.9}}$	$\underline{80.9_{2.1}}$
EDA	$63.6_{1.4}$	$62.1_{1.6}$	$39.1_{1.9}$	$37.9_{2.0}$	$57.4_{4.0}$	$52.9_{7.4}$	$\underline{95.8_{0.8}}$	$\underline{90.9_{1.7}}$	$56.1_{1.7}$	$55.8_{1.8}$	$80.0_{3.0}$	$79.8_{3.0}$
LAMBADA	$50.3_{0.7}$	$42.3_{5.4}$	$20.8_{1.06}$	$11.1_{6.3}$	$49.6_{1.3}$	$45.8_{3.3}$	$60.3_{19.7}$	$45.9_{17.4}$	$25.7_{4.7}$	$22.6_{3.2}$	$49.3_{9.9}$	$46.9_{10.3}$
CONDA\QA	$72.2_{6.9}$	$71.3_{8.0}$	$36.8_{0.6}$	$23.9_{1.7}$	$50.6_{0.5}$	$35.1_{0.5}$	$93.5_{0.8}$	$85.7_{1.2}$	$\underline{58.5_{0.3}}$	$\underline{57.3_{0.4}}$	$79.4_{1.6}$	$78.8_{1.8}$
CONDA-SQuAD	$53.9_{1.9}$	$45.9_{6.2}$	$37.9_{0.7}$	$31.1_{3.2}$	$51.5_{1.6}$	$39.8_{7.6}$	$93.2_{0.8}$	$86.0_{1.7}$	$56.9_{0.5}$	$55.4_{0.5}$	$\mathbf{81.6_{0.8}}$	$\mathbf{81.3_{0.9}}$
CONDA-NewsQA	$57.9_{3.7}$	$55.5_{5.4}$	$36.4_{1.1}$	$31.6_{2.0}$	$56.0_{6.2}$	$50.5_{10.3}$	$91.5_{0.3}$	$81.1_{0.6}$	$58.3_{0.7}$	$57.2_{0.8}$	$80.0_{3.3}$	$79.6_{3.6}$
CONDA-TweetQA	$\mathbf{75.1_{2.3}}$	$\mathbf{74.5_{2.5}}$	$\mathbf{42.9_{1.1}}$	$\mathbf{42.0_{1.8}}$	$\mathbf{67.7_{4.8}}$	$\mathbf{67.5_{4.9}}$	$94.1_{0.6}$	$86.6_{1.3}$	$\mathbf{59.4_{0.4}}$	$\mathbf{58.1_{0.3}}$	$\mathbf{83.0_{0.9}}$	$\mathbf{82.9_{0.9}}$
CONDA-SocialIQA	$\mathbf{79.5_{1.9}}$	$\mathbf{79.5_{1.9}}$	$39.4_{1.5}$	$32.2_{2.8}$	$\mathbf{75.4_{1.4}}$	$\mathbf{75.2_{1.6}}$	$93.2_{3.4}$	$85.8_{1.3}$	$\mathbf{61.9_{0.5}}$	$\mathbf{61.1_{0.6}}$	$\mathbf{81.9_{0.2}}$	$\mathbf{81.7_{0.2}}$
CONDA-CosmosQA	$\mathbf{77.0_{3.2}}$	$\mathbf{76.4_{3.7}}$	$\mathbf{42.3_{0.1}}$	$37.5_{1.0}$	$\mathbf{67.4_{0.6}}$	$\mathbf{66.9_{1.2}}$	$94.3_{0.4}$	$87.7_{1.1}$	$\mathbf{63.8_{0.6}}$	$\mathbf{63.3_{0.4}}$	$\mathbf{82.8_{0.8}}$	$\mathbf{82.5_{0.8}}$

Table 3: Few-Shot Evaluation Results. Micro- and Macro-F1 are used as evaluation metrics. All experiments are repeated with three random seeds. Mean and standard deviation (in the subscript) are reported. The best baseline for each dataset is underlined and all results of CONDA that outperform the best baseline are in bold.

	Sentiment						Topic					
	IMDb		Yelp		SST-2		NYT		Yahoo		AGNews	
Method	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1
CONDA\QA	$70.9_{3.7}$	$70.0_{3.8}$	$32.4_{3.7}$	$19.8_{2.0}$	$52.9_{3.6}$	$41.8_{9.9}$	$90.7_{0.9}$	$80.0_{1.8}$	$57.9_{0.3}$	$57.1_{0.7}$	$77.0_{1.5}$	$76.2_{1.6}$
CONDA-SQuAD	$53.3_{2.4}$	$42.7_{7.1}$	$30.2_{4.5}$	$21.4_{4.6}$	$52.4_{2.6}$	$47.3_{5.9}$	$85.7_{3.8}$	$74.3_{3.1}$	$56.5_{1.5}$	$54.9_{1.7}$	$79.3_{0.1}$	$78.9_{0.2}$
CONDA-NewsQA	$53.4_{2.6}$	$47.4_{10.0}$	$32.8_{1.0}$	$23.1_{3.6}$	$51.6_{1.7}$	$46.2_{3.6}$	$89.8_{0.2}$	$77.1_{1.1}$	$55.9_{1.1}$	$54.6_{0.8}$	$76.8_{3.0}$	$75.7_{3.5}$
CONDA-TweetQA	$72.4_{5.0}$	$70.6_{6.1}$	$\mathbf{38.0_{2.4}}$	$\mathbf{37.6_{2.3}}$	$61.2_{3.9}$	$56.5_{6.4}$	$90.3_{1.9}$	$78.5_{4.7}$	$54.0_{0.4}$	$52.4_{0.2}$	$76.4_{1.7}$	$76.2_{2.0}$
CONDA-SocialIQA	$\mathbf{78.3_{4.3}}$	$\mathbf{77.6_{5.1}}$	$36.9_{2.2}$	$31.7_{1.1}$	$\mathbf{76.2_{1.2}}$	$\mathbf{76.1_{1.2}}$	$87.0_{3.0}$	$77.6_{2.9}$	$56.1_{1.6}$	$55.2_{2.1}$	$79.6_{1.9}$	$79.4_{2.2}$
CONDA-CosmosQA	$75.9_{2.3}$	$75.5_{2.8}$	$37.4_{1.5}$	$35.1_{1.7}$	$66.2_{6.4}$	$66.0_{6.5}$	$\mathbf{92.8_{0.3}}$	$\mathbf{84.5_{1.3}}$	$\mathbf{63.4_{0.5}}$	$\mathbf{62.9_{0.3}}$	$\mathbf{81.8_{1.6}}$	$\mathbf{81.5_{1.4}}$

Table 4: Zero-Shot Evaluation Results. Mean and standard deviation (in the subscript) are reported.

six cases. CONDA performs better than BERT-FT on all datasets, achieving up to 14% improvement on SST-2. Although ITFT performs better than vanilla fine-tuning, CONDA demonstrates better performance than ITFT on all datasets. For example, CONDA-TweetQA shows 11% improvement over ITFT-SQuAD on AG-News. CONDA demonstrates higher performance than data-augmentation baselines on all datasets except NYT. The comparison between CONDA and LAMBADA shows that our QA formulation prompt is more intuitive and informative than just the target label. We attribute the superior performance of CONDA to the context-generating ability acquired during QAC fine-tuning that is efficiently leveraged by generating synthetic samples, which are added to the training set.

Abstractive vs Extractive QA Datasets. We observe that the performance of CONDA with abstractive QA datasets is significantly better than CONDA with extractive QA datasets in both few-shot and zero-shot settings. For example, CONDA-TweetQA has an improvement of more than 20% over CONDA-SQuAD on IMDb in few-shot setting. We surmise that this is because of the intrinsic nature of extractive QA datasets (i.e., the answer always being present in the context as a contiguous span). We observe that GLMs fine-tuned on an extractive QA dataset retain the ability to generate contexts that encompass the answer. Note that, while generating synthetic training samples, the answer in the prompt is its respective topic. For example, out of 500 generated samples by CONDA-SQuAD for Yelp dataset, 213 samples contain at least one occurrence of its corresponding verbalized label whereas it is only 73 for CONDA-CosmosQA. Thus, many synthetic samples generated contain their corresponding label in text. Therefore, a classifier trained on synthetic samples that have their corresponding labels in the text, easily overfits on the label tokens and does not generalize well to unseen test data.

Comparison with CONDA\QA. CONDA with abstractive QA datasets perform better than CONDA\QA in both few-shot and zero-shot settings, attaining improvements up to 40% and 35% respectively in macro-F1 on SST-2. This demonstrates that the context generating abilities are learnt and reinforced during the QAC fine-tuning on QA datasets which is efficiently utilized by generating synthetic samples.

QA Dataset	Setting	Sentiment			Topic		
		IMDb	Yelp	SST-2	NYT	Yahoo	AGNews
SQuAD	CONDA	$45.9_{6.2}$	$31.1_{3.2}$	$39.8_{7.6}$	$86.0_{1.7}$	$55.4_{0.5}$	$81.3_{0.9}$
	- DA	$51.3_{12.7}$	$28.2_{0.5}$	$33.4_{0.1}$	$87.1_{1.0}$	$55.0_{1.7}$	$82.5_{0.6}$
	- Few Shot	$49.4_{9.5}$	$25.9_{3.4}$	$43.7_{4.0}$	$75.0_{4.0}$	$47.4_{0.4}$	$77.9_{3.0}$
NewsQA	CONDA	$55.5_{5.4}$	$31.6_{2.0}$	$50.5_{10.3}$	$81.1_{0.6}$	$57.2_{0.8}$	$79.6_{3.6}$
	- DA	$60.9_{9.6}$	$32.0_{3.6}$	$46.2_{8.5}$	$79.8_{0.4}$	$56.4_{1.1}$	$79.2_{3.5}$
	- Few Shot	$50.9_{4.8}$	$23.4_{1.9}$	$46.0_{7.0}$	$77.2_{2.1}$	$54.3_{0.7}$	$76.0_{4.1}$
TweetQA	CONDA	$74.5_{2.5}$	$42.0_{1.8}$	$67.5_{4.9}$	$86.6_{1.3}$	$58.1_{0.3}$	$82.9_{0.9}$
	- DA	$80.5_{3.5}$	$42.1_{0.2}$	$63.2_{7.5}$	$85.3_{2.1}$	$57.1_{1.1}$	$81.1_{1.6}$
	- Few Shot	$74.0_{2.7}$	$40.6_{0.7}$	$59.3_{12.4}$	$77.3_{4.8}$	$53.8_{0.3}$	$77.4_{1.5}$
SocialIQA	CONDA	$79.5_{1.9}$	$32.2_{2.8}$	$75.2_{1.6}$	$85.8_{1.3}$	$61.1_{0.6}$	$81.7_{0.2}$
	- DA	$81.0_{1.9}$	$35.3_{0.8}$	$76.1_{0.6}$	$87.6_{1.3}$	$60.7_{0.8}$	$82.4_{1.2}$
	- Few Shot	$77.7_{4.0}$	$31.4_{3.0}$	$75.2_{1.0}$	$76.5_{1.7}$	$55.8_{1.2}$	$78.2_{1.4}$
CosmosQA	CONDA	$76.4_{3.7}$	$37.5_{1.0}$	$66.9_{1.2}$	$87.7_{1.1}$	$63.3_{0.4}$	$82.5_{0.8}$
	- DA	$76.3_{2.4}$	$36.8_{2.3}$	$51.8_{9.3}$	$87.1_{1.0}$	$62.9_{0.3}$	$82.6_{0.6}$
	- Few Shot	$74.9_{1.0}$	$35.8_{1.4}$	$64.4_{9.1}$	$82.9_{0.7}$	$60.1_{0.2}$	$80.3_{1.4}$

Table 5: Ablation Study. Macro-F1 is used as evaluation metric.

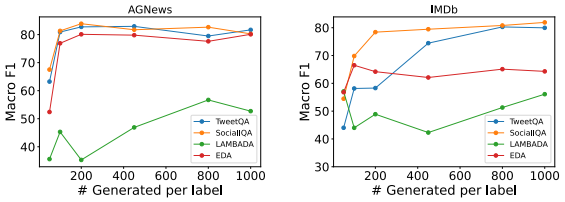
Zero-shot Performance. The zero-shot performance of CONDA follows a similar trend as few-shot performance: abstractive and common-sense reasoning QA datasets lead to better performance than extractive datasets and no QAC fine-tuning.

4.6 Ablation Study

To understand the impact of domain adaptation and few-shot samples, we compare CONDA with two ablated versions in Table 5: (1) CONDA-*DA* represents our method without domain adaptation (i.e., generating synthetic data using G_Q and training the classifier on combined few-shot supervision and synthetic data generated by G_Q), (2) CONDA-*Few Shot* represents the classifier trained only on the samples generated by G_T. We also present the results of our complete pipeline for reference. CONDA performs better than CONDA-Few shot in most cases, demonstrating the importance of including few-shot samples in the training set for the classifier. The comparison between CONDA and CONDA-DA suggests that fine-tuning the language model further on the target dataset helps in some scenarios but does not always improve performance. This is in line with previous research findings (Du et al., 2021; Vu et al., 2021; Pryzant et al., 2022). We conjecture that domain adaptation is important when the structure of the target task dataset is very different from the QA dataset. For example, domain adaptation helps most of the QA datasets on SST-2 dataset because the text in SST-2 is a single sentence, whereas most of the QA datasets have paragraphs as context. Moreover, it also depends on the number of samples the language model is fine-tuned on during domain adaptation. We observe that the higher the number of samples, the more positive their impact. For example, the number of few-shot samples is the highest in Yahoo compared to other datasets and domain adaptation positively contributes to the performance on Yahoo for all QA datasets.

Method	Sentiment			Topic		
	IMDb	Yelp	SST-2	NYT	Yahoo	AGNews
CONDA\QA-L	$79.7_{2.1}$	$43.2_{4.4}$	$67.6_{8.2}$	$84.8_{1.3}$	$60.3_{0.8}$	$77.7_{2.2}$
CONDA-L-SQuAD	$70.0_{15.2}$	$38.9_{1.5}$	$64.3_{7.2}$	$84.3_{2.9}$	$60.3_{1.3}$	$\mathbf{81.1_{1.7}}$
CONDA-L-NewsQA	$72.4_{9.4}$	$38.3_{0.3}$	$58.1_{6.2}$	$\mathbf{85.5_{2.4}}$	$\mathbf{61.4_{1.1}}$	$\mathbf{82.2_{1.2}}$
CONDA-L-TweetQA	$76.4_{5.7}$	$\mathbf{45.0_{1.3}}$	$\mathbf{74.6_{2.3}}$	$84.4_{0.1}$	$\mathbf{61.6_{0.1}}$	$\mathbf{79.7_{3.1}}$
CONDA-L-SocialIQA	$\mathbf{81.6_{2.4}}$	$\mathbf{43.9_{3.4}}$	$\mathbf{77.5_{1.3}}$	$\mathbf{89.0_{0.4}}$	$\mathbf{62.0_{0.5}}$	$\mathbf{80.9_{2.2}}$
CONDA-L-CosmosQA	$\mathbf{83.4_{0.6}}$	$\mathbf{43.2_{2.2}}$	$\mathbf{77.2_{1.7}}$	$\mathbf{86.5_{2.4}}$	$\mathbf{61.0_{0.6}}$	$\mathbf{79.5_{3.9}}$

Table 6: Few-Shot Evaluation Results with GPT2-Large as GLM (*-L* denotes GPT2-Large). Macro-F1 is used as evaluation metric. All results of CONDA-L that perform better than CONDA\QA-L are in bold.



(a) AGNews (b) IMDb

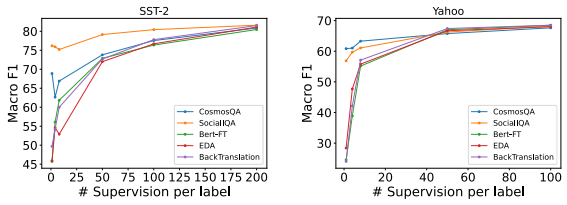
Figure 3: Macro-F_1 scores of CONDA-TweetQA and CONDA-SocialIQA w.r.t. number of generated samples per class. We fix the few-shot supervision size to 8 samples per label. Each experiment is repeated with three different seeds and the mean performance is plotted.

4.7 Larger Generative Language Models

Experimental results with GPT2-Large as the GLM are shown in Table 6. We observe that the relative performance trend remains the same as GPT2-Medium i.e. CONDA with abstractive datasets performs better than CONDA with extractive datasets and CONDA\QA-L. This indicates that QAC fine-tuning improves the performance of generative data augmentation with larger GLMs as well.

4.8 Performance vs No. of Generated Samples

We fix the few-shot supervision size to 8 samples per label and vary the number of generated samples per label and plot the performance of CONDA-TweetQA, CONDA-SocialIQA, and baselines such as LAMBADA and EDA on AGNews and IMDb datasets, shown in Figure 3. We repeat each setting with three different seeds and plot the mean performance. We observe that the performance increases and it plateaus after a while. This shows that synthetic training data can give a substantial boost to the few-shot training data, minimizing the human effort in manual annotations; however, it cannot replace the original training data completely as it



(a) SST-2 (b) Yahoo

Figure 4: Macro-F_1 scores of CONDA-CosmosQA and CONDA-SocialIQA w.r.t. number of few-shot annotated samples per class. Each experiment is repeated with three different seeds and mean performance is plotted.

Method	IMDb	SST-2	Yahoo	AGNews
LMPT	$64.8_{3.6}$	$57.5_{2.1}$	$49.9_{1.2}$	$79.2_{1.7}$
CONDA-TweetQA	$\mathbf{74.5_{2.5}}$	$\mathbf{67.5_{4.9}}$	$\mathbf{58.1_{0.3}}$	$\mathbf{82.9_{0.9}}$
CONDA-SocialIQA	$\mathbf{79.5_{1.9}}$	$\mathbf{75.2_{1.6}}$	$\mathbf{61.1_{0.6}}$	$\mathbf{81.7_{0.2}}$
CONDA-CosmosQA	$\mathbf{76.4_{3.7}}$	$\mathbf{66.9_{1.2}}$	$\mathbf{63.3_{0.4}}$	$\mathbf{82.5_{0.8}}$

Table 7: Few-Shot Evaluation comparison between language model pre-training on unlabeled data (LMPT) and CONDA. Macro-F1 is used as evaluation metric. All results of CONDA that perform better than LMPT are in bold.

requires more human annotated data to improve beyond some limit.

4.9 Performance vs Few-shot supervision Size

We fix the number of generated samples to 450 per label and vary the number of annotated samples and plot the performance of CONDA-CosmosQA and CONDA-SocialIQA on SST-2 and Yahoo datasets in Figure 4. We also plot the performance of baselines such as BERT-FT, EDA, BackTranslation for comparison. We repeat each experiment with three random seeds and plot the mean performance. We observe that the performance of CONDA increases with the size of supervision and the improvement over baselines in the low-data regime is substantial. For example, with only 4 annotated samples per label in Yahoo dataset, the macro F1 of CONDA-CosmosQA outperforms BERT-FT by 22% and EDA by 15%. However, we also observe that the performance gap between CONDA and baselines decreases with increase in supervision size and gets stagnated after a while. As the size of supervision increases, the supervision by itself is sufficient for high performance, thus reducing the performance boost due to synthetic training data.

4.10 Self-Training

We perform an experiment to demonstrate that the performance can be further improved through self-training when in-domain unlabeled samples are provided. In-domain unlabeled samples are often easily available in real-world scenarios. Self-training is a commonly-used approach to bootstrap the classifier on unlabeled samples (Mekala and Shang, 2020; Mekala et al., 2020; Vu et al., 2021). Following Vu et al. (2021), we obtain pseudo-labels by predicting on unlabeled samples using the trained classifier and train the classifier further on the available labeled and pseudo-labeled data. We consider the training set without ground truth labels as unlabeled data and experiment on SST-2, NYT, and AGNews datasets. We repeat this process for 3 iterations without any filtering of pseudo-labels. From the results in Table 8, we can observe a significant performance improvement up to 4 points with self-training. It is noteworthy that this improvement is consistent for both GPT2-Medium and Large models respectively.

QA Dataset	Setting	SST-2	NYT	AGNews
TweetQA	CONDA	$67.5_{4.9}$	$86.6_{1.3}$	$82.9_{0.9}$
	CONDA + ST	$\mathbf{69.2_{1.3}}$	$\mathbf{88.2_{1.0}}$	$82.4_{1.7}$
	CONDA-L	$74.6_{2.3}$	$84.4_{0.1}$	$79.7_{3.1}$
	CONDA-L + ST	$\mathbf{76.9_{1.1}}$	$\mathbf{87.4_{2.4}}$	$\mathbf{80.9_{3.4}}$
SocialQA	CONDA	$75.2_{1.6}$	$85.8_{1.3}$	$81.7_{0.2}$
	CONDA + ST	$\mathbf{79.8_{0.8}}$	$\mathbf{90.3_{1.9}}$	$\mathbf{83.9_{1.5}}$
	CONDA-L	$77.5_{1.3}$	$89.0_{0.4}$	$80.9_{2.2}$
	CONDA-L + ST	$\mathbf{78.6_{0.6}}$	$\mathbf{92.1_{0.8}}$	$\mathbf{81.1_{1.8}}$
CosmosQA	CONDA	$66.9_{1.2}$	$87.7_{1.1}$	$82.5_{0.8}$
	CONDA + ST	$\mathbf{71.6_{6.9}}$	$87.2_{2.3}$	$\mathbf{83.6_{0.6}}$
	CONDA-L	$77.2_{1.7}$	$86.5_{2.4}$	$79.5_{3.9}$
	CONDA-L + ST	$\mathbf{79.2_{1.3}}$	$\mathbf{87.2_{4.0}}$	$\mathbf{80.7_{3.6}}$

Table 8: Self-Training experiment results with Macro-F1 as evaluation metric. + *ST* denotes with self-training. Self-training improves the performance of both CONDA and CONDA-L significantly. All results where self-training improved the performance are in bold.

4.11 Synthetic Data Adds Value

Unsupervised language model pre-training(LMPT) on target-task unlabeled data can improve performance (Gururangan et al., 2020). We consider training set without ground truth labels as unlabeled data for LMPT and present a comparison in few-shot setting in Table 7. We observe CONDA performs better than LMPT demonstrating the quality and importance of generated synthetic data.

4.12 Case study: Evaluating Context Generator

We hypothesize that our method results in high-quality context generators that are capable of generating context for a given question and answer. To

validate this hypothesis in in-domain and out-of-domain settings, we perform two experiments on QA task.

In-domain Analysis. In this experiment, we validate whether the context generator is capable of generating context for question, answer pairs belonging to the same domain as QA dataset used for QAC fine-tuning. We consider SQuAD dataset and partition it into training set with 1000 (question, answer, context) triplets, dev set of size 1700 with only (question, answer) pairs and a test set of size 6570. First, we consider GPT2-Medium as GLM and perform QAC fine-tuning on the training set. Then, we generate contexts for the dev set and augment the (question, answer, generated context) triplets to the training set. Finally, we train a `BERT-base-uncased` QA model on the augmented data. We compare it with the BERT model trained only on the original training set. We report F1 scores on test set in Table 9. We observe a boost of 4% using our synthetic training data, validating our hypothesis in the in-domain setting.

Out-of-domain Analysis. In this experiment, we validate our hypothesis in the out-of-domain setting i.e. the domain of target dataset is different than the QA dataset used for QAC fine-tuning. We follow our proposed pipeline and consider SQuAD as the QA dataset for QAC fine-tuning and NewsQA as the target dataset. We partition NewsQA dataset into 1000 (question, answer, context) triplets for domain adaptation, 17000 (question, answer) pairs for context generation, and test on 10000 samples. We fine-tune GPT2-medium on SQuAD to obtain general context generator and adapt to the NewsQA domain by training it further on 1000 question, answer, context triplets from NewsQA. Using the target task context generator, we generate contexts for 17000 question, answer pairs, augment it to the training set, and train `BERT-base-uncased` QA model on the augmented data. From F1 scores reported in Table 9, we can observe more than 10% improvement in the performance, demonstrating the efficiency of our method in out-of-domain setting.

Table 9: Case Study: We evaluate our context generators in in-domain and out-of-domain settings. In both cases, we observe substantial improvement in the performance demonstrating the effectiveness of our method.

Setting	Model	F_1 score
In-domain	BERT	32.11
	CONDA	36.74
Out-of-domain	BERT	14.96
	CONDA	25.31

5 Conclusion

In this paper, we propose to train generative language models to be context generators for a given question and answer. To facilitate this, we use question answer as a format and utilize QA datasets for training generative language models into context generators. We view sentiment and topic classification tasks in question-answer form and generate contexts using our fine-tuned generative language models. These generated contexts are used as synthetic training data to augment existing few-shot data for training a classifier. Extensive experiments on multiple sentiment and topic classification datasets demonstrate strong performance of our method in few-shot and zero-shot settings.

6 Limitations

One limitation of our approach is the synthetic training data generated can boost the performance up to an extent and beyond that it requires more annotated samples. So, the generated synthetic training data cannot replace the training data altogether but could minimize the annotation effort significantly. Moreover, some tasks such as NER are challenging to cast into question-answering format, which hinders generating synthetic data using our method.

7 Acknowledgements

We thank anonymous reviewers and program chairs for their valuable and insightful feedback. The research was sponsored in part by National Science Foundation Convergence Accelerator under award OIA-2040727 as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *AAAI*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

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.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training 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.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Dheeraj Mekala, Varun Gangal, and Jingbo Shang. 2021. Coarse2Fine: Fine-grained text classification on coarsely-grained annotated data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 583–594, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333.

Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. Meta: Metadata-empowered weak supervision for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *ArXiv*, abs/1811.01088.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Reid Pryzant, Ziyi Yang, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Automatic rule induction for efficient semi-supervised learning. In *Arxiv*.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. In *Arxiv*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Arxiv*.

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, 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.

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.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. 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*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022a. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. *arXiv preprint arXiv:2205.12647*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022b. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Tu Vu, Minh-Thang Luong, Quoc Le, Grady Simon, and Mohit Iyyer. 2021. STraTA: Self-training with task augmentation for better few-shot learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5715–5731, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Tweetqa: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *ArXiv*, abs/1901.11373.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc.

Table 10: Dataset statistics.

Dataset	# Test Examples
IMDb	25000
Yelp	50000
SST-2	2211
Yahoo	60000
NYT	10582
AGNews	114000

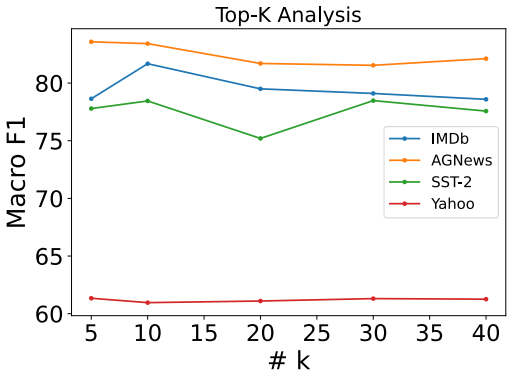


Figure 5: Macro-F_1 scores of CONDA-SocialIQA w.r.t. k. Each experiment is repeated with three different seeds and mean performance is plotted.

A Appendix

A.1 Target Task Datasets

The details of target task datasets are as follows:

- **IMDb:** (Maas et al., 2011) is a movie review dataset with positive and negative as sentiments.
- **Yelp:**[5] is a collection of reviews written by Yelp users with five fine-grained sentiment ratings.
- **SST-2:** (Socher et al., 2013) is a binary sentiment classification dataset with single sentence texts.
- **Yahoo:** (Zhang et al., 2015) is a topic classification dataset with question and answer pairs. Using these pairs, the task is to predict their corresponding topic.
- **The New York Times (NYT):** : contains news articles written and published by The New York Times that are classified into 5 wide genres.
- **AGNews:** (Zhang et al., 2015) is a topic categorization dataset in news domain from AG's corpus.

The size of test sets is mentioned in Table 10.

A.2 Performance vs k

We vary k in top-k sampling and plot the performance of CONDA-SocialIQA on IMDb, SST-2, AGNews, and Yahoo datasets in Figure 5. We fix the few-shot supervision size to 8 samples per label and generate 450 samples per label. We repeat each experiment thrice and plot the mean performance. Upon manual inspection, We observe that the samples generated with k=20 are more diverse than k=10, however, the influence of k on performance is not significant.

A.3 Experiments with a validation set

We perform experiments with a validation set. Since large validation sets are impractical in few-shot settings (Oliver et al., 2018), we consider the validation set to be of same size as the few-shot training set i.e. 8 annotated samples per label. In the experiments with validation set, we perform early stopping based on validation set performance. We present experimental results on few-shot setting with validation set in Table 11. We seldom observe significant improvement upon introducing the validation set. This is because a small validation set which is of same size as few-shot supervision is not large enough to tune the hyperparameters.

A.4 Examples of Generated Training Data

Table 12 shows a few examples of synthetic training data corresponding to IMDb and AGNews datasets generated by our method with all QA datasets.

[5] https://www.yelp.com/dataset/

Table 11: Evaluation Results with validation set.

	Sentiment						Topic					
	IMDb		Yelp		SST-2		NYT		Yahoo		AGNews	
Method	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1	Mi-F_1	Ma-F_1
BERT-FT	$68.7_{3.6}$	$68.5_{3.6}$	$38.1_{5.2}$	$36.3_{5.2}$	$57.5_{1.4}$	$55.8_{2.3}$	$88.7_{5.5}$	$83.9_{4.6}$	$54.4_{1.8}$	$53.9_{1.2}$	$74.6_{5.6}$	$74.7_{5.3}$
ITFT-MNLI	$66.2_{3.3}$	$64.6_{4.2}$	$35.3_{3.7}$	$33.3_{3.5}$	$60.8_{1.8}$	58.71.6	$78.0_{2.9}$	$63.1_{6.2}$	$28.2_{4.5}$	$27.2_{4.2}$	$52.9_{3.2}$	$51.4_{4.5}$
ITFT-SQuAD	$61.1_{2.0}$	$59.7_{2.6}$	$34.0_{2.3}$	$31.6_{3.6}$	$56.5_{1.4}$	$56.0_{1.5}$	$88.9_{2.2}$	$75.8_{4.6}$	$36.2_{4.0}$	$35.3_{4.4}$	$58.2_{5.5}$	$56.2_{6.6}$
BackTranslation	$67.4_{2.0}$	$66.9_{2.0}$	$38.7_{3.6}$	$36.4_{4.4}$	$61.0_{4.3}$	$60.4_{5.0}$	$93.7_{1.4}$	$88.7_{0.3}$	$56.8_{1.4}$	$56.2_{1.2}$	$80.3_{2.0}$	$80.3_{2.0}$
PEGASUS	$66.2_{3.8}$	$65.3_{3.9}$	$32.8_{7.0}$	$29.5_{8.0}$	$61.9_{3.9}$	$60.6_{3.9}$	$93.9_{0.6}$	$87.3_{1.7}$	$57.7_{3.0}$	$56.3_{1.0}$	$79.7_{1.6}$	$79.9_{1.6}$
EDA	$63.3_{4.2}$	$61.6_{4.4}$	$32.5_{6.7}$	$30.6_{8.2}$	$58.8_{2.9}$	$58.1_{3.4}$	$95.7_{0.7}$	$90.6_{2.0}$	$55.7_{1.1}$	$56.3_{1.0}$	$79.8_{0.7}$	$79.9_{0.4}$
CONDA$\backslash QA$	$71.8_{4.5}$	$71.1_{4.9}$	$38.0_{0.3}$	$36.0_{0.5}$	$60.1_{4.7}$	$58.0_{6.2}$	$92.3_{0.5}$	$84.2_{0.5}$	$53.8_{0.9}$	$52.8_{0.7}$	$80.0_{1.6}$	$79.6_{1.7}$
CONDA-SQuAD	$58.5_{2.5}$	$56.5_{1.7}$	$37.6_{1.6}$	$36.4_{0.5}$	$56.3_{2.2}$	$55.8_{1.9}$	$93.4_{0.5}$	$86.6_{0.9}$	$56.1_{1.7}$	$54.8_{1.8}$	$\mathbf{82.1_{0.2}}$	$\mathbf{82.1_{0.2}}$
CONDA-NewsQA	$61.5_{6.7}$	$60.1_{8.1}$	$34.8_{0.9}$	$32.3_{2.5}$	$57.1_{5.8}$	$56.2_{6.2}$	$92.5_{0.8}$	$83.8_{1.5}$	$55.3_{2.5}$	$54.8_{2.8}$	$\mathbf{80.6_{3.7}}$	$80.3_{3.9}$
CONDA-TweetQA	$\mathbf{78.3_{2.8}}$	$\mathbf{78.1_{3.0}}$	$\mathbf{41.3_{0.5}}$	$\mathbf{37.7_{3.5}}$	$\mathbf{71.6_{4.8}}$	$\mathbf{70.9_{5.3}}$	$93.1_{1.5}$	$85.5_{3.0}$	$\mathbf{58.8_{1.6}}$	$\mathbf{57.9_{2.4}}$	$\mathbf{81.2_{1.9}}$	$\mathbf{81.1_{1.9}}$
CONDA-SocialIQA	$\mathbf{78.3_{2.2}}$	$\mathbf{78.1_{1.5}}$	$\mathbf{41.5_{0.8}}$	$\mathbf{39.0_{1.8}}$	$\mathbf{74.4_{3.9}}$	$\mathbf{74.3_{4.0}}$	$92.3_{0.7}$	$84.7_{1.3}$	$\mathbf{58.5_{2.0}}$	$\mathbf{58.0_{2.3}}$	$\mathbf{82.4_{1.6}}$	$\mathbf{82.2_{1.6}}$
CONDA-CosmosQA	$\mathbf{74.1_{4.9}}$	$\mathbf{73.6_{5.5}}$	$\mathbf{38.9_{2.1}}$	$31.3_{4.3}$	$\mathbf{64.5_{2.3}}$	$\mathbf{63.3_{3.1}}$	$93.6_{1.1}$	$86.2_{2.2}$	$\mathbf{59.4_{0.6}}$	$\mathbf{59.3_{0.1}}$	$\mathbf{82.7_{1.3}}$	$\mathbf{82.5_{1.4}}$

Table 12: Example generated texts using our method with all QA datasets

Dataset	Label	QA Dataset	Generated Text
IMDb	good	SQuAD	the film was first screened in the united kingdom and ireland in june 1985, where it received generally favourable reviews.
		NewsQA	i love that "babylon 5" has a story in the form of an animated television show. when the first "babylon" appeared on the vhs channel, i was a kid.
		TweetQA	this is a fantastic film, and i loved the trailer. you may know me and you may also know a friend of mine, so i can totally see
		SocialIQA	skylar went to the movies and saw a good movie.
		CosmosQA	this is one of my favorite movies of all time because the movie was made way back in the 80s and the dialogue is pretty hilarious.
AGNews	sports	SQuAD	this document is about sports, and details the general aspects of the sport. it lays out the sport's rules, and sets out the objectives and common grounds upon which the sport is based.
		NewsQA	(cnn) – when the cincinnati bengals and new england patriots played at a stadium on friday night, an audience of about 2 million people was in tow. and those people weren't the only ones enjoying the game.
		TweetQA	it's a football document and i want to make sure all our fans know. #fcbtampa. thanks for the #baltimorebluhnnnnnnnnnt
		SocialIQA	carson got the ball in their possession after scoring a goal in the soccer match.
		CosmosQA	we just played a nice game, and i thought we 'd be better off if they could come up with a good plan to do it. we did have a little time, however, and we 'd have a chance to score and give the ball back, but it just never worked out