

Domain Adaptation for Question Answering via Question Classification

Zhenrui Yue and Huimin Zeng and Ziyi Kou and Lanyu Shang and Dong Wang
School of Information Sciences

University of Illinois Urbana-Champaign

{zhenrui3, huiminz3, ziyikou2, lshang3, dwang24}@illinois.edu

Abstract

Question answering (QA) has demonstrated impressive progress in answering questions from customized domains. Nevertheless, domain adaptation remains one of the most elusive challenges for QA systems, especially when QA systems are trained in a source domain but deployed in a different target domain. In this work, we investigate the potential benefits of question classification for QA domain adaptation. We propose a novel framework: Question Classification for Question Answering (QC4QA). Specifically, a question classifier is adopted to assign question classes to both the source and target data. Then, we perform joint training in a self-supervised fashion via pseudo-labeling. For optimization, inter-domain discrepancy between the source and target domain is reduced via maximum mean discrepancy (MMD) distance. We additionally minimize intra-class discrepancy among QA samples of the same question class for fine-grained adaptation performance. To the best of our knowledge, this is the first work in QA domain adaptation to leverage question classification with self-supervised adaptation. We demonstrate the effectiveness of the proposed QC4QA with consistent improvements against the state-of-the-art baselines on multiple datasets.

1 Introduction

Question Answering (QA) or Reading Comprehension (RC) refers to the task of extracting answers from given context paragraphs based on input questions. QA systems predict the start and end positions of possible answer spans in given context documents upon input questions. In recent studies, QA systems have achieved significant improvements with transformer models and large-scale datasets (Rajpurkar et al., 2016; Devlin et al., 2019; Yue et al., 2022a).

Once deployed, QA systems often experience performance deterioration upon user-generated questions. Such performance drops can be traced

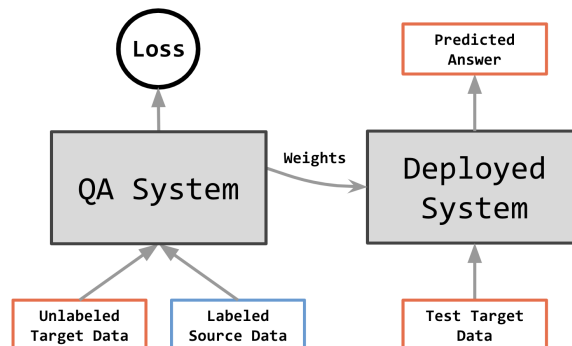


Figure 1: Overview for QA domain adaptation. A QA model is trained with labeled source data and unlabeled target data. The resulting QA system is deployed to answer target questions.

back to domain shifts in two input elements: (1) User-generated questions are syntactically more diverse and thus, different from the training QA pairs; (2) The context domain of test-time input (target domain) can oftentimes diverge from the training corpora (source domain), e.g., from news snippets to biomedical articles (Hazen et al., 2019; Fisch et al., 2019; Miller et al., 2020).

To alleviate the performance issue in QA domain adaptation, several approaches have been proposed to reduce the discrepancy between the source and target domains. Integrating labeled target QA pairs in training can effectively improve the QA system in answering out-of-domain questions (Kamath et al., 2020; Shakeri et al., 2020; Yue et al., 2021, 2022b), where the target data can be human-annotated QA pairs or synthetic data using question generation methods. When only unlabeled questions are available (see Figure 1), another possible approach is to reduce inter-domain discrepancy via domain-adversarial training (Lee et al., 2019). Combined with pseudo labeling, QA systems demonstrate improved generalization in answering target domain questions (Cao et al., 2020).

Nevertheless, previous methods either require large amounts of annotated target data or exten-

sive computing power (Lee et al., 2019; Cao et al., 2020; Yue et al., 2021, 2022b). Additionally, different types of QA pairs and their distributional changes are not taken into account. As a result, existing approaches are less effective for adapting QA systems to an unseen target domain. In this paper, we propose a domain adaptation framework for QA: *question classification for question answering* (QC4QA). Unlike existing methods, we innovatively adopt a question classification (QC) model to classify input questions from both the source and target domains into different question classes. Moreover, we pseudo label the target data using a pretrained QA system and perform distribution-aware sampling to build mini-batches that resemble the target question distribution. In the adaptation stage, we propose a self-supervised adaptation framework to minimize the domain gap, in which inter-domain and intra-class discrepancies are simultaneously regularized. This is in contrast to existing baselines (e.g., domain adversarial adaptation methods) where the source data is solely used for training but without explicitly accounting for domain shifts and question distribution changes (Lee et al., 2019; Cao et al., 2020; Yue et al., 2021). To the best of our knowledge, QC4QA is the first work that combines question classification and self-supervised adaptation for learning domain-invariant representation in QA domain adaptation.

Our main contributions are as follows¹:

1. We propose QC4QA for QA domain adaptation. QC4QA innovatively adopts question classification to identify question types (classes) of the source and target QA pairs for intra-class discrepancy reduction.
2. Our QC4QA can be combined with supervised QC or unsupervised clustering. In the latter case, we show that QC4QA can transfer knowledge to the target domain even without additional model or annotation.
3. We design a distribution-aware sampling strategy and an objective function that incorporates MMD distances for minimizing inter-domain and intra-class discrepancies to transfer knowledge to the target domain.
4. We demonstrate the effectiveness of QC4QA,

¹Our implementation is publicly available at <https://github.com/Yueeeeeee/Self-Supervised-QA>.

where QC4QA consistently outperforms state-of-the-art baselines by a significant margin.

2 Related Work

QA systems have achieved significant improvements in extracting answers upon input context and questions. However, trained QA systems are known to experience performance drops when context paragraphs and questions diverge from the training corpora. That is, when domain shifts exist between the training and test distributions (Hazen et al., 2019; Fisch et al., 2019; Miller et al., 2020; Zeng et al., 2022).

To adapt QA systems for domain changes, methods for QA domain adaptation have been proposed in two different settings: (1) Access to contexts and QA pairs from the target domain. Here, partial access to target data is provided, or a question generation model is introduced for producing synthetic QA pairs. The target data is then used to train and improve adaptation performance (Shakeri et al., 2020; Yue et al., 2021); (2) Access to context paragraphs and unlabeled input questions from the target domain. Here, unsupervised or self-supervised adaptation can be used to improve the performance in the target domain (Cao et al., 2020). In this paper, we focus on the latter setting and study QA domain adaptation with access to target contexts and unlabeled questions.

Domain adaptation in computer vision: Domain adaptation methods have been primarily studied for image classification problems. Such approaches focus on minimizing the representation discrepancy between the source and target distributions. Some methods design objective functions that encourage domain-invariant features in training (Long et al., 2015; Kang et al., 2019). Other methods leverage domain-adversarial training with a discriminator to implicitly impose regularization when source and target features are distinguishable (Tzeng et al., 2017; Zhang et al., 2019a), with successful applications in various vision tasks (Zhang et al., 2019b, 2020, 2021).

Domain adaptation in QA: Various approaches are designed to improve QA performance by generating and refining synthetic QA pairs. Based on target contexts, question generation models are introduced to produce a surrogate dataset, which is used to train QA systems (Kamath et al., 2020; Shakeri et al., 2020; Yue et al., 2022b). Contrastive adaptation minimizes inter-domain discrepancy

ancy with question generation and maximum mean discrepancy (MMD) distances (Yue et al., 2021, 2022c). When unlabeled questions are accessible, domain-adversarial training can be applied to reduce feature discrepancy between domains (Lee et al., 2019). Pseudo labeling and iterative refinements of such labels can be used for improved joint training (Cao et al., 2020).

Question classification (QC): Classifying questions of different types is a common task in natural language processing. One of the widely-used question taxonomy TREC divides questions into 6 coarse classes and 50 fine classes (Li and Roth, 2002). Early machine learning methods perform QC with hand-crafted features (Li and Roth, 2002; Huang et al., 2008). Neural networks improve the classification performance with sentence embeddings (Howard and Ruder, 2018; Cer et al., 2018).

However, the aforementioned approaches in QA domain adaptation encourage domain-invariant features without considering samples from different classes and their distributional changes. Moreover, it is hitherto unclear how to estimate class discrepancies in QA, since class labels are not available in QA datasets. To solve this problem, we propose to use QC to divide QA pairs into different classes, where questions can be classified via an additional QC model or unsupervised clustering with minimum computational costs. We exploit the question classes by reducing the discrepancy among samples from the same class (‘intra-class’). Additionally, we design a distribution-aware sampling strategy in QC4QA to account for distributional changes between the source and target domains. By incorporating the discrepancy terms in the objective function, our self-supervised adaptation framework QC4QA achieves significant improvements against the state-of-the-art baseline methods.

3 Methodology

3.1 Setup

Data: Our setting focuses on improving QA performance when domain shifts exist in the test data distribution. For this purpose, labeled source data and unlabeled target data are available, we denote the domain of source data with \mathcal{D}_s and target data with \mathcal{D}_t . Formally, the input data is defined by:

- *Source data:* Labeled source data \mathbf{X}_s from \mathcal{D}_s . Individual sample $\mathbf{x}_s^{(i)} \in \mathbf{X}_s$ is defined by a triplet consisting of a question $\mathbf{x}_{s,q}^{(i)}$, a

context $\mathbf{x}_{s,c}^{(i)}$, and an answer $\mathbf{x}_{s,a}^{(i)}$. The exact answer tokens can be found in context, answer $\mathbf{x}_{s,a}^{(i)}$ is represented by the start and end position in $\mathbf{x}_{s,c}^{(i)}$.

- *Target data:* Unlabeled target data \mathbf{X}_t from \mathcal{D}_t . For target sample $\mathbf{x}_t^{(i)} \in \mathbf{X}_t$, we only have access to the question $\mathbf{x}_{t,q}^{(i)}$ and context $\mathbf{x}_{t,c}^{(i)}$. Ground truth answer $\mathbf{x}_{t,a}^{(i)}$ is not given for training.

Model: The QA system can be represented with function f . f takes an input question \mathbf{x}_q and context document \mathbf{x}_c as input and yields answer prediction \mathbf{x}_a , namely $\mathbf{x}_a = f(\mathbf{x}_q, \mathbf{x}_c)$. The output \mathbf{x}_a is represented as a subspan of \mathbf{x}_c and comprises of the answer start and end positions.

Objective: The objective is to learn a f^* , which maximizes the performance in answering questions from the target domain \mathcal{D}_t . In other words, f^* minimizes the negative log likelihood (i.e., cross entropy) for \mathbf{X}_t from the target domain distribution:

$$f^* = \arg \min_f \sum_{i=1}^{|\mathbf{X}_t|} \mathcal{L}_{\text{NLL}}(f(\mathbf{x}_{t,q}^{(i)}, \mathbf{x}_{t,c}^{(i)}), \mathbf{x}_{t,a}^{(i)}). \quad (1)$$

3.2 The QC4QA Framework

3.2.1 Overall Framework

In the proposed QC4QA, we design a self-supervised framework that facilitates question classification for QA domain adaptation. QC4QA can be divided into three stages: (1) Question classification; (2) Pseudo labeling & sampling and (3) Self-supervised adaptation. In the first stage, we perform classification for all input questions, which provides additional question class information for the adaptation stage. In the next stage, we label and filter all target samples and perform distribution-aware sampling to build mini-batches that resemble the target data distribution. Finally, we perform self-supervised adaptation on the QA system to minimize inter-domain and intra-class discrepancies. Once input questions are classified, we iteratively perform stage 2 and stage 3 in each epoch. The QA system is trained with both source and target data, where we encourage domain-invariant features and minimize intra-class discrepancies of data samples from the same question class.

Our approach leverages question classification for fine-grained domain adaptation. Here, QC is

designed for evaluating intra-class discrepancies and distributional changes by introducing the additional question classes instead of using QA labels. The idea behind it is that QA labels are defined by subspans in input contexts, if we treat every combination of start and end position as a class, the corresponding label space would be too large and sparse for any meaningful discrepancy estimation. Therefore, we propose the question classification stage to introduce additional semantic knowledge for intra-class discrepancy estimation. Moreover, by performing pseudo labeling and distribution-aware sampling, we resemble the target question distribution in the adaptation stage to correct the potential bias in the pretrained QA system. In other words, QC4QA simulates the target data distribution and ‘pulls together’ source and target samples of the same question class to encourage domain invariance.

3.2.2 Question Classification

For question classification, we adopt the commonly used question taxonomy in TREC and categorize all questions into 6 coarse classes Q : **ABBR**: *Abbreviation*, **DESC**: *Description*, **ENTY**: *Entity*, **HUM**: *Human*, **LOC**: *Location* and **NUM**: *Numeric Value*. Each class indicates the potential answer type to the question (Li and Roth, 2002). In practice, we rarely find **ABBR** questions.

The proposed QC model leverages pretrained sentence embedding methods to generate vectorized feature for input questions. We then build a multilayer perceptron (MLP) to perform classification on the encoded questions, see Figure 2. Specifically, we adopt InerSent and Universal Sentence Encoder to encode the input questions separately (Conneau et al., 2017; Cer et al., 2018). The encodings are concatenated and used as an input feature for the MLP classifier. With the trained QC model, inference can be performed on all training questions for the later adaptation stage.

To further examine the effectiveness of question classification without additional model and annotation, we introduce an unsupervised clustering method, where we refrain from using an additional dataset or classifier to perform question classification. In particular, we feed the input data within the transformer encoder (part of the QA system) and utilize the output from the [CLS] token position as features (Devlin et al., 2019). We sample a fixed number of source features (10k in our experiments) and perform KMeans clustering with a predefined

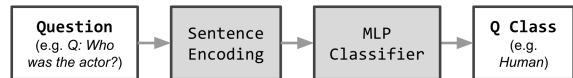


Figure 2: Model architecture for question classifier.

number of clusters k (Similar to TREC, we use 5 as default). Then, cluster centroids are preserved to classify source and target QA datasets.

3.2.3 Pseudo Labeling & Sampling

Provided with the access to labeled source data, we pretrain the QA system f to learn to answer questions. After pretraining, we can use f to predict target answers for self-supervised adaptation. The pseudo labels are filtered according to the answer confidence, we preserve the target samples above confidence threshold λ_{con} . The pseudo labeling and confidence thresholding steps are repeated in each epoch to dynamically adjust the target distribution used for training.

For mini-batch training, we sample the same amounts of QA pairs from both domains to minimize the inter-domain and intra-class discrepancies. However, with randomly sampled data, training is less efficient as source and target questions in each batch can be entirely different (e.g., source samples are all *Human* questions and target samples are all *Description* questions). To solve this problem, we design a distribution-aware sampling strategy: we first sample target QA pairs from X_t and within the same question classes, we sample from X_s such that the source and target question classes in each batch are identical. Consequently, the QA system can be trained on a data distribution similar to the target dataset. Moreover, the estimation of intra-class discrepancy between both domains can be performed more efficiently.

3.2.4 Self-supervised Adaptation

The sampled batches are used to adapt the pretrained QA system, where we optimize the model to reduce the negative log likelihood loss, as in Equation (1). Meanwhile, we encourage the domain invariance by computing the discrepancies and incorporate them in the training objective.

To measure the discrepancy between samples from different domains, we adopt the maximum mean discrepancy (MMD) distance (Gretton et al., 2012). MMD estimates the distance between two distributions with samples drawn from them, with f and \mathcal{H} representing the feature mapping and the

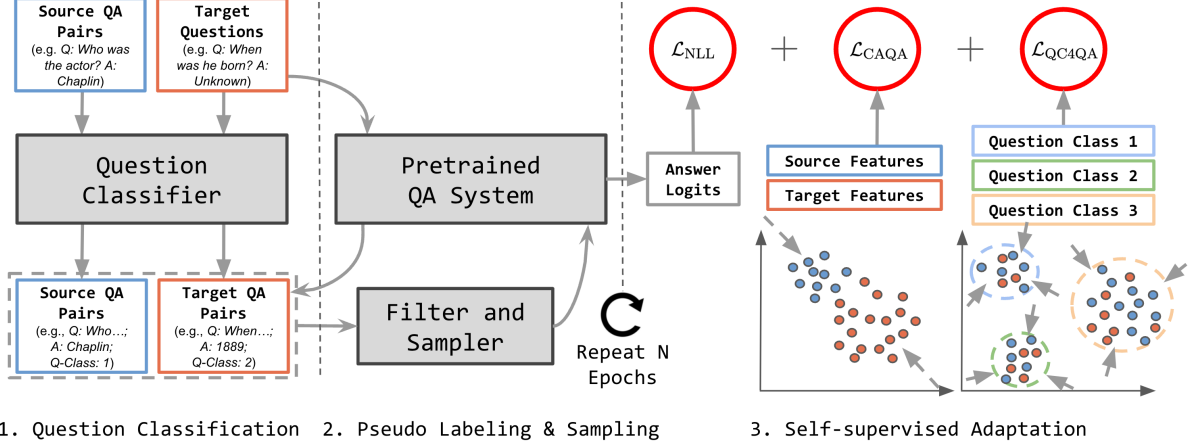


Figure 3: Overview for the proposed method. QC4QA can be divided into three stages: (1) Question classification where all questions are assigned different classes; (2) Pseudo labeling & sampling, where we label and sample target examples with the proposed distribution-aware sampling strategy; (3) Self-supervised adaptation, in which we train the QA system jointly with source and target data. In the experiments, stage 2 and 3 are iteratively performed and we apply the proposed objective Equation (5) to minimize both inter-domain and intra-class discrepancies.

reproducing kernel Hilbert space:

$$\mathcal{D} = \sup_{f \in \mathcal{H}} \left(\frac{1}{|\mathbf{X}_s|} \sum_{i=1}^{|\mathbf{X}_s|} f(\mathbf{x}_s^{(i)}) - \frac{1}{|\mathbf{X}_t|} \sum_{i=1}^{|\mathbf{X}_t|} f(\mathbf{x}_t^{(i)}) \right). \quad (2)$$

To simplify the computation, we adopt the Gaussian kernel as feature mapping, i.e., $k(\mathbf{x}_s^{(i)}, \mathbf{x}_t^{(j)}) = \exp(-\frac{\|\mathbf{x}_s^{(i)} - \mathbf{x}_t^{(j)}\|^2}{\gamma})$. We further leverage empirical kernel mean embeddings (Long et al., 2015) to estimate the MMD distance between samples from \mathbf{X}_s and \mathbf{X}_t :

$$\begin{aligned} \mathcal{D}^{\text{MMD}} &= \frac{1}{|\mathbf{X}_s||\mathbf{X}_s|} \sum_{i=1}^{|\mathbf{X}_s|} \sum_{j=1}^{|\mathbf{X}_s|} k(\phi(\mathbf{x}_s^{(i)}), \phi(\mathbf{x}_s^{(j)})) \\ &+ \frac{1}{|\mathbf{X}_t||\mathbf{X}_t|} \sum_{i=1}^{|\mathbf{X}_t|} \sum_{j=1}^{|\mathbf{X}_t|} k(\phi(\mathbf{x}_t^{(i)}), \phi(\mathbf{x}_t^{(j)})) \\ &- \frac{2}{|\mathbf{X}_s||\mathbf{X}_t|} \sum_{i=1}^{|\mathbf{X}_s|} \sum_{j=1}^{|\mathbf{X}_t|} k(\phi(\mathbf{x}_s^{(i)}), \phi(\mathbf{x}_t^{(j)})), \end{aligned} \quad (3)$$

where ϕ represents the transformer encoder in the QA system. With \mathcal{D}^{MMD} , it is possible to measure the discrepancies between different domains and question classes. The discrepancy values are used to guide the self-supervised adaptation and encourage domain-invariant features.

Among all tokens in each QA sample \mathbf{x} , we distinguish two types of features \mathbf{x}_a and \mathbf{x}_o . \mathbf{x}_a stands for the mean vector of answer token representations, while \mathbf{x}_o is the mean vector of all other

tokens in the representation space (Yue et al., 2021). The QA system extracts the answer when the answer tokens in the representation space are separated from \mathbf{x}_o (van Aken et al., 2019). Therefore, we adopt \mathbf{x}_a as feature representation to compute MMD distances. By introducing question classification, we introduce an additional term w.r.t. the intra-class discrepancy in the objective function:

$$\mathcal{L}_{\text{QC4QA}} = \frac{1}{|Q|} \sum_{q \in Q} \mathcal{D}^{\text{MMD}}(\mathbf{X}_s^{(q)}, \mathbf{X}_t^{(q)}), \quad (4)$$

\mathbf{X}_s refers to all answer features in source samples and \mathbf{X}_t represents answer features in target samples, while $\mathbf{X}^{(q)}$ denotes the set of samples that belong to question class $q \in Q$. $\mathcal{L}_{\text{QC4QA}}$ ‘pulls together’ features from the same question class across domains by minimizing their MMD distances.

3.2.5 Overall Objective

To encourage domain-invariant features, we incorporate Equation (4) into the training objective. Using the NLL loss and the contrastive adaptation loss (Yue et al., 2021, 2022c), the overall objective function can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \lambda(\mathcal{L}_{\text{CAQA}} + \mathcal{L}_{\text{QC4QA}}), \quad (5)$$

in which $\mathcal{L}_{\text{CAQA}}$ is the same as in (Yue et al., 2021), while λ is a scaling factor we choose empirically. Although we introduce $\mathcal{L}_{\text{CAQA}}$ in our training objective function, QC4QA is largely different from CAQA as: (1) $\mathcal{L}_{\text{CAQA}}$ only reduces the

inter-domain discrepancy, we incorporate question classification to additionally reduce intra-class discrepancy via $\mathcal{L}_{\text{QC4QA}}$ for fine-grained adaptation; (2) We perform pseudo labeling and distribution-aware sampling to account for the distribution shifts between the source and target dataset; and (3) QC4QA leverages an efficient self-supervised adaptation framework instead of the computationally expensive question generation in (Yue et al., 2021). As such, the proposed QC4QA efficiently reduces the domain discrepancy and effectively transfers learnt knowledge from the source domain to the target domain.

The overall framework is illustrated in Figure 3. We first generate question classes for all data samples. Next, the source-pretrained QA model generates pseudo labels for target data and we select target samples above the confidence threshold λ_{con} for training. Pseudo labeling and self-supervised adaptation are performed iteratively to refine the pseudo labels and improve the performance in the target domain using Equation (5). Unlike previous works (Lee et al., 2019; Cao et al., 2020; Shakeri et al., 2020; Yue et al., 2021), we discard domain-adversarial training or question generation and introduce a self-supervised adaptation framework based on question classification for improved efficiency and adaptation performance. We also design a distribution-aware sampling strategy to resemble the target data distribution and correct the potential bias in the pretrained QA system. Additionally, a fine-grained adaptation loss based on question classification is introduced in training to minimize both the inter-domain and intra-class discrepancies across the source and target domain.

4 Experiments

4.1 Datasets and Baselines

For supervised question classification, we adopt the TREC dataset (Li and Roth, 2002), a widely used dataset containing $\sim 5\text{k}$ training questions and 500 questions for testing. Following (Cao et al., 2020; Shakeri et al., 2020; Yue et al., 2021), we use SQuAD as our source domain QA dataset (Rajpurkar et al., 2016). For target domain, we adopt multiple QA datasets (details in Appendix A) and refrain from using labels in training (Cao et al., 2020; Shakeri et al., 2020; Yue et al., 2021).

For comparison, we adopt 4 baseline methods. We first pretrain a QA system on the source dataset and then evaluate on each target dataset with zero

knowledge of the target domain. We additionally adopt 3 state-of-the-art baselines: Domain-adversarial training (DAT) (Lee et al., 2019), conditional adversarial self-training (CASE) (Cao et al., 2020) and contrastive adaptation for QA (CAQA*) (Yue et al., 2021). For fair comparison, we adapt the original CAQA to our self-supervised adaptation framework as a baseline, we denote the adapted CAQA with CAQA*.² BERT-QA is selected as the QA model (Devlin et al., 2019). Details of the baselines are elaborated in Appendix A.

4.2 Training and Evaluation

We train our QC model on the TREC training set and evaluate on the test set, the best model is saved to perform classification on all QA datasets. For unsupervised question classification, sampled [CLS] features from the source dataset are used to perform KMeans clustering, followed by question class inference on all QA datasets.

After question classification, we adopt a QA model (pretrained on the source dataset) and iteratively perform: (1) Pseudo labeling and distribution-aware sampling to select data batches that resemble the target data distribution; (2) Self-supervised adaptation with the proposed objective Equation (5) for learning domain-invariant representation. For evaluation, we adopt two metrics: exact match (EM) and F1 score (F1). We compute the metrics on target dev sets to evaluate the adaptation performance. Details of our implementation can be found in Appendix B.

4.3 Main Results

We first report the question classification performance on TREC dataset. The MLP classifier has 2.36M parameters and can be trained efficiently in less than one minute (57.6s on average) with GPU acceleration. We perform the evaluation with the proposed MLP QC model and reach an accuracy of 96.6% on the TREC test set. Similar magnitude of efficiency can be observed in KMeans clustering for unsupervised question classification. We provide detailed quantitative analysis and qualitative examples in Appendix C.1.

The QA system is first pretrained in the source domain with 79.60 EM and 87.64 F1 score on

²We exclude question generation and adopt the same process of pseudo labeling, distribution-aware sampling and self-supervised adaptation as QC4QA in CAQA*. Different from QC4QA, we use the same objective function as in (Yue et al., 2021). A direct comparison between the proposed QC4QA and the original CAQA can be found in Appendix C.2.

Model	CNN EM / F1	Daily Mail EM / F1	NewsQA EM / F1	HotpotQA EM / F1	SearchQA EM / F1
(I) Zero-shot target performance					
BERT-QA	14.30/23.57	15.38/25.90	39.17/56.14	43.34/60.42	16.19/25.03
(II) Target performance with domain adaptation					
DAT (Lee et al., 2019)	21.89/27.37	26.98/32.72	38.73/54.24	44.25/61.10	22.31/31.64
CASe (Cao et al., 2020)	20.77/29.37	25.40/35.85	43.43/59.67	47.16/63.88	26.07/35.16
CAQA* (Yue et al., 2021)	21.97/30.97	32.08/41.47	44.26/60.83	48.52/64.76	32.05/41.07
QC4QA KMeans (Ours)	<u>25.04/33.20</u>	<u>35.53/44.32</u>	<u>44.40/60.91</u>	<u>49.58/65.78</u>	<u>34.44/43.78</u>
QC4QA TREC (Ours)	28.05/36.18	36.43/45.85	45.62/61.71	50.02/66.10	35.75/44.37

Table 1: Main results of QA adaptation performance on target dataset.

Model	CoQA EM / F1	DROP EM / F1	Natural Questions EM / F1	TriviaQA EM / F1
(I) Zero-shot target performance				
BERT-QA	12.42/17.30	19.36/30.28	39.06/53.75	49.70/59.09
(II) Target performance with domain adaptation				
DAT (Lee et al., 2019)	11.98/14.72	18.53/29.34	44.94/58.91	49.94/59.82
CASe (Cao et al., 2020)	13.71/18.57	21.78/31.44	46.53/60.19	54.74/63.61
CAQA* (Yue et al., 2021)	14.41/19.28	22.48/31.56	47.37/60.52	54.30/62.98
QC4QA KMeans (Ours)	<u>14.83/19.60</u>	<u>23.13/31.73</u>	<u>49.37/62.25</u>	<u>54.99/63.58</u>
QC4QA TREC (Ours)	15.03/19.71	23.46/32.22	50.59/62.98	55.98/64.57

Table 2: Results of QA adaptation performance on additional target dataset.

the SQuAD dev set. Then, we perform adaptation experiments and report the main results in Table 1, results on the additional target datasets can be found in Table 2. Both tables are divided into 2 parts: (1) QA systems pretrained on SQuAD as naïve baseline (‘Zero-shot target performance’); (2) Baseline methods and QC4QA for QA domain adaptation (‘Target performance with domain adaptation’). The proposed approach with TREC supervised classification is denoted with ‘QC4QA TREC’, unsupervised KMeans question clustering is denoted with ‘QC4QA KMeans’.

The following observations can be made from our experiments: (1) Unsupervised adaptation methods achieve superior performance than the naïve baseline in most cases. Compared to the naïve baseline, QC4QA can lead to improvements of over 100% in EM and over 75% in F1. (2) Compared to contrastive adaptation (e.g., CAQA*), the proposed QC4QA is particularly effective on cloze questions (i.e., CNN and Daily

Mail), with average performance gains of 16.5% and 10.4% in EM and F1. This suggests that we can benefit more from QC when the target questions are less similar to source questions. (3) By comparing CAQA* and both QC4QA methods, we find consistent performance improvements due to question classification for all datasets. (4) Both QC4QA methods outperform baseline methods with considerable improvements, from which QC4QA TREC demonstrates the best performance on all datasets. For example, QC4QA KMeans significantly outperforms the best baseline CAQA* with 5.2% and 3.2% performance increases in EM and F1 on average. For QC4QA TREC, the relative improvements are 8.6% and 5.5% respectively. Altogether, the results suggest that both the TREC and KMeans question classification are effective for improving the performance on out-of-domain data. Additional results and analysis can be found in Appendix C.

4.4 Ablation Studies

4.4.1 Question Classification

We first study the benefits of question classification. The performance gains can be achieved by comparing the results between CAQA* and QC4QA in Table 1 and Table 2. This is because CAQA* is adapted to the same self-supervised adaptation framework as in QC4QA. CAQA* models are trained without minimizing the intra-class discrepancies in Equation (5). The improvements from QC4QA suggest that both supervised and unsupervised question classification can consistently improve QA systems in answering questions from unseen domains. Moreover, QC4QA is particularly effective on target datasets with different question formats (e.g., CNN and Daily Mail).

4.4.2 Distribution-aware Sampling

To study the influence of distribution-aware sampling in QC4QA, we replace the distribution-aware sampling strategy with random sampling. Then we perform unsupervised adaptation with QC4QA TREC on CNN, Daily Mail and NewsQA to verify the merits of the sampling strategy. Results are presented in Table 3.

Dataset	Rand. Sampling EM / F1	QC4QA EM / F1
CNN	26.69/35.24	28.05/36.18
Daily Mail	35.83/45.51	36.43/45.85
NewsQA	44.72/61.02	44.86/61.40

Table 3: QC4QA performance with the random sampling strategy.

In all target datasets, we see performance drops when we replace the proposed strategy with random sampling. In particular, we find relatively large performance deterioration on CNN without distribution-aware sampling. We believe the reason is that the question distribution in CNN is less similar to SQuAD (see Table 5), the resulting inconsistency in sampled batches reduces the effectiveness in discrepancy estimation.

4.4.3 Sensitivity of Hyperparameter λ

Now we evaluate the influence of λ to study the robustness of the proposed objective function. We select different values ranging from 0 to $5e-2$ and perform adaptation with QC4QA TREC. Experiments on CNN, Daily Mail and NewsQA are presented to estimate the influence of λ .

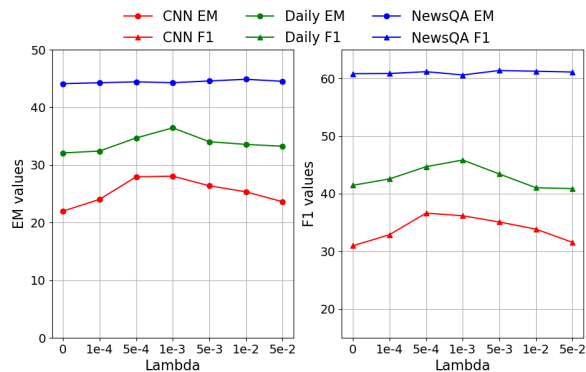


Figure 4: QC4QA adaptation performance for different lambda values. X-axis represents lambda and y-axis represents EM / F1 scores.

Figure 4 visualizes EM / F1 with increasing λ . Despite certain variations, we observe the results first go up and then decrease. Although CNN and Daily are more sensitive to λ , we observe greater improvements by reducing inter-domain and intra-class discrepancies. Overall, QC4QA consistently improves adaptation performance.

4.4.4 Confidence Threshold in Pseudo labeling

We study the influence of λ_{con} to understand how the performance varies with different confidence thresholds in pseudo labeling. We select different threshold values ranging from 0.2 to 0.8 to filter pseudo labels and train with QC4QA TREC. Experiments are performed on HotpotQA and SearchQA to estimate the influence of λ_{con} .

λ_{con} Selection	HotpotQA EM / F1	SearchQA EM / F1
0.2	48.12/64.56	25.60/34.50
0.4	50.02/66.10	33.56/42.62
0.6	49.95/69.84	35.75/44.37
0.8	49.44/65.11	37.29/46.31

Table 4: QC4QA adaptation performance for different confidence thresholds in pseudo labeling.

Table 4 shows adaptation performance with different λ_{con} . The best performance can be reached with λ_{con} ranging from 0.4 to 0.8. For large datasets like SearchQA (with over 100k QA pairs), a higher confidence threshold yields a better adaptation performance since we avoid noisy pseudo labels. In sum, carefully selected λ_{con} yields comparatively large improvements for QC4QA.

5 Conclusion

In this paper, we propose a novel framework for QA domain adaptation. The proposed QC4QA combines question classification with self-supervised adaptation techniques. QC4QA leverages question classes to reduce domain discrepancies and resemble target data distribution in training. Different from existing works, QC4QA achieves superior performance by introducing a simple question classifier and incorporating the question class information in the training objective. We demonstrate the efficiency and effectiveness of QC4QA compared to state-of-the-art approaches by achieving a substantially better performance on multiple datasets.

Despite having adopted question classification to adapt QA systems to unseen target domains, the proposed QC4QA has certain limitations. For example, we assume access to unlabeled questions in QA datasets and have not exploited the potential benefits of different question samples and question classes. For future work, we plan to relax our settings and explore question generation and question value estimation for QA domain adaptation.

Acknowledgments

This research is supported in part by the National Science Foundation under Grant No. IIS-2202481, CHE-2105005, IIS-2008228, CNS-1845639, CNS-1831669. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. 2020. Unsupervised domain adaptation on reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7480–7487.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. **Universal sentence encoder for English**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. **Supervised learning of universal sentence representations from natural language inference data**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. **MRQA 2019 shared task: Evaluating generalization in reading comprehension**. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Timothy J Hazen, Shehzaad Dhuliawala, and Daniel Boies. 2019. Towards domain adaptation from limited data for question answering using deep neural networks. *arXiv preprint arXiv:1911.02655*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

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.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. **Question classification using head words and their hypernyms**. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language*

- Processing*, pages 927–936, Honolulu, Hawaii. 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.
- 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.
- Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. 2019. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902.
- 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.
- Seanie Lee, Donggyu Kim, and Jangwon Park. 2019. [Domain-agnostic question-answering with adversarial training](#). In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 196–202, Hong Kong, China. Association for Computational Linguistics.
- Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR.
- John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR.
- 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.
- Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. [End-to-end synthetic data generation for domain adaptation of question answering systems](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.
- Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176.
- Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832.
- 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.
- Xiang Yue, Xiaoman Pan, Wenlin Yao, Dian Yu, Dong Yu, and Jianshu Chen. 2022a. [C-MORE: Pretraining to answer open-domain questions by consulting millions of references](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 371–377, Dublin, Ireland. Association for Computational Linguistics.
- Xiang Yue, Ziyu Yao, and Huan Sun. 2022b. [Synthetic question value estimation for domain adaptation of question answering](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1340–1351, Dublin, Ireland. Association for Computational Linguistics.
- Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. [Contrastive domain adaptation for question answering using limited text corpora](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022c. Contrastive domain adaptation for early misinformation detection: A case study on covid-19. In *arXiv preprint arXiv:2208.09578*.

- Huimin Zeng, Zhenrui Yue, Yang Zhang, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022. On attacking out-domain uncertainty estimation in deep neural networks. In *IJCAI*.
- Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. 2019a. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5040.
- Yang Zhang, Daniel Zhang, and Dong Wang. 2021. On migratable traffic risk estimation in urban sensing: A social sensing based deep transfer network approach. *Ad Hoc Networks*, 111:102320.
- Yang Zhang, Ruohan Zong, Jun Han, Hao Zheng, Qi-uwen Lou, Daniel Zhang, and Dong Wang. 2019b. Transland: An adversarial transfer learning approach for migratable urban land usage classification using remote sensing. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1567–1576. IEEE.
- Yang Zhang, Ruohan Zong, and Dong Wang. 2020. A hybrid transfer learning approach to migratable disaster assessment in social media sensing. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 131–138. IEEE.

A Dataset and baseline details

A.1 Dataset details

For QA datasets, we follow (Lee et al., 2019; Cao et al., 2020; Yue et al., 2021) and select SQuAD v1.1 as our source dataset (Rajpurkar et al., 2016). SQuAD is a crowdsourced QA dataset based on Wikipedia articles. For target domain, we adopt multiple datasets to evaluate QC4QA:

1. **CNN** (Hermann et al., 2015) leverages CNN articles as contexts. Cloze QA pairs are generated by replacing answers with ‘@placeholder’.
2. **CoQA** (Reddy et al., 2019) is a conversational dataset with rationales and QA pairs. Contexts are given as multi-turn conversations.
3. **Daily Mail** (Hermann et al., 2015) is similar to CNN and consists of news from Daily Mail. Cloze questions and answers are used.
4. **DROP** (Dua et al., 2019) requires QA systems to resolve references, reasoning, matching and understanding context implications.
5. **NewsQA** (Trischler et al., 2016) provides news as contexts and challenging questions beyond simple matching and entailment.
6. **HotpotQA** (Yang et al., 2018) provides multi-hop questions with challenging contexts (distractor contexts excluded).
7. **Natural Questions** (Kwiatkowski et al., 2019) has user questions. We adopt short answers and use long answers as contexts.
8. **TriviaQA** (Joshi et al., 2017) is a large-scale QA dataset that includes QA pairs and supporting facts for supervised training.
9. **SearchQA** (Dunn et al., 2017) is constructed through existing QA pairs by searching for context from online search results.

A.2 Baseline details

For naïve baseline, we adopt BERT-QA (uncased base version with additional batch normalization layer) and train on the source dataset (Devlin et al., 2019; Cao et al., 2020). Additionally, we select 3 baselines in unsupervised QA domain adaptation:

1. **Domain adversarial training (DAT)** (Lee et al., 2019) comprises of a QA system and a discriminator using [CLS] output in BERT. The QA system is first trained on labeled source data. Then, input data from both domains is used for domain-adversarial training to learn generalized features.
2. **Conditional adversarial self-training (CAsE)** (Cao et al., 2020) leverages self-training with domain-adversarial learning. CAsE iteratively perform self-training and domain adversarial training to reduce domain discrepancy. We adopt the entropy weighted version CAsE+E in our work as baseline.
3. **Contrastive adaptation for QA (CAQA*)** (Yue et al., 2021) proposes contrastive adaptation based on token-level features. CAQA utilizes answer tokens as features and reduce the domain gap by minimizing MMD distances. We exclude question generation and adopt the same process of pseudo labeling, distribution-aware sampling and self-supervised adaptation. In particular, we perform training using the original contrastive adaptation loss as in (Yue et al., 2021).

B Implementation

We first train a question classifier on the TREC dataset. The QC model is trained for 4 epochs using RMSprop optimizer with learning rate of 0.01 and batch size of 64. We evaluate the QC model on the TREC test set and report the accuracy of the best QC model.

For pretraining BERT-QA on the source dataset (i.e., SQuAD), we follow (Devlin et al., 2019; Yue et al., 2021) to preprocess data and perform training. We select the AdamW optimizer and train BERT-QA for 2 epochs without linear warmup. Learning rate is 3e-5 and batch size is 12. After pretraining, we validate the model with the provided dev set and report the EM and F1 scores.

For baseline methods, we use our pretrained BERT-QA and follow their default settings for domain adaptation. For QC4QA, adaptation is performed 4 epochs with the AdamW optimizer, learning rate of 3e-5 and 10% proportion as warmup in training. In the pseudo labeling stage, we first perform inference on unlabeled target data and preserve the target samples above confidence threshold

λ_{con} . For batching in self-supervised adaptation stage, we sample 12 target examples and perform distribution-aware sampling to sample another 12 source QA pairs. The sampled source data has same question classes as the target examples. Validation is performed every 2000 iterations and after every epoch to save the best QA model. In our experiments, we empirically select λ from [1e-4, 1e-3, 1e-2], we select λ_{con} from [0.4, 0.6]. Our system setup is Intel Xeon Gold 6326 CPU, NVIDIA A40 GPU and 128GB RAM.

C Additional results

C.1 Question Classification Results

Due to our light-weight design, the TREC question classifier can perform training and inference efficiently within a few minutes. For example, we achieve an average training time of 57.6s on TREC in repeated experiments with GPU acceleration. Inference on QA datasets are of similar efficiency and depends on the individual size of each dataset.

Since TREC classes are not provided in QA datasets, it’s not possible to directly evaluate the supervised QC model on them. We report the distribution of different question classes in Table 5, where we observe significant distribution shifts between the source dataset and certain target datasets (e.g., CNN and Daily Mail). Additionally, we present selected examples of classified questions in Table 9, from which we observe the following: (1) Cloze questions are more difficult to classify. Unlike natural questions, cloze questions usually do not contain auxiliary verb and wh-words (e.g., what, where etc.) as indicator of the question classes. (2) Multiple question classes may qualify for cloze questions. In some examples, different types of tokens can be filled in the placeholder position (e.g., both **DESC** and **ENTY** qualify for Q3). (3) The TREC question classifier can be less accurate on cloze questions. This is the case for Q5 in Table 9, where the questions are more likely to be **DESC** and **LOC** than **HUM**. (4) For natural questions, the question classifier performs generally well and makes fewer mistakes due to the similarity of natural questions across QA datasets. More examples can be found in the released code and data.

For KMeans unsupervised question classification, we focus on the discrepancy among question samples and perform KMeans clustering using the [CLS] output from BERT encoder, see Figure 5. The plot shows a principle component

analysis (PCA) visualizing the BERT-encoder output of NewsQA examples, where different colors represent question class predictions via KMeans algorithm. We observe that [CLS] features are comparatively homogeneous, making it hard to determine cluster boundaries that clearly separate different classes of questions. This might cause performance deterioration in case of increasing outliers. Overall, KMeans can successfully cluster QA examples within each neighborhood on the target dataset. Ideally, the cluster labels can be used to reduce intra-class discrepancies for fine-grained domain adaptation similar to TREC classification. Both adaptation results and cluster visualization suggest that KMeans is effective in improving the performance on out-of-domain data.

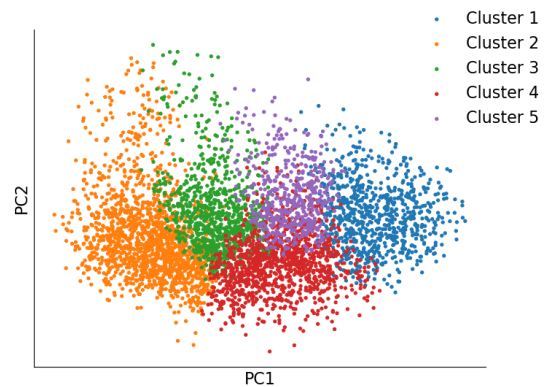


Figure 5: Visualization of KMeans cluster analysis on NewsQA using [CLS] features from BERT.

C.2 Comparison with CAQA

We study the effectiveness of the proposed QC4QA by comparing the performance between QC4QA, the original CAQA and the adapted CAQA* (Yue et al., 2021). The results are presented in Table 6, we observe that the best-performing method is QC4QA with TREC question classification for 7 out of 8 metric values. For SearchQA, the original CAQA performs the best in EM, with QC4QA TREC is of similar magnitude and clearly ranks second. On average, QC4QA TREC performs the best with EM of 48.09 and F1 of 59.51. Despite discarding question generation using the T5 transformer, QC4QA KMeans and the original CAQA performs similarly. Interestingly, we observe that CAQA* outperforms the original CAQA in HotpotQA, suggesting that the distribution-aware sampling and iterative pseudo-processing can effectively improve

Dataset	ABBR	DESC	ENTY	HUM	LOC	NUM
SQuAD	0.5%	12.5%	31.4%	19.9%	11.6%	24.1%
CNN	0.0%	5.3%	39.2%	43.6%	5.3%	6.5%
Daily Mail	0.0%	3.9%	38.6%	46.1%	4.4%	6.9%
NewsQA	0.1%	19.5%	21.4%	29.4%	10.3%	19.3%
HotpotQA	0.0%	1.6%	21.1%	51.6%	14.8%	10.9%
CoQA	0.1%	14.1%	14.2%	53.7%	9.3%	8.5%
DROP	0.0%	2.3%	24.0%	51.8%	7.3%	14.5%
Natural Questions	0.2%	5.0%	13.2%	39.2%	13.4%	29.0%
SearchQA	0.2%	3.7%	43.1%	4.8%	13.5%	34.7%
TriviaQA	0.2%	1.6%	37.8%	36.5%	19.3%	4.7%

Table 5: Question class distribution in all datasets.

Model	Natural Questions	HotpotQA	SearchQA	TriviaQA
	EM / F1	EM / F1	EM / F1	EM / F1
(I) Zero-shot target performance				
BERT-QA	39.06/53.75	43.34/60.42	16.19/25.03	49.70/59.09
(II) Target performance with domain adaptation				
CAQA* (Yue et al., 2021)	47.37/60.52	48.52/64.76	32.05/41.07	54.30/62.98
CAQA (Yue et al., 2021)	48.55/62.60	46.37/61.57	36.05/42.94	55.17/63.23
QC4QA KMeans (Ours)	<u>49.37/62.25</u>	<u>49.58/65.78</u>	34.44/43.78	<u>54.99/63.58</u>
QC4QA TREC (Ours)	50.59/62.98	50.02/66.10	35.75/44.37	55.98/64.57

Table 6: Comparison between QC4QA and CAQA.

the adaptation performance.

C.3 Cluster Number in QC4QA KMeans Classification

To study the influence of cluster number in unsupervised question clustering for QC4QA, we initialize KMeans algorithm with different number of clusters. Then we perform QC4QA KMeans adaptation on HotpotQA and SearchQA to examine the influence of cluster number.

Cluster Number	HotpotQA EM / F1	SearchQA EM / F1
3	49.82/65.73	30.34/39.49
5	49.58/65.78	32.19/41.27
7	49.87/65.66	30.51/39.25
9	50.45/66.14	29.59/38.14

Table 7: QC4QA adaptation performance for different numbers of KMeans clusters.

Results are presented in Table 7, we observe performance drops when we reduce number of clusters from the default of 5. Surprisingly, the performance on HotpotQA grows consistently with increasing number of clusters. A potential explanation for such improvements is that fine-grained question

classification is more helpful for complex multi-hop QA datasets.

C.4 Human Annotation

Training Method	Daily Mail EM / F1	NewsQA EM / F1
0 Annotation	36.43/45.85	44.86/61.40
5k Annotations	48.04/56.27	45.71/62.21
10k Annotations	55.37/61.95	47.17/63.46
20k Annotations	66.83/72.18	48.72/64.92

Table 8: Semi-supervised adaptation performance with QC4QA.

We also study the influence of human annotations by introducing labeled target examples. We present the results on Daily Mail and NewsQA in Table 8. We observe that human annotations improve the adaptation performance in general. With the increasing amount of annotations, the performance gains of QC4QA rise rapidly and then stay steady. In both cases, introducing limited annotations can significantly improve model performance. The results indicate that the introduction of even limited amount of annotations helps QA systems reach comparable magnitude of supervised results.

TREC classification examples

- Q1:** Judges in @placeholder and Oregon this week overturn marriage bans. → **DESC**
- Q2:** Spain international Mata close to joining English club @placeholder. → **ENTY**
- Q3:** The Surprise will be sold in 120 @placeholder stores, costing 1.75 for four? → **ENTY**
- Q4:** School bus drivers union will strike wednesday if it doesn't reach deal with @placeholder. → **HUM**
- Q5:** Serial killer Israel Keyes may have killed missing @placeholder woman. → **HUM**
- Q6:** Which is the latest version of corel draw? → **ENTY**
- Q7:** Who did say South Africa did not issue a visa on time? → **HUM**
- Q8:** Census bureaus are hiring people from where? → **LOC**
- Q9:** How long was the lion's longest field goal? → **NUM**
- Q10:** Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who? → **HUM**
- Q11:** To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France? → **HUM**
- Q12:** When did the Scholastic Magazine of Notre dame begin publishing? → **NUM**
- Q13:** The Basilica of the Sacred heart at Notre Dame is beside to which structure? **ENTY**
- Q14:** How often is Notre Dame's the Juggler published? **NUM**
- Q15:** Where is the headquarters of the Congregation of the Holy Cross? **LOC**
- Q16:** What is the oldest structure at Notre Dame? **ENTY**
- Q17:** Which organization declared the First Year of Studies program at Notre Dame "outstanding"? **HUM**
- Q18:** The College of Science began to offer civil engineering courses beginning at what time at Notre Dame? **HUM**
- Q19:** In what year was the College of Engineering at Notre Dame formed? **NUM**
- Q20:** Which prize did Frederick Buechner create? **ENTY**
- Q21:** What was the amount of children murdered? **NUM**
- Q22:** Where was one employee killed? **HUM**
- Q23:** What happened in Chad? **DESC**
- Q24:** What did one of John II's replacements do in captivity? **ENTY**
- Q25:** Who threw the first touchdown pass of the game? **HUM**
- Q26:** Which player scored touchdowns running and receiving? **HUM**
- Q27:** What all field goals did Olindo Mare make? **ENTY**
- Q28:** Which team had a safety scored on them in the first half? **HUM**
- Q29:** What was the difference between the role of blacks and whites in the draft? **DESC**
- Q30:** What was burned last: city of Ryazan or suburbs of Moscow? **LOC**
-

Table 9: Qualitative examples of classified questions in target datasets.