Contrastive Domain Adaptation for Question Answering using Limited Text Corpora

Zhenrui Yue
Technical University of Munich
zhenrui.yue@tum.de

Bernhard Kratzwald
ETH AI Center
bkratzwald@ethz.ch

Stefan Feuerriegel
ETH Zurich
sfeuerriegel@ethz.ch

Abstract

Question generation has recently shown impressive results in customizing question answering (QA) systems to new domains. These approaches circumvent the need for manually annotated training data from the new domain and, instead, generate synthetic question-answer pairs that are used for training. However, existing methods for question generation rely on large amounts of synthetically generated datasets and costly computational resources, which render these techniques widely inaccessible when the text corpora is of limited size. This is problematic as many niche domains rely on small text corpora, which naturally restricts the amount of synthetic data that can be generated. In this paper, we propose a novel framework for domain adaptation called contrastive domain adaptation for QA (CAQA). Specifically, CAQA combines techniques from question generation and domain-invariant learning to answer out-of-domain questions in settings with limited text corpora. Here, we train a QA system on both source data and generated data from the target domain with a contrastive adaptation loss that is incorporated in the training objective. By combining techniques from question generation and domain-invariant learning, our model achieved considerable improvements compared to state-of-the-art baselines.

1 Introduction

Question answering (QA) systems generate answers to questions over text. Formally, such systems are nowadays trained end-to-end to predict answers conditional on an input question and a context paragraph (e.g., Seo et al., 2016; Chen et al., 2017a; Devlin et al., 2019). Therein, every QA sample is a 3-tuple consisting of a question, a context, and an answer. In this paper, we consider the subproblem of extractive QA, where the task is to extract answer spans from an unstructured context information for a given question as input. In extractive QA, both question and context are represented by running text, while the answer is defined by a start position and an end position in the context.

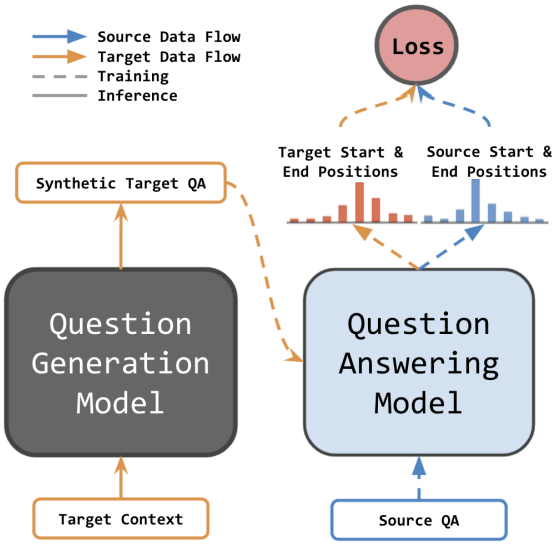

Figure 1: Overview of a common framework for QA domain adaptation. A question generation model is used to generate synthetic target data, which can be used for training the QA model with source data. The resulting QA model can answer target questions upon deployment.

An existing challenge for extractive QA systems is the distributional change between training data (source domain) and test data (target domain). If there is such a distribution change, the performance on test data is likely to be impaired. In practice, this issue occurs due to the fact that users, for instance, formulate text in highly diverse language or use QA for previously unseen domains (Hazen et al., 2019; Miller et al., 2020). As a result, out-of-domain (OOD) samples occur that diverge from the training corpora of QA systems (i.e., which can be traced back to the invariance of the training data) and, upon deployment, lead to a drastic drop in the accuracy of QA systems.

One solution to the above-mentioned challenge

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9575–9593
November 7–11, 2021.

of a domain shift is to generate synthetic data from the corpora of the target domain using models for question generations and then use the synthetic data during training (e.g., Lee et al., 2020; Shakeri et al., 2020). For this purpose, generative models have been adopted to produce synthetic data as surrogates from target domain, so that the QA system can be trained with both data from the source domain and synthetic data, which helps to achieve better results on the out-of-domain data distribution (Puri et al., 2020; Lee et al., 2020; Shakeri et al., 2020), see Figure 1 for an overview of such approach. Nevertheless, large quantities of synthetic data require intensive computational resources. Moreover, many niche domains rely upon limited text corpora. Their limited size puts barriers to the amount of synthetic data that can be generated and, as well shall see later, render the aforementioned approach for limited text corpora largely ineffective.

In computer vision, some works draw upon another approach for domain adaptation, namely discrepancy reduction of representations (Long et al., 2013; Tzeng et al., 2014; Long et al., 2015, 2017; Kang et al., 2019). Here, an adaptation loss or adversarial training approaches are often designed to learn domain-invariant features, so that the model can transfer learnt knowledge from the source domain to the target domain. However, the aforementioned approach for domain adaptation was designed for computer vision tasks, and, to the best of our knowledge, has not yet been tailored for QA.

In this paper, we develop a framework for answering out-of-domain questions in QA settings with limited text corpora. We refer to our proposed framework as **contrastive domain adaptation for question answering** (CAQA). CAQA combines question generation and contrastive domain adaptation to learn domain-invariant features, so that it can capture both domains and thus transfer knowledge to the target distribution. This is in contrast to existing question generation where synthetic data is solely used for joint training with the source data but without explicitly accounting for domain shifts, thus explaining why CAQA improves the performance in answering out-of-domain questions. For this, we propose a novel contrastive adaptation loss that is tailored to QA. The contrastive adaptation loss uses maximum mean discrepancy (MMD) to measure the discrepancy in the representation between source and target features, which is reduced while it simultaneously separates answer tokens for answer extraction.[1]

The main **contributions** of our work are:

1. We propose a novel framework for domain adaptation in QA called CAQA. To the best of our knowledge, this is the first use of contrastive approaches for learning domain-invariant features in QA systems.
2. Our CAQA framework is particularly effective for limited text corpora. In such settings, we show that CAQA can transfer knowledge to target domain without additional training cost.
3. We demonstrate that CAQA can effectively answer out-of-domain questions. CAQA outperforms the current state-of-the-art baselines for domain adaptation by a significant margin.

2 Related Work

The performance of extractive question answering systems (e.g., Chen et al., 2017b; Kratzwald et al., 2019; Zhang et al., 2020) is known to deteriorate when the training data (source domain) differs from the data used during testing (target domain) (Talmor and Berant, 2019). Approaches to adapt QA systems to a certain domain can be divided in (1) supervised approaches, where one has access to labeled data from the target domain (i.e., transfer learning; Kratzwald and Feuerriegel, 2019), or (2) unsupervised approaches, where no labeled information is accessible. The latter is our focus. Unsupervised approaches are primarily based on question generation techniques where one generates synthetic training data for the target domain.

Question generation (QG): Question generation (Rus et al., 2010) is the task of generating synthetic QA pairs from raw text data. Several approaches have been developed to generate synthetic questions in QA. Du et al. (2017) propose an end-to-end seq2seq encoder-decoder for the generation. Recently, question generation and answer generation are observed as dual tasks and combined in various ways. Tang et al. (2017) train both simultaneously; Golub et al. (2017) split the process in two consecutive stages; and Tang et al. (2018) use policy gradient to improve between-task learning.

Question generation is a common technique for domain adaptation in QA. Here, the generated questions are used to fine-tune QA systems to the new target domain (Dhingra et al., 2018). Oftentimes,

[1] The code from our CAQA framework is publicly available via https://github.com/Yueeeeeeee/CAQA

only a subset of generated questions is selected to increase the quality of the generated data. Common approaches are based on curriculum learning (Sachan and Xing, 2018); roundtrip consistency, where samples are selected when the predicted answers match the generated answer (Alberti et al., 2019); iterative refinement (Li et al., 2020); and conditional priors (Lee et al., 2020).

Unsupervised domain adaptation: A large body of work on unsupervised domain adaptation has been done in the area of computer vision, where the representation discrepancy between a labeled source dataset and an unlabeled target dataset is reduced (e.g., Tzeng et al., 2014; Saito et al., 2018; Long et al., 2015). Recent approaches are often based on adversarial learning, where one minimizes the distance between feature distributions in both both the source and target domain, while simultaneously minimizing the error in the labeled source domain (e.g., Long et al., 2017; Tzeng et al., 2017). Moreover, adversarial training is also applied to train generalized QA systems across domains to improve performance on the data distribution of the target domain (Lee et al., 2019).

Unlike adversarial approaches, contrastive methods (i.e., Hadsell et al., 2006) utilize a special loss that reduces the discrepancy of samples from the same class ('pulled together') and that increases the distances for samples from different classes ('pushed apart'). This is achieved by using either pair-wise distance metrics (Hadsell et al., 2006), or a triplet loss and clustering techniques (Schroff et al., 2015; Cheng et al., 2016). Recently, a contrastive adaptation network (CAN) has been shown to achieve state-of-the-art performance by using maximum mean discrepancy to build an objective function that maximizes inter-class distances and minimizes intra-class distances with the help of pseudo-labeling and iterative refinement (Kang et al., 2019). Yet, it is hitherto unclear how this technique can be used to improve domain adaptation in QA.

3 The CAQA Framework: Contrastive Domain Adaptation for QA

3.1 Setup

Input: Our framework is based on QA data under a distributional change. Thus let $\boldsymbol{\mathcal{D}}_s$ denote the source domain and $\boldsymbol{\mathcal{D}}_t$ the target domain. We expected a distributional change, that is, both domains are different (i.e., $\boldsymbol{\mathcal{D}}_s \neq \boldsymbol{\mathcal{D}}_t$). Formally, the input is given by:

- *Training data from source domain*: We are given labeled data from the source domain $\boldsymbol{X}_s$. Each sample $\boldsymbol{x}_s^{(i)} \in \boldsymbol{X}_s$ from the source domain $\boldsymbol{\mathcal{D}}_s$ comprises of a 3-tuple with a question $\boldsymbol{x}_{s,q}^{(i)}$, a context $\boldsymbol{x}_{s,c}^{(i)}$, and an answer $\boldsymbol{x}_{s,a}^{(i)}$.

- *Target contexts*: We have access to target domain data. Yet, of note, the data is unlabeled. That is, we have only access to the contexts. We further assume that the amount of target contexts is limited. Let $\boldsymbol{X}_t^{'}$ denote the unlabeled target data, where each sample $\boldsymbol{x}_t^{(i)} \in \boldsymbol{X}_t^{'}$ from the target domain $\boldsymbol{\mathcal{D}}_t$ consists of only a context $\boldsymbol{x}_{t,c}^{(i)}$.

Objective: Upon deployment, we aim at maximizing the performance of the QA system when answering questions from the target domain $\boldsymbol{\mathcal{D}}_t$, that is, minimizing the cross-entropy loss of the QA system $\boldsymbol{f}$ for $\boldsymbol{X}_t$ from the target domain $\boldsymbol{\mathcal{D}}_t$, i.e.,

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

However, actual question-answer pairs from the target domain are unknown until deployment. Furthermore, we expect that the available contexts are limited in size, which we refer to as limited text corpora. For instance, our experiments later involve only 5 QA pairs per context and, overall, 10k paragraphs as context.

Overview: The proposed CAQA framework has three main components (see Figure 2): (1) a **question generation model**, (2) a **QA model**, and (3) a **contrastive adaptation loss** for domain adaptation, as described in the following. We refer to the question generation model via $\boldsymbol{f}_{\text{gen}}$ and to the QA model via $\boldsymbol{f}$. The question generation model $\boldsymbol{f}_{\text{gen}}$ is used for generate synthetic QA data $\boldsymbol{X}_t = \boldsymbol{f}_{\text{gen}}(\boldsymbol{X}_t^{'})$. This yields additional QA pairs consisting of $\boldsymbol{x}_{t,q}^{(i)}$ and $\boldsymbol{x}_{t,a}^{(i)}$ for $\boldsymbol{x}_{t,c}^{(i)} \in \boldsymbol{X}_t^{'}$. Then, we use both source data $\boldsymbol{X}_s$ and synthetic data $\boldsymbol{X}_t$ to train the QA model via our proposed contrastive adaptation loss. The idea behind it is to help transfer knowledge to the target domain via discrepancy reduction and answer separation.

3.2 Question Generation

The question generation (QG) model QAGen-T5 is designed as follows. The QG model takes a

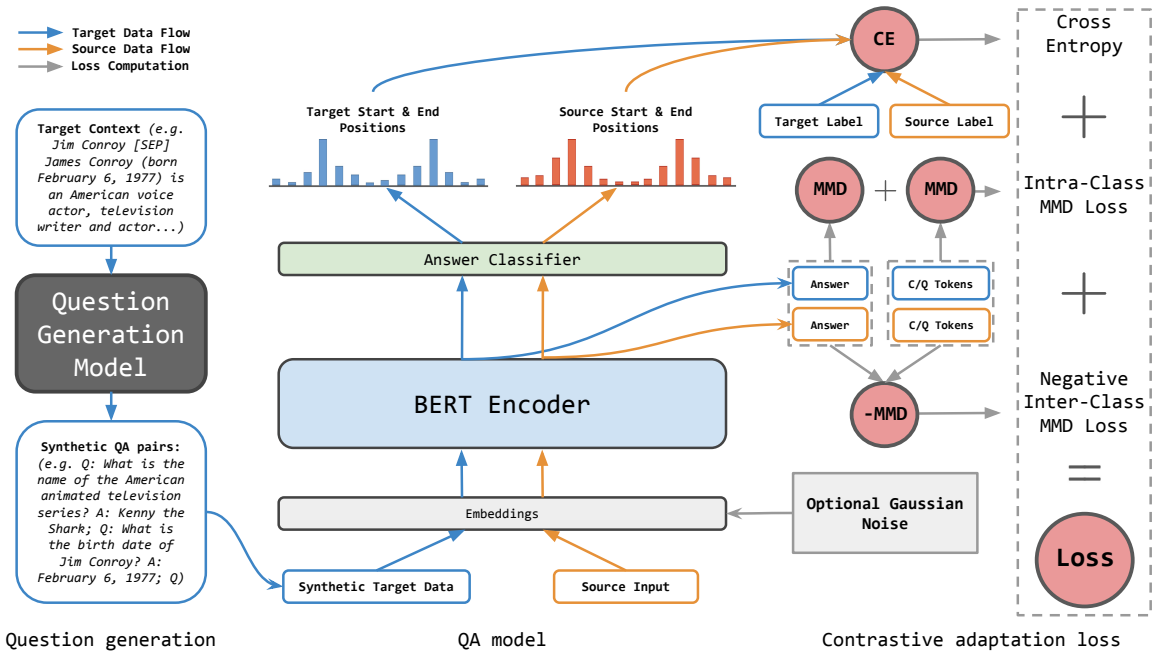

Figure 2: Overview of the proposed CAQA framework. A question generation model is used to generate synthetic data, which are then used for training the QA model using the contrastive adaptation loss. The resulting QA model is thus designed to handle QA data from the target domain upon deployment.

context as input and then involves two steps: (i) it first generates a question $\boldsymbol{x}_q$ based on context $\boldsymbol{x}_c$ in the target domain, and then (ii) a corresponding answer $\boldsymbol{x}_a$ conditioned on given $\boldsymbol{x}_c$ and $\boldsymbol{x}_q$. Using a two-step generation of questions and answers to build synthetic data is consistent with earlier literature on QG (e.g., Lee et al., 2020; Shakeri et al., 2020) and thus facilitates larger capacity while facilitating comparability. The maximum number k of synthetic QA data is determined later.

In our QG model, we utilize a text-to-text transfer transformer (T5) encoder-decoder transformer (Raffel et al., 2019). This transformer is able of performing multiple downstream tasks due to its the multi-task pretraining approach. This is beneficial in our case as we later use T5 transformers for conditional generation of two different outputs $\boldsymbol{x}_q$ and $\boldsymbol{x}_a$, respectively. Specifically, we use two T5 transformers for generating end-to-end (i) the question and (ii) the answer. We later refer to the combined variant for QG as 'QAGen-T5'.

Our QAGen-T5 is fed with the following input/output. The **input** to generate questions is only a context paragraph, and, therefore, we prepend the token `generate question:` in the beginning (which is then followed by the context paragraph). For answer generation, **input** using both a question and a context is specified via tokens `question:` and `context:`. The **output** varies across (i) question and (ii) answer. For (i), the output $\boldsymbol{x}_q$ are questions divided by the `[SEP]` token (e.g., input: '*generate question: python is a programming language...*' output: '*when was python released?*'). For (ii), the output $\boldsymbol{x}_a$ is an answer, for which we specify question and context information in the input by inserting tokens `question:` and `context:` (e.g., the input becomes '*question: when was python released? context: python is a programming language...*'). The output is the decoded answer.

QAGen-T5 is trained as follows. For (i) and (ii), we separately minimize the negative log likelihood of output sequences via

$$\mathcal{L}_{\text{qg}}(\boldsymbol{X}) = \sum_{i=1}^{|\boldsymbol{X}|} -\log p_{\theta_{\text{qg}}}(\boldsymbol{x}_q^{(i)} \mid \boldsymbol{x}_c^{(i)}), \quad (2)$$

$$\mathcal{L}_{\text{ag}}(\boldsymbol{X}) = \sum_{i=1}^{|\boldsymbol{X}|} -\log p_{\theta_{\text{ag}}}(\boldsymbol{x}_a^{(i)} \mid \boldsymbol{x}_c^{(i)}, \boldsymbol{x}_q^{(i)}), \quad (3)$$

where $\boldsymbol{x}_q^{(i)}$, $\boldsymbol{x}_a^{(i)}$, and $\boldsymbol{x}_c^{(i)}$ refer to question, answer, and context in the i-th sample of $\boldsymbol{X}$. Fine-tuning is done as follows. Both T5 models inside QAGen-T5 are fine-tuned on SQuAD separately. For selecting QA pairs, we draw upon LM-filtering (Shakeri et al., 2020) to select the best k QA pairs per

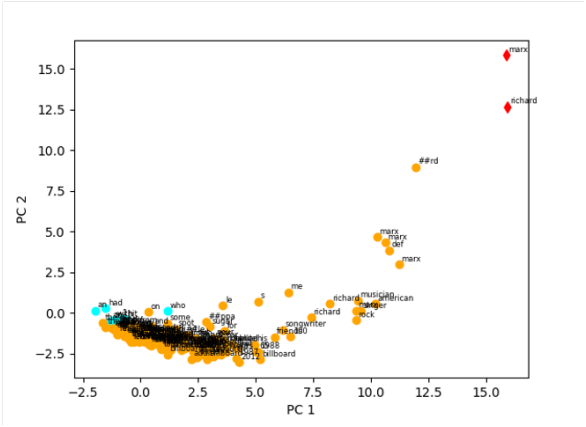

Figure 3: The 2D PCA visualization of BERT-encoder output on a SQuAD example. Answer tokens are in red and question tokens in cyan (all other tokens in orange).

context (e.g., $k = 5$ is selected later in our experiments). We compute the LM scores for the answer by multiplying the scores of each token over the output length. This ensures that only synthetic QA samples are generated where the combination of both question and answer has a high likelihood.

3.3 QA Model

Our QA model is set to BERT-QA (Devlin et al., 2019). BERT-QA consists of two components: the BERT-encoder and an answer classifier. The BERT-encoder extracts features from input tokens, while the answer classifier outputs two probability distributions for start and end positions to form answer spans based on the token features extracted by BERT-encoder. In our paper, the BERT-encoder is identical to the original BERT model and has an embedding component as well as transformer blocks (Devlin et al., 2019).

BERT-QA is trained using a cross-entropy loss $\mathcal{L}_{\text{ce}}$ to predict correct answer spans, yet additionally using our contrastive adaptation loss as described in the following.

3.4 Contrastive Adaptation Loss

We now introduce our contrastive adaptation loss, which we use for training the QA model. The idea in our proposed contrastive adaptation loss is two-fold: (i) We decrease the discrepancy among answer tokens and among other tokens, respectively ('intra-class'). This should thus encourage the model to learn domain-invariant features that are characteristic for both the source domain and the target domain. (ii) We enlarge the answer–context and answer–question discrepancy in feature representations ('inter-class').

Our approach is somewhat analogous yet different to contrastive domain adaptation in computer vision, where also the intra-class discrepancy is reduced, while the inter-class discrepancy is enlarged (Kang et al., 2019). In computer vision, the labels are clearly defined (e.g., object class), such labels are not available in QA. A natural way would be to see each pair of start/end location of an answer span as a separate class. Yet the corresponding space would be extremely large and would not represent specific semantic information. Instead, we build upon a different notion of classes: we treat all answer tokens as one class and the combined set of question and context tokens as a separate class. When we then reduce intra-class discrepancy and enlarge inter-class discrepancy, knowledge is transferred from source to target domain.

We focus on the discrepancy between answer and the other tokens since a trained QA model can well separate answer tokens in the source domain; see Figure 3. The plot shows a principle component analysis (PCA) visualizing the BERT-encoder output of a SQuAD example (van Aken et al., 2019). Answer tokens are well separated from all other tokens in this case, nevertheless, the same QA model can fail to perform answer separation in an unseen domain; see examples in Appendix D. Therefore, we apply contrastive adaptation on the token level and define classes by token types. Ideally, this reduces feature discrepancy between domains and, by separating answer tokens. Both should help improving the performance on out-of-domain data.

Discrepancy: In our contrastive adaptation loss, we measure the discrepancy among token classes using maximum mean discrepancy (MMD). MMD measures the distance between two data distributions based on the samples drawn from them (Gretton et al., 2012). Empirically, we compute the distance D between tokens X and Y represented by their mean embeddings in reproducing kernel Hilbert space $\mathcal{H}$, i.e.,

$$D = \sup_{f \in \mathcal{H}} \Big(\frac{1}{|\boldsymbol{X}|} \sum_{i=1}^{|\boldsymbol{X}|} f(\boldsymbol{x}_i) - \frac{1}{|\boldsymbol{Y}|} \sum_{i=1}^{|\boldsymbol{Y}|} f(\boldsymbol{y}_i) \Big). \quad (4)$$

MMD can be simplified by choosing a unit ball in $\mathcal{H}$, such that $D(\boldsymbol{X}, \boldsymbol{Y})^2 = \|\boldsymbol{\mu}_x - \boldsymbol{\mu}_y\|^2_{\mathcal{H}}$, where $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ represent the sample mean embeddings. Similar to (Long et al., 2015), we adopt Gaussian kernel with multiple bandwidths to estimate distances two samples, i.e., $k(\boldsymbol{x}_i, \boldsymbol{x}_j) =$

$\exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\gamma})$.

Contrastive adaptation loss: We define the contrastive adaptation loss of a mixed batch $\boldsymbol{X}$ with samples from both the source domain and the target domain as

$$\begin{aligned}\mathcal{L}_{\text{con}}(\boldsymbol{X}) = &\frac{1}{|\boldsymbol{X}|^2}\sum_{i=1}^{|\boldsymbol{X}|}\sum_{j=1}^{|\boldsymbol{X}|} k(\phi(\boldsymbol{x}_{\text{a}}^{(i)}), \phi(\boldsymbol{x}_{\text{a}}^{(j)})) \\ &+ \frac{1}{|\boldsymbol{X}|^2}\sum_{i=1}^{|\boldsymbol{X}|}\sum_{j=1}^{|\boldsymbol{X}|} k(\phi(\boldsymbol{x}_{\text{cq}}^{(i)}), \phi(\boldsymbol{x}_{\text{cq}}^{(j)})) \\ &- \frac{1}{|\boldsymbol{X}|^2}\sum_{i=1}^{|\boldsymbol{X}|}\sum_{j=1}^{|\boldsymbol{X}|} k(\phi(\boldsymbol{x}_{\text{a}}^{(i)}), \phi(\boldsymbol{x}_{\text{cq}}^{(j)})),\end{aligned} \tag{5}$$

where $\boldsymbol{x}_{\text{a}}$ is the mean vector of answer tokens, while $\boldsymbol{x}_{\text{cq}}$ is the mean vector of the context/question tokens. Further, ϕ is a feature extractor (i.e., the BERT-encoder). Equation (5) fulfills our objectives (i) and (ii) from above. The first two terms estimate the mean distance among all answers tokens and the other tokens, respectively, This should thus fulfill objective (i): to minimize the intra-class discrepancy. The last term maximizes the distance between answer and rest tokens (i.e., by taking the negative distance) and enables an easier answer extraction. This should thus fulfill objective (ii): to maximize the inter-class discrepancy.

Overall objective: We now combine both the cross-entropy loss from BERT-QA and the above contrastive adaptation loss into a single optimization objective for the QA model:

$$\mathcal{L}_{\text{qa}}(\boldsymbol{X}) = \mathcal{L}_{\text{ce}}(\boldsymbol{X}) + \beta\mathcal{L}_{\text{con}}(\boldsymbol{X}), \tag{6}$$

where $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss for the training QA model to predict correct answer spans. Here, β is hyperparameter that we choose empirically.

In our experiments, we sample mixed mini-batches and compute the overall loss to update the QA model. We encourage correct answer extraction by maximizing the representation distance between answer tokens and the other tokens. Additionally, we apply Gaussian noise at different scales σ on the token embeddings to learn a smooth and generalized feature space (Cai et al., 2019).

4 Experiment Setup

4.1 Datasets

In our experiments, we use SQuAD v1.1 as our **source domain** dataset (Rajpurkar et al., 2016). For the target domain we adopt four other datasets from MRQA (Fisch et al., 2019). This allows us to evaluate the performance in answering out-of-domain questions. For **target domain** datasets, only context paragraphs are accessible for question generation. In this paper, we use TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019), and SearchQA (Dunn et al., 2017). For more details, see Appendix A.1.

4.2 Baselines

We draw upon the following state-of-the-art baselines for question generation: Info-HCVAE (Lee et al., 2020), AQGen (Shakeri et al., 2020), and QA-Gen (Shakeri et al., 2020). These are used to generate synthetic QA data in order to train BERT-QA as the underlying QA model (i.e., the QA model is the same as in CAQA, the only difference is in how synthetic QA data is generated and how the QA model is trained). For more details, see Appendix A.2.

Additionally, we train BERT-QA (Devlin et al., 2019) on SQuAD and target datasets, respectively. This is to evaluate its base performance with zero knowledge of the target domain ('lower bound') and supervised training performance ('upper bound') on target datasets. We further report QAGen-T5 as an ablation study reflecting CAQA without the contrastive adaptation loss.

4.3 Training and Evaluation

Training:

We perform experiments based on limited text corpora: we only allow 5 QA pairs per context and 10k context paragraphs in total to be generated as the surrogate dataset. As such, no intensive computational resources are required for QA domain adaptation. First, we randomly select 10k context paragraphs and generate QA pairs with all mentioned generative models as synthetic data (abbreviated as '10k Syn'). Then, QA pairs are filtered using roundtrip consistency (baseline models) or LM-filtering (QAGen-T5), such that max. 5 QA pairs are preserved for each context. The final training data is then given by the combination of the generated target QA pairs and the SQuAD training set ('S + 10k Syn'). Based on this, we train BERT-QA on it and evaluate the model on target dev sets.

Evaluation: For evaluation, we adopt two metrics: exact match (EM) and F1 score (F1). We

Model	Training data	TriviaQA EM / F1	HotpotQA EM / F1	NaturalQ. EM / F1	SearchQA EM / F1
(I) Performance on target datasets w/o domain adaptation					
BERT-QA	SQuAD	50.84/60.48	43.57/61.09	45.14/59.35	19.66/27.90
(II) Performance on target datasets w/ supervised training					
BERT-QA	10k Target	55.86/62.14	49.30/66.23	55.26/68.19	62.48/69.39
BERT-QA	All Target	65.28/70.80	57.69/74.78	67.25/79.03	72.46/78.76
(III) Performance on target datasets w/ question generation					
Info-HCVAE	S + 10k Syn	45.66/55.28	39.47/55.60	37.12/51.17	17.41/24.24
AQGen	S + 10k Syn	51.41/60.60	44.79/59.99	40.78/54.02	34.03/42.08
QAGen	S + 10k Syn	50.52/59.79	45.67/60.88	44.13/57.84	29.59/36.63
QAGen-T5 (proposed)	S + 10k Syn	54.32/62.74	**46.50/62.03**	46.48/60.65	32.54/39.44
CAQA (proposed)	S + 10k Syn	**55.17/63.23**	46.37/61.57	**48.55/62.60**	**36.05/42.94**

Table 1: Main results comparing question-answering performance on out-of-domain data.

compute these metrics on target dev sets to evaluate the out-of-domain performance of the trained QA systems. For details, see Appendix A.3.

5 Results

5.1 Performance on Target Questions

Table 1 reports the main results. The table includes several baselines: (I) BERT-QA using only SQuAD as naïve baseline without domain adaptation ('lower bound'); (II) BERT-QA with supervised learning and thus access to the target data, which are otherwise not used ('upper bound'); and (III) several state-of-the-art baselines for question generation.

We make the following observations: (1) The naïve baseline is consistently outperformed by our proposed CAQA framework. Compared to the SQuAD baseline, CAQA leads to a performance improvement in EM of at least 2.80% and can go as high as 16.39%, and an improvement in F1 of at least 0.48% and up to 15.04%. (2) The naïve baseline provides a challenge for several question generation baselines from the literature, which are often inferior in our setting with limited text corpora. (3) The best-performing approach is our CAQA framework for three out of four datasets. For one dataset (HotpotQA), it is our QAGen-T5 variant without contrastive adaptation loss. However, the performance of CAQA is of similar magnitude and is clearly ranked second. (4) By comparing CAQA and QAGen-T5, we yield an ablation study quantifying the gain that should be attributed to using a contrastive adaptation loss. Here, we find distinctive performance improvements due to our contrastive adaptation loss for three out of four datasets. (5) Compared to the question generation baselines, our CAQA framework is superior. Compared to AQGen, the average improvements in EM and F1 are 3.78% and 3.41%, respectively, and, compared QAGen, the average improvements are 4.06% and 3.80%, respectively. (6) In the case of TriviaQA and HotpotQA, the performance of CAQA is close to that of supervised training results using 10k paragraphs from the target datasets. We discuss reasons reasons for performance variations across datasets in Section 6.

Altogether, the results suggest that the proposed combination of QAGen-T5 and contrastive adaptation loss is effective in improving the performance for out-of-domain data.

5.2 Sensitivity Analysis for Text Corpora Size

We now perform a sensitivity analysis studying how the performance varies across different text corpora sizes, that is, the number of context paragraphs generated. For this, we randomly select 10k, ..., 50k context paragraphs for training and then report two variants: (i) the QG performance using QAGen-T5 with varying context numbers and (ii) our CAQA with 10k context paragraphs. Here, results are reported for TriviaQA and NaturalQuestions[2].

For QAGen-T5, we see a comparatively large improvement when increasing the size from 10k to 20k context paragraphs. A small performance improvement among QAGen-T5 can be obtained when choosing 50k context paragraphs. In contrast to that, CAQA is superior, even when using only 10k context paragraphs. Put simply, it does so

[2] A sensitivity analysis varying the number of QA pairs (k) is reported using HotpotQA and SearchQA in Appendix B.2

much with much fewer samples and thus without additional costs due to extra computations. In sum, this demonstrates the effectiveness of CAQA for improving QA domain adaptation in settings with limited text corpora.

Model	Training data	TriviaQA EM / F1	NaturalQ. EM / F1
Performance w/o domain adaptation			
BERT-QA	SQuAD	50.84/60.48	45.14/59.35
Performance w/ question generation			
QAGen-T5	S + 10k Syn	54.32/62.74	46.48/60.65
QAGen-T5	S + 20k Syn	53.73/62.45	47.98/61.90
QAGen-T5	S + 30k Syn	54.75/63.12	47.76/61.93
QAGen-T5	S + 40k Syn	54.82/63.09	48.47/62.55
QAGen-T5	S + 50k Syn	54.90/63.11	48.23/62.71
CAQA	S + 10k Syn	**55.17/63.23**	**48.55/62.60**

Table 2: Sensitivity analysis across different corpora sizes. Top: QAGen-T5 w/o contrastive adaptation loss; below: CAQA with such loss.

5.3 Comparison: Training Baselines with Contrastive Adaptation Loss

We perform a sensitivity analysis examining whether the baselines models (Info-HCVAE, AQGen, and QAGen) can be improved when training them using our contrastive domain adaptation. For this, we repeat the above experiments with 10k synthetic samples (i.e., S + 10k Syn). The only difference is that we use our contrastive adaptation loss. The results are in Table 3. Here, a positive value means that the use of a contrastive adaptation loss results in a performance gain (since everything else is kept equal). Note that combining QAGen-T5 with our contrastive adaptation loss yields CAQA. Overall, we see that the performance of almost baselines can be improved due to our contrastive adaptation loss.

Model	TriviaQA EM / F1	HotpotQA EM / F1
Info-HCVAE	-0.33/-0.45	+0.03/+0.01
AQGen	+0.21/+0.26	**+0.86/+0.88**
QAGen	-0.09/-0.37	+0.42/+0.37
QAGen-T5 (=CAQA)	**+0.85/+0.49**	+0.70/+0.69
Model	**NaturalQ.** EM / F1	**SearchQA** EM / F1
Info-HCVAE	+0.66/+0.47	+1.01/+1.04
AQGen	+1.90/+1.85	-0.25/-0.28
QAGen	+1.23/+0.79	**+4.89/+5.50**
QAGen-T5 (=CAQA)	**+2.07/+1.95**	+3.51/+3.50

Table 3: Performance improvements (absolute) when training baselines with our proposed contrastive adaptation loss.

6 Discussion

We now discuss variations in the performance across models and datasets. For this, we also investigate synthetic data generated by CAQA manually (see Appendix C).

Why is the performance sometimes below the upper bound (i.e., supervised training)?

We see two explanations for the performance gap between supervised training and CAQA (as well as the other baselines). (i) Despite domain adaptation, some of the generated synthetic data cannot perfectly match the characteristics of the target domain but still reveal differences. We found this behavior, e.g., for NaturalQuestions. Here, the average length of the synthetic answers are all below 3, as compared to 4.35 the training set of NaturalQuestions. This may lead to a performance gap at test time. (ii) The generated QA pairs are comparatively homogeneous and lack the diversity of the target domain. To examine this, we manually inspected synthetic samples from CAQA (see Appendix C). We found that the generated QA pairs cannot fully capture the diversity that is otherwise common in question formulation. For example, almost all questions in the synthetic data start with '*What*', '*When*', and '*Who*'. In contrast, in NaturalQuestions, we find many questions that we perceived as more diverse or event more difficult. Examples are '*The court aquitted Moninder Singh Pandher of what crime?*' and '*Why does queen elizabeth sign her name elizabeth r?*'. Such behavior is particularly exacerbated for NaturalQuestions, which was intentionally designed to introduce more variety in question formulation, and, hence, our contrastive domain adaptation approach might implicitly learn some of the characteristics (as compared to the state-of-the-art baselines).

Why does the performance improvements vary across datasets?

The different improvements with contrastive adaptation can be further attributed to the target domain itself. When source and target datasets are similar, a model trained on the source dataset would naturally have better performance on the target dataset, but the improvements with contrastive adaptation can be limited due to the small domain variation. In TriviaQA and HotpotQA, the context paragraph originates – partially or completely – from Wikipedia and answer lengths are similar. In contrast, NaturalQuestions have differ-

ent text styles and sources including raw HTML like '<Table>', SearchQA context involves web articles and user contents, their average answer lengths are different, amounting to 1.89 and 4.43 respectively. Additionally, supervised training results using 10k HotpotQA and TriviaQA yield moderate improvements (5.02%, 5.73%), compared to 10.12% and 42.82% in NaturalQuestions and SearchQA. This also suggests that the difference between the previous datasets and SQuAD is comparatively small. Similar trends can be found in Table 3, where our contrastive adaptation on baseline models proves to be more effective in NaturalQuestions and SearchQA. Therefore, the discrepancy between source domain and target domain can be crucial for domain adaptation results according to our observations.

How does our contrastive adaptation loss affect the discrepancy among answer tokens?
To further examine how the contrastive adaptation loss improves the discrepancy among answers, we draw upon methods in (van Aken et al., 2019) and visualize the representations of the answer tokens using PCA (see Appendix D). Based on it, we empirically make the following observations. (i) In correct predictions, answer tokens are separated very well from questions and context tokens. (ii) In incorrect predictions, the correct answer is either not separated from the other tokens, or wrong tokens are separated and predicted as answers. In the latter case, such behavior is termed as overconfidence in out-of-domain data (cf. Kamath et al., 2020). In sum, contrastive adaptation helps in separating tokens that are likely to be answers, though sometimes incorrect tokens are identified as answers, thereby worsening the problem of overconfidence, which may explain the occasional decrease in performance.

7 Conclusion

This work contributes a novel framework for domain adaptation of QA systems in settings with limited text corpora. We develop CAQA in which we combine techniques from from question generation and domain-invariant learning to answer out-of-domain questions. Different from existing works in question answering, we achieve this by proposing a contrastive adaptation loss. Extensive experiments show that CAQA is superior to other state-of-the-art approaches by achieving a substantially better performance on out-of-domain data.

References

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

Guanyu Cai, Yuqin Wang, and Lianghua He. 2019. Learning smooth representation for unsupervised domain adaptation. *arXiv preprint arXiv:1905.10748*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017b. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. 2016. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344.

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

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

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

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.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 835–844, Copenhagen, Denmark. 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.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE.

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*.

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.

Bernhard Kratzwald, Anna Eigenmann, and Stefan Feuerriegel. 2019. RankQA: Neural question answering with answer re-ranking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6076–6085, Florence, Italy. Association for Computational Linguistics.

Bernhard Kratzwald and Stefan Feuerriegel. 2019. Putting question-answering systems into practice: Transfer learning for efficient domain customization. *ACM Trans. Manage. Inf. Syst.*, 9(4).

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.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online. Association for Computational Linguistics.

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.

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

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.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. 2013. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2200–2207.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217. PMLR.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*.

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition*, pages 3723–3732.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

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.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. 2018. Learning to collaborate for question answering and asking. 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 1564–1574, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

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.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension. *arXiv preprint arXiv:2001.09694*.

A Implementation Details

A.1 Datasets

In our experiments we used the following datasets, for target datasets, we adopt modified versions from MRQA (Fisch et al., 2019).

1. *SQuAD* (Rajpurkar et al., 2016) is a crowd-sourced dataset with context passages from Wikipedia and human-labeled question-answer pairs. The adopted SQuAD v1.1 training set has 18,885 context paragraphs and 86,588 QA pairs.
2. *TriviaQA* (Joshi et al., 2017) is a large-scale QA dataset that includes QA pairs and supporting facts for supervised training.
3. *HotpotQA* (Yang et al., 2018) provides multi-hop questions with human annotations (distractor paragraphs are excluded).
4. *NaturalQuestions* (Kwiatkowski et al., 2019) contains actual questions from users issued to the Google search engine. In the MRQA version, short answers are adopted, while long answers are used as context paragraphs.
5. *SearchQA* (Dunn et al., 2017) was constructed through a pipeline that starts from existing QA pairs and search for context information based on crawled online search results.

A.2 Baselines

1. **Info-HCVAE** (Lee et al., 2020) leverages a hierarchical variational autoencoder to encode context paragraph. Latent variables for questions and answers are sampled from the latent distributions conditional on a context. During training, mutual information between question and answer representations are maximized to provide consistent QA pairs at test time.
2. **AQGen** (Shakeri et al., 2020) is a generative baseline model for question generation. We modify the AQGen architecture similar to the QAGen2S in the original paper, namely using encoder-decoders to separately generate answer and question based on the input context information.
3. **QAGen** is based on QAGen2S in (Shakeri et al., 2020). Similar to AQGen, we exchange the task and generate possible questions end-to-end in the first step. Then, we generate answers based on both questions and contexts.

A.3 Implementation

Baselines: For Info-HCVAE, AQGen, and QAGen, we apply roundtrip filtering and limited the maximum QA pairs per context to 5 (Alberti et al., 2019). Info-HCVAE is trained for 20 epochs with the default settings (Lee et al., 2020). For AQGen and QAGen, we implement the models based on (Shakeri et al., 2020) and train for 10 epochs wit the learning rate of $1 \cdot 10^{-4}$ and the batch size of 16. The optimizer is set to AdamW without weight decay and a linear warmup (Loshchilov and Hutter, 2017), we validate the model with SQuAD dev set in training.

QAGen-T5: We apply LM-filtering as in (Shakeri et al., 2020) and select QA pairs with highest scores for each context paragraph. QAGen-T5 models are trained similarly to AQGen and QAGen, we separately keep the best QG and QA models according to validation performance on the SQuAD dev set.

QA model: We follow (Devlin et al., 2019; Kamath et al., 2020) and train BERT-QA with learning rate of $3 \cdot 10^{-5}$ for two epochs and with a batch size of 16. The AdamW optimizer is adopted and no linear warmup is used during training (Loshchilov and Hutter, 2017).

Hyperparameter search: In our experiments, we empirically search for hyperparameters β and σ in the contrastive adaptation loss through additional experiments. We experiment with different values of β in the range $[10^{-1}, 10^{-2}, 10^{-3}]$ and Gaussian noise $N(0, \sigma)$ applied on all token embeddings with standard deviation σ ranging from 0 to 10^{-2}. The best combination of β and σ as per the training set is then selected, these numbers can be found in Table 4.

Dataset	**Hyperparameter** β / σ	**CAQA Results** EM / F1
TriviaQA	0.001/0.01	55.17/63.23
HotpotQA	0.001/0.	46.37/61.57
NaturalQ.	0.01/0.01	48.55/62.60
SearchQA	0.001/0.01	36.05/42.94

Table 4: Hyperparameter selection for each target dataset in our main results.

All parameters that have not been mentioned explicitly above were used as reported in their original paper

B Additional Results

B.1 Comparison Limited Text Corpora vs. 'Large' Text Corpora

In this section, we compare our setting based on limited text corpora against the setting from the literature involving 'large' text corpora. Hence, we report the results from (a) the baseline models trained on SQuAD data (i.e., 'SQuAD' as in our main paper), (b) the baseline models using both SQuAD the 10k synthetic text corpora (i.e., 'S + 10k Syn' as in our main paper) and (c) the baseline models using both SQuAD the all provided text corpora, results are from (Lee et al., 2020). We also report (d), where ∼100k paragraphs are generated as synthetic QA data, which we take from (Shakeri et al., 2020). We refer to our implementation of (a) and (b) by marking the models using a '*'.

The results are in Table 5 (TriviaQA) and Table 6 (NaturalQuestions). As expected, the setting (b) is responsible for a lower performance due to the limited text corpora. The performance in (b), as compared to (c) and (d), is lower by around 5% to 10%. Importantly, our proposed CAQA still outperforms (b) by a considerable margin. Hence, despite using a considerable number sample of synthetic QA data, our CAQA is superior.

Model	**TriviaQA** EM / F1
Performance on Target Domain w/o Domain Adaptation	
(a) BERT-QA (Lee et al., 2020)	48.96/57.98
(a) BERT-QA*	50.84/60.48
Performance on Target Domain w/ Question Generation	
(c) HCVAE (Lee et al., 2020)	50.14/59.21
(b) HCVAE*	45.66/55.28
(b) QAGen*	50.52/59.79

Table 5: BERT-QA and question generation results of our implementation and original work(s) on TriviaQA.

Model	**NaturalQ.** EM / F1
Performance on Target Domain w/o Domain Adaptation	
(a) BERT-QA (Lee et al., 2020)	42.77/57.29
(a) BERT-QA (Shakeri et al., 2020)	44.66/58.94
(a) BERT-QA*	45.14/59.35
Performance on Target Domain w/ Question Generation	
(c) HCVAE (Lee et al., 2020)	48.19/62.21
(b) HCVAE*	37.12/51.17
(d) QAGen (Shakeri et al., 2020)	51.91/65.62
(b) QAGen*	44.13/57.84

Table 6: BERT-QA and question generation results of our implementation and original work(s) on NaturalQuestions

B.2 Sensitivity Analysis Varying the Number of QA Pairs per Context

We now perform a sensitivity analysis in which we vary the number of QA pairs per context (i.e., k). For this, we again adopt our CAQA framework (with both QAGen-T5 model and contrastive adaptation loss) using a combination of the SQuAD dataset and 10k context paragraphs. We vary the number of QA pairs for each context in the range $k = 1$, 3, 5, 7, and 9 QA pairs. The results are presented in Table 7. We note only some minor variation. The improvements tend to be larger when increasing the number of QA pairs per context in HotpotQA, while the results for SearchQA are less stable when increasing the number of of synthetic QA data.

Training data	**HotpotQA** EM / F1	**SearchQA** EM / F1
Performance w/o domain adaptation		
SQuAD	43.57/61.09	19.66/27.90
Performance w/ question generation		
S + 1QA/C	45.03/60.39	32.40/40.29
S + 3QA/C	45.92/61.13	34.96/42.47
S + 5QA/C	46.50/62.03	32.54/39.44
S + 7QA/C	47.45/62.83	**37.05/44.46**
S + 9QA/C	**47.57/63.26**	33.10/40.12
S + 5QA/C /w CAQA	46.37/61.57	36.05/42.94

Table 7: Sensitivity analysis varying the number of QA pairs per context (k).

C Qualitative Analysis of Synthetic Data Samples

We present qualitative examples of generated synthetic QA data using the proposed CAQA framework with our QAGen-T5 model. For this, two context paragraphs and five QA pairs for each paragraph are presented in the following; see Tables 8 to 11.

Table 8: Synthetic TriviaQA samples generated by QAGen-T5.

Synthetic TriviaQA samples
Context: UEFA European Championship Top 10 Midfielders - Football BibleUEFA European Championship Top 10 Midfielders UEFA European Championship Top 10 Midfielders Published on Country: Portugal Years Active: 1990 to 2008 Rui Costa represented Portugal in three UEFA European Championships . He was a good provider for other players to score goals but still scored a good number himself. At his time, the national side had a steady performance making it to the quarter finals in 1996, then the semifinals at the Euro 2000 and eventually the Euro 2004 final. On the progress to the final, he helped the Portugal team to eliminate England at the quarter finals through a remarkable strike which secured a 2-1 win.....
QA 1: What country did Rui Costa represent? Portugal
QA 2: In what year did Iniesta win UEFA Team of the Tournament? 1988
QA 3: What team did he play for in the European Football Championships? Portugal
QA 4: Who is the second highest assist-getter in La Liga? Luis Figo
QA 5: Who was a good provider for other players to score goals? Rui Costa
Context: Spiers on Sport: the unjust sacking of Kenny Shiels (From ...Spiers on Sport: the unjust sacking of Kenny Shiels (From HeraldScotland) / Spiers on Sport , Graham Spiers When a manager wins one of only four trophies collected by a football club in 80 years, there has to be a degree of respect shown towards him, right? When he also works slavishly on all aspects of a club due to staffing limitations - training, recruiting, video-editing, youth development etc - wouldn't that admiration for him grow even greater? Loading article content Kenny Shiels, sacked by Kilmarnock, is by no means perfect. But he has been a pretty good manager at Rugby Park, whose dismissal is hard to fathom.....
QA 1: Where did most of Shiels' felonies occur? Rugby Park
QA 2: What club did he manage? Kilmarnock
QA 3: Who is the chairman of the rugby club? Michael Johnston
QA 4: What was the name of the team that he managed? Kilmarnock
QA 5: Who was sacked by Kilmarnock? Kenny Shiels

Table 9: Synthetic HotpotQA samples generated by CAQA.

Synthetic HotpotQA samples
Context: Cascade Range [SEP] The Cascade Range or Cascades is a major mountain range of western North America, extending from southern British Columbia through Washington and Oregon to Northern California. It includes both non-volcanic mountains, such as the North Cascades, and the notable volcanoes known as the High Cascades. The small part of the range in British Columbia is referred to as the Canadian Cascades or, locally, as the Cascade Mountains. The latter term is also sometimes used by Washington residents to refer to the Washington section of the Cascades in addition to North Cascades, the more usual U.S. term, as in North Cascades National Park. The highest peak in the range is Mount Rainier in Washington at 14411 ft.....
QA 1: What is one of Oregon's most popular outdoor recreation sites? Lake of the Woods
QA 2: Who named the island? Oliver C. Applegate
QA 3: What is the name of the lake in Oregon? Lake of the Woods
QA 4: What is the name of the unincorporated community located on the east shore of the lake? Lake of the Woods
QA 5: What is another name for the Cascade Range? Cascades
Context: Jim Conroy [SEP] James Conroy (born February 6, 1977) is an American voice actor, television writer and actor. He is known for appearing on television shows, such as "Celebrity Deathmatch", "Kenny the Shark" and "Fetch with Ruff Ruffman", radio commercials and video games. He worked for companies such as WGBH, The Walt Disney Company and Discovery Channel. [PAR] [TLE] Kenny the Shark [SEP] Kenny the Shark is an American animated television series produced by Discovery Kids. The show premiered on NBC's Discovery Kids on NBC from November 1, 2003 and ended February 18, 2006 with two seasons and 26 episodes in total having aired.....
QA 1: How many episodes did the show have? 26
QA 2: What is the birth date of Jim Conroy? February 6, 1977
QA 3: What is the name of the American animated television series? Kenny the Shark
QA 4: Who produces Kenny the Shark? Discovery Kids
QA 5: What is Jim Conroy's birth date? February 6, 1977

Table 10: Synthetic NaturalQuestions samples generated by CAQA.

Synthetic NaturalQuestion samples
Context: <Table><Tr><Th>Rank </Th><Th>Chg </Th><Th>Channel name </Th><Th>Network </Th><Th>Primary language (s) </Th><Th>Subscribers (millions) </Th><Th>Content category </Th></Tr><Tr><Td>1 . </Td><Td></Td><Td>PewDiePie </Td><Td>N / A </Td><Td>English </Td><Td>64 </Td><Td>Comedy </Td></Tr><Tr><Td>2 . </Td><Td></Td><Td>T - Series </Td><Td>T - Series </Td><Td>Hindi </Td><Td>53 </Td><Td>Music </Td></Tr><Tr><Td>3 . </Td><Td></Td><Td>Justin Bieber </Td><Td>N / A </Td><Td>English </Td><Td>40 </Td><Td>Entertainment </Td></Tr><Tr><Td>4 . </Td><Td></Td><Td>Canal KondZilla.....
QA 1: What language is used in the Disney films? Hindi
QA 2: What company is Sony Pictures? Sony Pictures
QA 3: What is the name of the band that released the single "One Direction"? One Direction
QA 4: What is the name of the band that is known for its trap music? PewDiePie
QA 5: What is Katy Perry's real name? Katy Perry
Context: Red blood cell distribution width (RDW or RDW - CV or RCDW and RDW - SD) is a measure of the range of variation of red blood cell (RBC) volume that is reported as part of a standard complete blood count . Usually red blood cells are a standard size of about 6 - 8 \u03bcm in diameter . Certain disorders , however , cause a significant variation in cell size . Higher RDW values indicate greater variation in size . Normal reference range of RDW - CV in human red blood cells is 11.5 - 14.5 % . If anemia is observed , RDW test results are often used together with mean corpuscular volume (MCV) results to determine the possible causes of the anemia . It is mainly used to differentiate an anemia of mixed causes from an anemia of a single cause.....
QA 1: What do higher RDW values indicate? Greater variation in size
QA 2: What is the measure of the range of variation of red blood cell volume? Red blood cell distribution width
QA 3: What can cause a significant variation in cell size? Certain disorders
QA 4: What is the normal reference range of RDW - CV in human red blood cells? 11.5 - 14.5 %
QA 5: What is the main purpose of the test? To differentiate an anemia of mixed causes from an anemia of a

Table 11: Synthetic SearchQA samples generated by CAQA.

Synthetic SearchQA samples
Context: A white elephant - Idioms by The Free Dictionary [PAR] Definition of a white elephant in the Idioms Dictionary. a white elephant phrase. What does a white elephant expression mean? Definitions by the largest Idiom... [DOC] [TLE] Can an elephant stand up after laying down? [Archive] - Straight ... [PAR] We often receive e-mails from avid EleCam viewer saying, "There are elephants lying down in the pasture. They have been there a long time. [DOC] [TLE] Elephants sleep in zoo and circus [PAR] That is one of the reasons why elephants do not sleep much, and then only with ... The first elephant starts to lie down on its side towards 11 o'clock at night. [DOC] [TLE] Elephant Who Gives Rides All Day Can't Even Lie Down To Rest.....
QA 1: What can an elephant do after lying down? Stand up
QA 2: What do I struggle with? People who lie
QA 3: What is the name of the elephant who gives rides all day? Elephant Who Gives Rides All Day Can't Even Lie Down
QA 4: What is the official website of South African National Parks? SANParks
QA 5: What is more concerning to me than lies? Misbehavior
Context: jeopardy/1333_Qs.txt at master jedoublen/jeopardy GitHub [PAR] Number: 2. ANIMAL SONGS | British singer Robyn Hitchcock is known for his tunes about these animals, including "Bass" & "Aquarium" | Fish. right: Matt. Wrong:. [DOC] [TLE] Robyn Hitchcock - Wikipedia [PAR] Robyn Rowan Hitchcock (born 3 March 1953) is an English singer-songwriter and guitarist. While primarily a vocalist and guitarist, he also plays harmonica, piano, and bass guitar. ... Hitchcock's lyrics tend to include surrealism, comedic elements, ... Hitchcock released his solo debut, Black Snake Diamond Rle in 1981,... [DOC] [TLE] Positive Vibrations: Softcore - fegMANIA! [PAR] An except from Positive Vibrations' complete guide to the songs of Robyn Hitchcock.....
QA 1: What is the dance music of northeastern Argentina known as? Chaman
QA 2: What was Hitchcock's solo debut called? Black Snake Diamond Rle
QA 3: When did Hitchcock release his solo debut? 1981
QA 4: What is the name of the book that contains a complete guide to the songs of Robyn Hitchcock? Positive Vibrations: Softcore - fegMA
QA 5: What was the name of the singer who performed on The House List? Robyn Hitchcock

D PCA Visualization of Data

We visualize the BERT-QA output for the synthetic QA data generated by our QAGen-T5 model. Here, BERT-QA models are trained with contrastive adaptation loss on all target datasets separately. The results are shown for TriviaQA (Figure 4), HotpotQA (Figure 5), NaturalQuestions (Figure 6), and SearchQA (Figure 7). Answer tokens are in red diamond shapes, question tokens in cyan circles, while all other tokens are represented in orange circles.

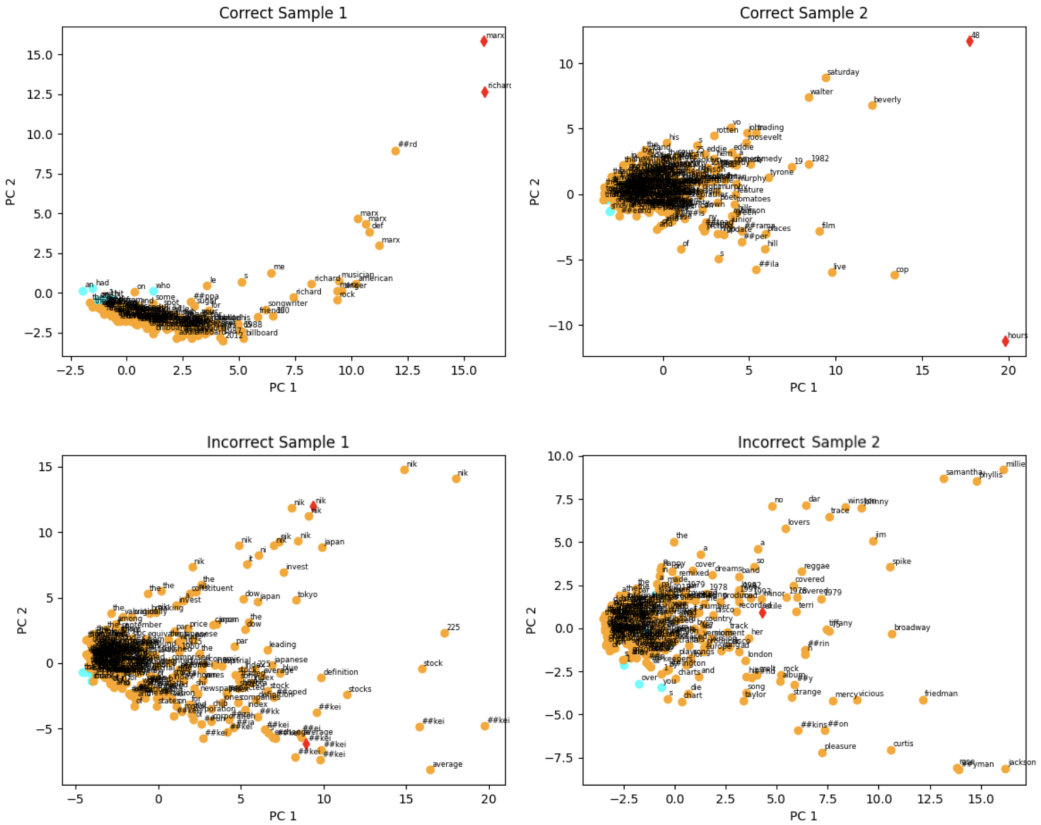

Figure 4: Visualization of BERT-encoder output on TriviaQA w/ contrastive adaptation loss.

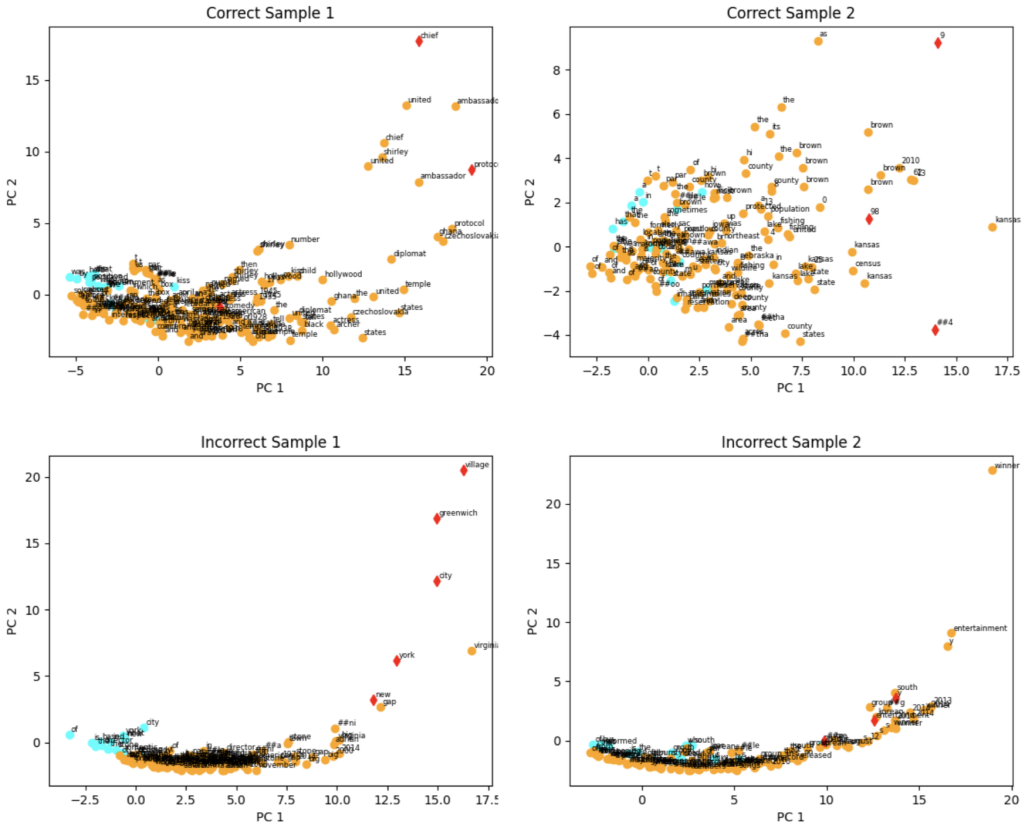

Figure 5: PCA visualization of BERT-encoder output on HotpotQA w/ contrastive adaptation loss.

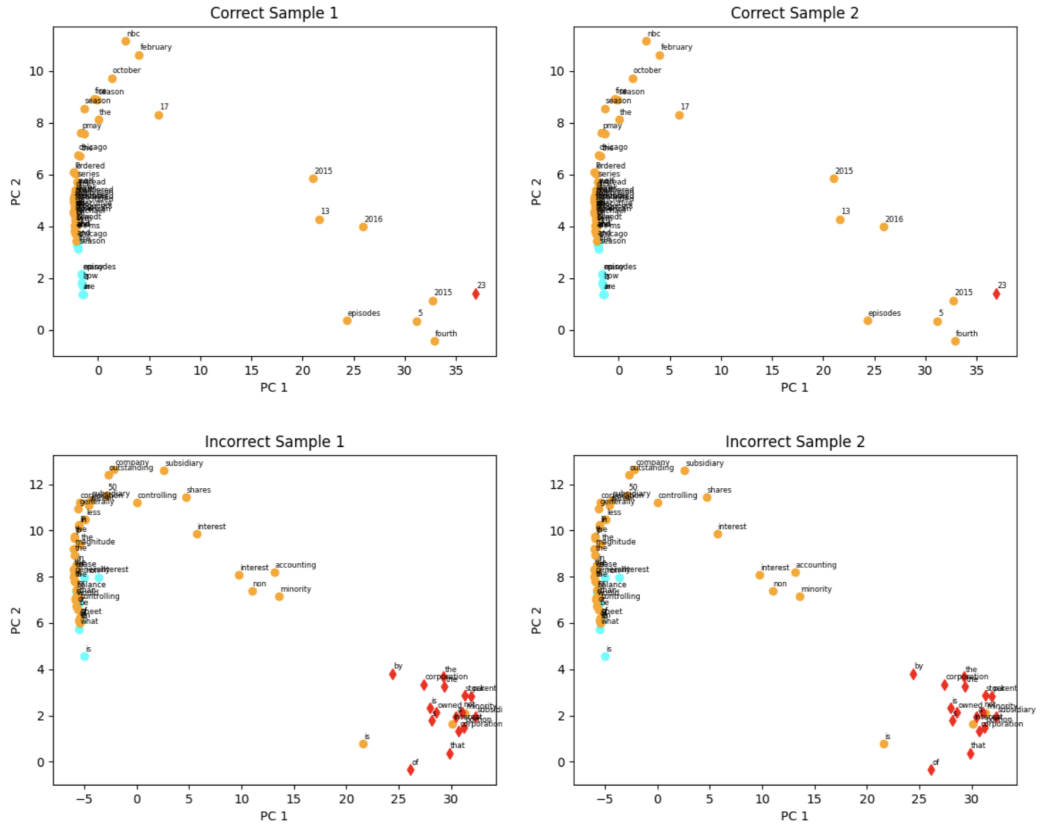

Figure 6: PCA visualization of BERT-encoder output on NaturalQuestions w/ contrastive adaptation loss.

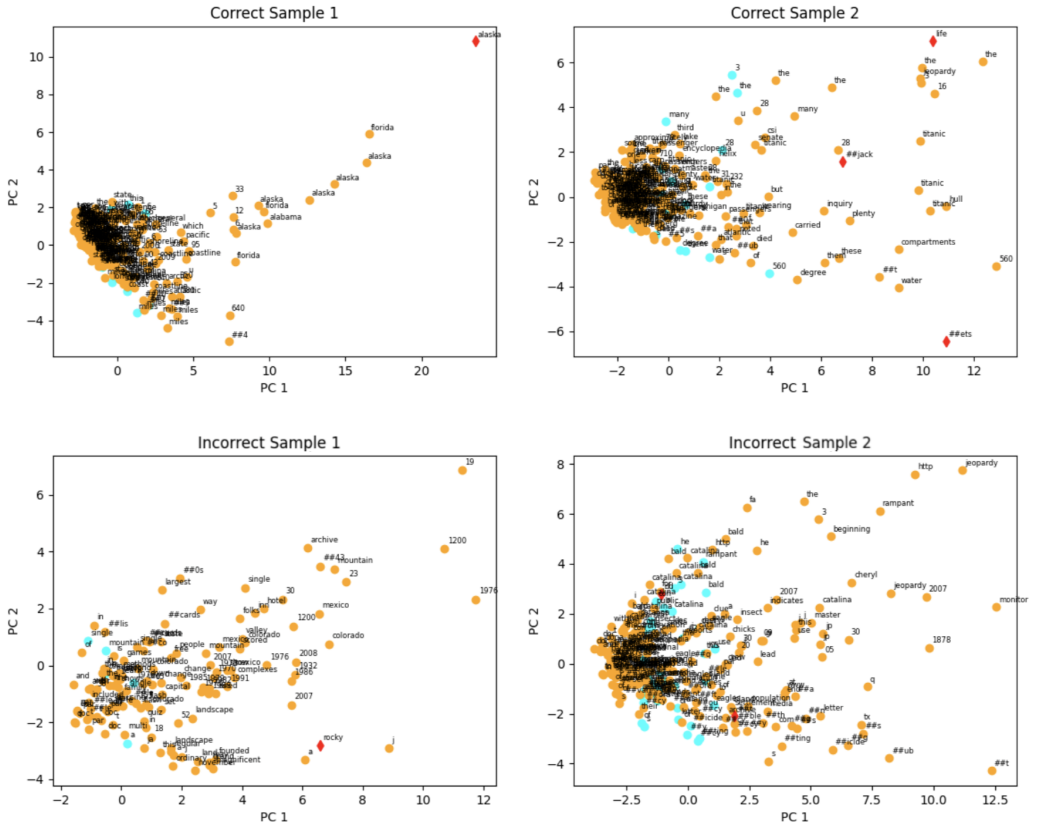

Figure 7: PCA visualization of BERT-encoder output on SearchQA w/ contrastive adaptation loss.