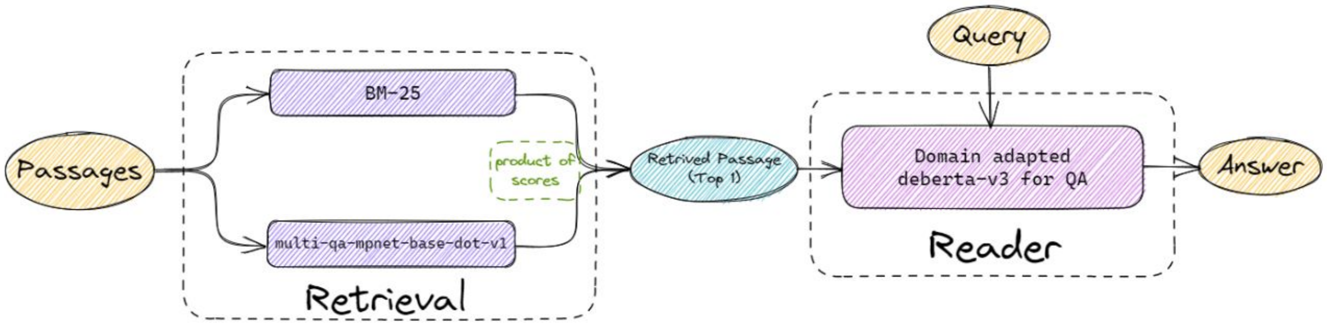
Improving Domain Specific Question Answering

> Primary Team ID: 17
> Secondary Team ID: 34

Abstract

Question Answering (QA) has been a pursuit of various product based firms to ease human-machine interaction. It can be viewed as a form of human-machine interaction to retrieve information from data (knowledge base) using natural language queries. We start off by preparing the retriever by combining lexical and semantic approaches. DeBERTa-V3-Base is selected as an efficient reader model to answer queries once the context is provided. After the eval set is provided, we adapt our reader to the eval domain using Contrastive Domain Adaptation. Code for the experiments can be accessed here.



Paragraph Retrieval

The basic objective of the system is to provide relevant information to users in response to their information needs. Thus, the most fundamental problem is to estimate the degree of relevance between a query `q` and a document `d`. The evaluation task provides us with question and theme pairs which allows us to create *theme-wise indices* which improve the latency and accuracy of the retriever system.

Lexical

BM25 (Best Match 25) is a bag of words retriever that ranks documents based on query terms appearing in each document. It does not take into consideration proximity of words. It is based on the probabilistic retrieval framework. Although BM-25 doesn't take the semantic meaning of words into consideration, on datasets like SQuAD and SQuAD V2, *where context and questions have high lexical similarity*, lexical approaches like BM25 work well. Another advantage of BM25 is that it is *more robust to out-of-domain data* as compared to semantic approaches as it doesn't need to understand the semantic meaning of the words.

Semantic

Semantic search seeks to improve search accuracy by understanding the content of the search query. In contrast to finding documents based on lexical matches, semantic search can also find synonyms. Semantic Search improves retriever performance as ~43% of the SQuAD queries vary lexically to the answer sentence.

The `multi-qa-mpnet-base-dot-v1` sentence-transformer model maps sentences and paragraphs to a 768-dimensional dense vector space and is *designed for semantic search*. It has been trained on 215M (question and answer) pairs from multiple datasets such as WikiAnswers, MS MARCO, TriviaQA, SQuAD 2.0 etc. *making it robust against domain shifts*. The model is trained using a self-supervised contrastive learning objective: given a sentence from the pair, the model should predict which out of a set of randomly sampled other sentences, was actually paired with it in our dataset.

Lexical + Semantic

The final pipeline *combines the lexical and semantic retrieval models* to best perform the paragraph retrieval task. Retrieval results from BM25 (lexical) and SBERT `multi-qa-mpnet-base-dot-v1` (semantic) are combined using Lexical Enhanced Dense Retrieval (LEDR) method proposed by LaPraDoR [1]. LEDR is a method to enhance the performance of dense retrievers by combining its results with BM25 to make it more robust to unseen data. During inference, we multiply the score of BM25 with the similarity score of LaPraDoR. We took Top-5 paragraph retrieval scores from both the models multiplied and then based on the final score Top-1 paragraph is selected and fed to the Question-Answering model.

What did not work

1. **Domain Adaptation**: Many pre-trained retrieval models have been observed diminishing advantages over term-based retrieval models like BM25 in various benchmarks if they are not fine-tuned with adequate labels. We find that current *domain adaptation techniques fail to improve paragraph retrieval task* for SQuAD queries.
2. **Generative Pseudo Labeling**: GPL [2] combines a query generator with pseudo labelling from a cross-encoder. It tries to improve on the significant shortcoming of dense retrievers, being worse performance in lexical queries; requiring large amounts of training data; and susceptibility to domain shifts. *Experiments show a slight increase in the top-1 retrieval scores but a decrease in the top-5 scores.*

Evaluation

All the evaluations below have been performed on SQuAD V2 dev set with the settings mentioned in the problem statement (i.e. theme is provided alongside the query, which restricts the retrieval search to the theme passages)

Technique	Top-1 Retrieval Accuracy	Time per Query (Colab CPU)
Lexical: BM25	58.45	2ms
Semantic: MPNet	63.57	110ms
Lexical + Semantic (BM25 + MPNet)	70.67	115ms

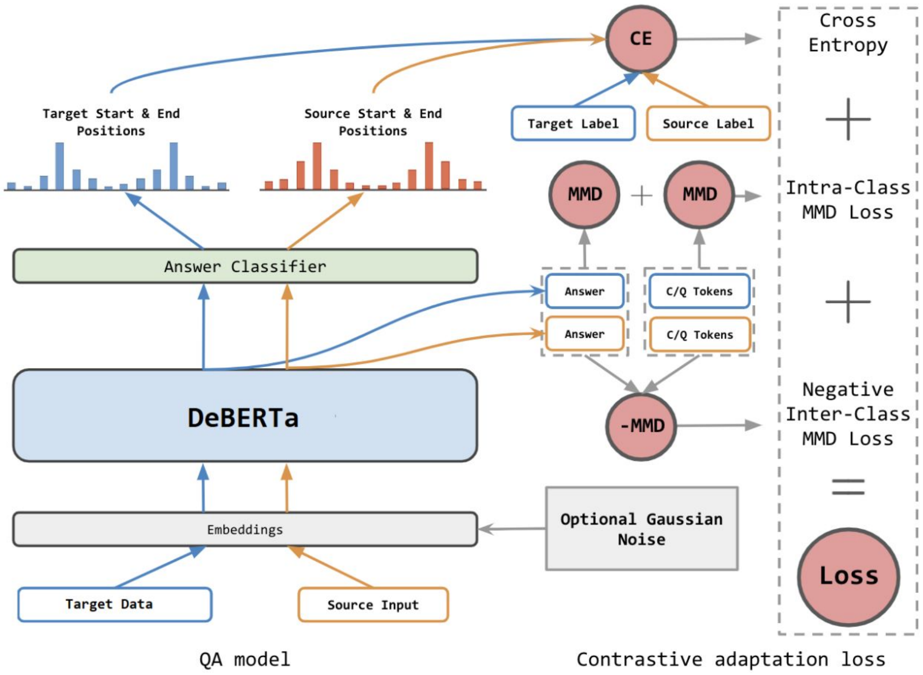
Question Answering

Transformer based reader models are the go to approach for question answering since pretrained models can be finetuned on the required domain for accurate QA. Many pre-trained language models have proven to be incredibly effective at the task of extractive question answering. However, generalizability remains as a challenge for the majority of these models. Transformer models overcome this challenge as they can be adapted to any required domain.

DeBERTa

DeBERTa v3 is used as the QA Reader Model in our pipeline. It is a significant improvement upon BERT and RoBERTa, which is achieved mainly by its Disentangled attention mechanism wherein each word is represented using two vectors that encode its content and position, respectively and hence eventually suits well to the MRC task of comprehending the text while also considering the position of the relevant answer text.

Domain Adaptation



For domain adaptation given source domain data and a target domain data we choose CAQA [3] for domain adaptation.
A contrastive adaptation loss-based training where the maximum mean discrepancy (MMD) score is optimized to minimize the **intra-class discrepancies** and to maximize the **inter-class discrepancies**. We do not use Synthetic Generated questions for domain adaptation since in our experiments we observe empirically that when using SQuADv2 dev set as the target domain, generated questions hurt the performace when perfoming ablations. Using just the contrastive losses helps with getting better results.

What did not work

Using synthetic questions on the target domain data has been the go to approach in the domain adaptation literature but in our preliminary evaluations this did not help our model performance. We provide the results for the experiments in a table in the evaluation section below.
We hypothesize that the reason for this observation could be the small domain gap between the source and target domain and the number of synthetic examples being very few.

Evaluation

All the evaluations have been performed on the SQuAD V2 dev set. For evaluating the performance of the reader models, we assume the knowledge of the context corresponding to the query.

Technique	F1	Exact Match
DeBERTa-V3-Base	87.41	83.83
DeBERTa + CAQA (using sample questions)	88.93	85.07
DeBERTa + CAQA (using generated questions)	86.12	82.68

Results

Finally, we pick the best techniques of each stage to finish the 2 stage retrieval-reader pipeline. A lexical + semantic strategy is first used to select the possible context that can answer the query. Then the context-query pair is passed through the DeBERTa reader domain adapted on the eval set.

Evaluating the above pipeline on SQuAD V2 dev set, with the settings mentioned in the problem statement gave us the following results:

DevRev metric	Time per Query
79.03	1005ms

Literature Review

The SQuAD Dataset

The original SQuAD dataset [4] is a reading comprehension dataset consisting of 100,000+ questions posed by crowd workers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage.

1. *Diverse Data*: The top 10000 articles of English Wikipedia were used, from which they sampled 536 articles uniformly at random.
2. *Reasoning required to answer questions*: The data has a mix of questions with lexical similarity and syntactical variation to the passages.

SQuAD V2

SQuAD 2.0 [5] combines existing SQuAD data with over *50,000 unanswerable questions written adversarially* by crowd workers to look similar to answerable ones. Crowd workers crafted these questions so that (1) they are relevant to the paragraph, and (2) the paragraph contains a plausible answer—something of the same type as what the question asks for. For both computer systems and humans, roughly half of all wrong answers on unanswerable questions exactly matched the plausible answers. This suggests that the plausible answers do indeed serve as effective distractors.

Retriever

Classic IR: Lexical

TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a measure that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc.) in a document amongst a collection of documents (a corpus). It can be used for *information retrieval by using a cosine similarity* on the vectors obtained using the TF-IDF vectorizer. It considers the importance of the words due to how it weighs them, but it *cannot necessarily derive the contexts of the words*.

BM25

BM (Best Match) 25 is a probabilistic retrieval framework that extends the idea of TF-IDF and improves some drawbacks of TF-IDF which concern term saturation and document length. It is a family of scoring functions with slightly different components and parameters.

Neural IR: Semantic

DPR

Dense Passage Retriever (DPR) [6] uses a dense encoder to index all the passages in a low-dimensional and continuous space. At run-time, a different encoder maps the input question to a vector and retrieves the top-k passages relevant to the input question using the dot product of their vectors. DPR *outperforms BM25 for open-domain QA tasks*. It also verified that in the context of open-domain QA, *a higher retrieval precision indeed translates to a higher end-to-end QA accuracy*.

We found that for domain-specific QA for SQuAD V2.0, BM25 performs better than DPR due to the domain-specific nature of the task and the lexical similarity of the SQuAD question context pairs.

SBERT

SBERT is a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. SBERT can be adapted to a specific task.It enables BERT to be used for certain new tasks, which up-to-now were not applicable for BERT. These tasks include `large-scale semantic similarity comparison`, `clustering`, and `information retrieval via semantic search`. It sets new state-of-the-art performance on a challenging argument similarity datasets.

Reader

BERT-Based Models

1. **BERT** (Bidirectional Encoder Representations from Transformers) [7] is a pre-trained model that achieved state-of-the-art performance on a variety of Natural Language Tasks. BERT uses Masked Language Model (MLM) which improves upon the earlier unidirectional language models. For the SQuAD V2.0 dataset, BERT treats questions that do not have an answer as having an answer span with start and end at the [CLS] token.
2. **SpanBERT** [8] builds upon BERT by masking contiguous random spans, rather than random tokens, and training to predict the entire content of the masked span, without relying on the individual token representations within it. SpanBERT outperforms BERT with substantial gains on span selection tasks such as question answering
3. Robustly Optimized BERT (**RoBERTa**) [9] improves upon BERT by *training the model longer, with bigger batches, over more data*. A novel dataset *CC-News* was used and showed that using more data improves performance on downstream tasks. The model uses Byte-Pair Encoding (BPE) for text encoding which is a hybrid between character and word-level representations.
4. **ELECTRA** [10] uses pre-training text encoders as discriminators rather than generators. Compared to BERT, it uses a more sample-efficient pre-training task called replaced token detection. The key idea (similar to GANs) is to train a text encoder to distinguish input tokens from high-quality negative samples produced by a small generator network.

DeBERTa

DeBERTa V1

DeBERTa [11] gets its name from the two novel techniques it introduces, through which it claims to improve over BERT and RoBERTa, *disentangled attention* and *enhanced mask decoder*

- *Disentangled attention* for position information and context information:
  - Disentangled attention mechanism, where each word is represented using two vectors that encode its content and position, respectively, and the attention weights among words are computed using disentangled matrices on their contents and relative positions, respectively.
- *Enhanced mask decoder*:
  - DeBERTa uses an enhanced mask decoder to improve MLM by adding absolute position information of the context words at the MLM decoding layer before softmax operation.

DeBERTa V3

Following ELECTRA-style training, MLM objective in DeBERTa is replaced with Replaced Token Detection [12] where the model is trained as a discriminator to predict whether a token in the corrupted input is either original or replaced by a generator. The MLM used for training the generator tries to pull the tokens that are semantically similar close to each other while the RTD of the discriminator tries to discriminate semantically similar tokens and pull their embeddings as far as possible to optimize the binary classification accuracy.

Gradient-Disentangled Embedding Sharing method: (Disentangled gradients but shared embeddings in generator discriminator) so as to improve both pre-training efficiency and the quality of the pre-trained models. In this the generator shares its embeddings with the discriminator but stops the gradients in the discriminator from back-propagating to the generator embeddings.

Domain Adaptation

The goal of domain adaptation is to adapt to a target domain given a model initially trained on a source domain, given a domain gap between the two domains such that the task remains the same but the data distributions for the source and target domains are different.

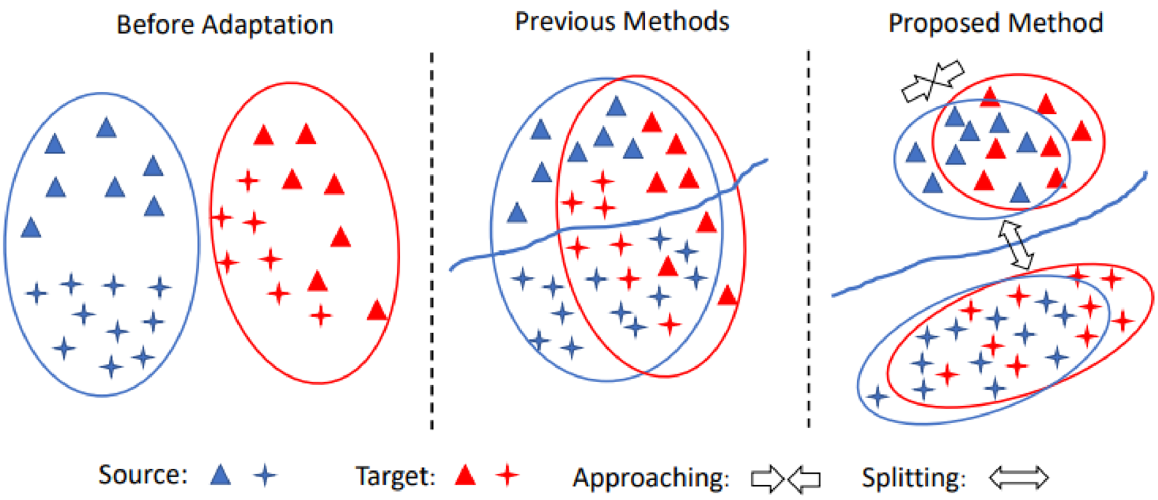
The distributions being different leads to poor performance in the target domain when trained only on the source domain data.

Domain adaptation can be formulated as a task of learning **domain invariant features** such that the model generalizes well on out-of-domain data rather than learning features specific to a domain.

There exist approaches to learning domain invariant features such as adversarial and contrastive learning [cite papers here].

Adversarial training approaches are often designed to learn domain-invariant features so that the model can transfer learned knowledge from the source domain to the target domain. Minimizes the distance between feature distributions in both the source and target domain, while simultaneously minimizing the error in the labeled source 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.



Unlike adversarial approaches, contrastive methods utilize a special loss that reduces the discrepancy of samples from the same class (pulled together) and increases the distances for samples from different classes (pushed apart) rather than reducing the discrepancy in feature representations of source and target domains.

Question Generation

Info-HCVAE

The QAG task is modeled as learning the conditional joint distribution of the question and answer given the context, p(x, y|c), from which question answering pairs can be sampled [13].
We discarded this approach as a potential Question Generation algorithm in our pipeline because of empirical evidence of its generated questions hurting domain adaptation performance.

QAGen

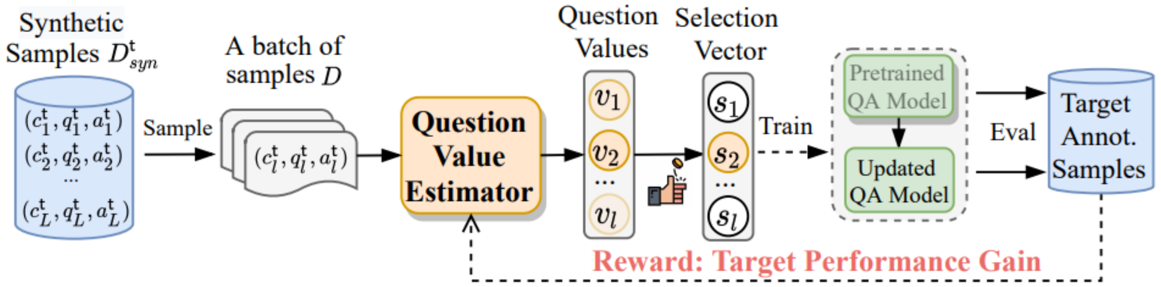
QAGen [14] model generates question and answer jointly given the input passage: (q, a) ~ p(a, q|c). Question tokens are generated first autoregressively, which are followed by answer tokens. This makes the generation of the answer conditioned on both input context (through attention on the encoder) and question.

QAGen-T5

QAGen-T5 uses a 2-step pipeline for question generation where first the question is generated given a context and then a question answering model is used to generate answers given the generated question and context pair.

QVE

Question Value Estimator [15] is a method to select questions that can improve performance on target data. Given a question answer pair it outputs a real valued score.



Training of QVE is a reinforcement learning task where synthetic data is first sampled based on bernoulli sampling. These synthetic QA pairs are then passed through a QA model. The performance gain of the QA model on the target domain is the reinforced learning reward for the QVE training. In order to mitigate instability of reinforcement learnig, the QA model is set to its pretrained checkpoints after each iteration.

Retriever

Generative Pseudo Labeling

GPL [2-1] combines a query generator with pseudo-labelling from a cross-encoder. It tries to improve on the significant shortcoming of dense retrievers. For a collection of paragraphs from the desired domain, GPL uses an existing pre-trained T5 encoder-decoder to generate suitable queries. For each generated query, It retrieves the most similar paragraphs using an existing dense retrieval model. Finally, it uses an existing cross-encoder to score each (query, and passage)-pair and train a dense retrieval model on these generated, pseudo-labelled queries.

Experiments show a slight increase in the top-1 retrieval scores but a decrease in the top-5 scores. We conclude that through the GPL technique, the *model fails to learn the key representations of the SQuAD dataset and hence fails to improve on the correct passage retrieval task*.

Model	Top5	Top1
multi-qa-mpnet-base-dot-v1	86.44	66.53

LaPraDoR

LaPraDoR [1-1] provides two ways to improve the performance of retriever models for domain adaptation.

- *Iterative Contrastive Learning*: training mechanism where question and document encoders are trained iteratively i.e., one is kept frozen while the other is being trained. The encoded sequences from the frozen encoder are used as negative instances. These sequences are added to a cache queue which is cleared whenever the training is switched between the encoders. Inbatch hard negatives were shown to be very effective in the training of retriever models, but because of a limited batch size of GPUs, a limited amount of negative instances can be used, therefore the cache mechanism helps, to overcome this problem and provides better performance. We couldn't use ICoL because the training implementation code has not been released yet.

- *LEDR*: It is a method where, during inference we multiply the score of BM25 with similarity score of LaPraDoR. This helps in improving performance of dense retrievers on unseen data.

We did not proceed with this model as we did not have the training code for this and the pretrained checkpoint for the model could only get a retrieval top 1 accuracy of ~51 on the SQuADv2 dev set.

Reader

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. As a result, out-of-domain (OOD) samples occur that diverge from the training corpora of QA systems and, upon deployment, lead to a drastic drop in the accuracy of QA systems.

A solution to the problem 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. 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.

CAQA

The authors propose a contrastive adaptation loss-based training [3-1] where the maximum mean discrepancy (MMD) metric is used in order to minimize the **intra-class discrepancies** and to maximize the **inter-class discrepancies**. Synthetic Question Generation is done using QAGen-T5 which uses a two-stage process where first given context questions are generated and then given context-query pairs, the answers for each pair are generated.

The first and the second terms in the contrastive loss below minimize the distance between the answers and context-query pairs within the same batch (minimizing intra-class discrepancies) and the last term maximizes the distance between the

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

Experiments have been reported using SQuAD dataset as the source domain given that the SQuAD dataset contains diverse domains at the intersection of various domains for experimenting with generalization performance across domains.

Training settings used in the experiments are training only on SQuAD data, using target data, questions corresponding to 10k paragraphs on the target domain, the use of all target data, or using 50k generated questions (synthetic) on 10k paragraphs, 5 questions per paragraph.

The results mentioned in the paper suggest that using generated data from Info-HCVAE hurts the performance on the target domain, similarly, with AQGen-generated data, there is a slight drop in accuracy when compared to BERT-QA trained only on the SQuAD dataset. QAGen and QAGen-T5 lead to improvements over the SQuAD-trained model. CAQA uses contrastive adaptation losses along with synthetic questions which leads to **sample efficiency** and **further accuracy gains**.

QC4QA

The paper proposes using Question Classification based filtering and sampling [16]. The questions are classified based on the type of question asked (Who, What, When...), and random subsets are sampled during training time for each epoch. The filtered and sampled questions are then pseudo-labeled and are used in a self-learning-based domain adaptation process where the distance between representations belonging to the same domain is minimized and the distance between representations for separate domains is maximized. In addition to this, the Cross-Entropy Loss for the answer spans and CAQA contrastive losses are minimized.

QADA

The authors introduce several novel ideas in order to learn better representations using hidden space augmentations, and attention-based contrastive learning losses [17]. This approach does not depend on generated synthetic data tuples (query, context, answer) as it uses pseudo labeling and learns from the labels it generates for itself given (query, context) pairs.

The target domain data is first pseudo-labeled using the QA system to generate and filter labeled target QA data. An augmentation component then performs query augmentation using multi-hop synonyms for query tokens and then sampling using **Dirichlet neighborhood sampling** in the embedding space to generate augmented tokens. For contexts, an attentive context cutoff method learns to drop context spans via a sampling strategy using attention scores. The contrastive learning-based losses are modified to account for the importance of each token rather than an average over tokens thus improving the distance metric for optimization.

Ablation studies prove the effectiveness of each of these approaches across domains where different augmentation techniques and losses lead to improvements on different datasets. Contrastive loss for source and domain data minimizing distances within a domain and maximizing distance for separate domains is the most important component leading to a huge drop in accuracy when removed.

All these models have been trained on the official train file for 2 epochs and evaluated on the dev file unless mentioned otherwise.

Reader Training	F1	EM
BERT-Base	74.67	71.15
BERT-Base domain adaptation CAQA	76.27	72.87
BERT-Base CAQA + Synthetic Data	72.41	68.91
BERT-Base QADA (4 epochs)	76.50	73.23

Theme-Wise Finetuning Reader

The training setting we use for theme-wise setting is to train on a list of held out dev set themes, finetune on a subset of paragraph question pairs from each themes and evaluate on the other subset for 30 themes.
For instance, we use question-answer pairs for a theme “Romans” to fine-tune our model and test how well it performs for an unseen question related to this theme.
We define a general model as a model which has been finetuned on the train split for all 30 themes.
For fine-tuning we used the `deepset/minilm-uncased-squad2` model finetuned on SQUAD 2.0 Dataset for QA task. From our set of experiments we conclude that:

- Performance for theme-wise finetuned models improves as the number of samples within a theme increase.

- A general model results in better performance across themes as more data helps in better generalization and language models learn few-shot.

Applying Question Generation to increase question-answer pairs in the theme wise dataset results in a similar performance where the general model beats the theme-wise model.
A downside of finetuning models on individual themes is storage require take up a lot of memory if the themes provided are many in number. For instance one MiniLM model takes 80 MB space and for 360 themes, the space would go to 28.8 GB. Given the constraints and results, the approach didn't make it to our final pipeline.

Optimization Techniques

Pruning

Pruning can reduce heavy networks to lightweight ones by removing redundancies, thus, decreasing model size and increasing inference speed. This is done by inserting additional trainable parameters, masks, into a transformer. The value of each mask variable controls whether an entire block of transformer parameters (e.g. an attention head) is used by the model. Most literature report a near-negligible performance loss upon pruning on GLUE benchmarks, however answer selection via pointer networks requires token level predictions rather than passage level classification, and requires long range attention between query and passage. Hence, results are often not promising enough while pruning transformers for Question Answering task.

Quantization

Quantization is used to convert 32-bit floating point data to smaller precision, such as 8-bit integers, which significantly cuts down the processing time. Optimum, Intel's Neural Compressor supports a variety of compression techniques, including quantization, knowledge distillation, and pruning. IncQuantizer was used to apply post-training dynamic quantization, where the scale factor for activations (to convert from floating point to integers) is dynamically based on the data range observed at runtime. This ensures that the scale factor is "tuned" so that as much signal as possible about each observed dataset is preserved. The accuracy difference was observed to be negligible. Optimum allows for building and running inference with accelerated runtime like ONNX Runtime, a cross-platform inference and training machine-learning accelerator that significantly improves transformer inference on the CPU.

Knowledge Distillation

Knowledge Distillation based approaches help with distilling knowledge from a large model trained on huge datasets to a smaller model which results in lesser inference times and comparatively better performance. Several small models have been trained distilling knowledge from larger BERT models to smaller models such as xtreme-distil which lead to inference times of upto 120ms per query-context pair but at the same time results in higher training costs as the inputs need to be passed through a larger model.
Moreover when evaluating our pipelines for open domain question answering tasks we find that the improvements of the reader model based on the reported f1 scores transfers to the pipeline. Based on this obseration we choose to use a better performing reader model than a knowledge distilled smaller model.

Using previously answered questions

Improving Latency

One way to utilize the previously answered question is by calculating the semantic and lexical similarity between the input query and all the previous queries. Semantically trained models such as Sentence BERTs can be used to find a fine threshold such as to classify a query as *same* as a previous one in the semantic sense. Once this decision is taken, the answer previously output by the pipeline can be used (provided all such QA pairs are stored offline). This could help in improving latency requirements of the system if a lot of questions are repeated. This however was not seen in the eval/train data provided.

Improving Accuracy - Active Learning

Neural models for Question-Answering require a large amount of annotated data and perform well only in the domains they were trained on. *Active Learning (AL) aims to reduce the amount of annotated data required for training* based on iteratively selecting a set of specific samples to be labelled by a human with expert knowledge of the domain. The paper [18] *combines data generation with AL to improve performance in low-resource settings*, where the target domains are diverse in terms of difficulty and similarity to the source domain.

Questions are generated using `QA2S` model and only a subset of the generated questions are kept using `LM score filtering` and `Round-trip consistency`. Different context scoring functions are used in an AL scenario with *pool-based sampling* for our specific task. AL performs better than random sampling, both when it is applied on the MRQA model and when it is applied on the data generation model as well.

For all domains, applying AL improves the QA task, especially when it is applied at the *data generation stage*.

We make use of previously answered questions by our model to improve performance. We propose to maintain a dataset consisting of the questions answered by our model along with its confidence scores. We then *select questions (answerable and non-answerable) which produced a high confidence score* from our model. These questions are then annotated by a larger model and are then used to fine-tune our original model.

Reader Training	F1	EM
BERT-Base	74.67	71.15
BERT-Base with Active Learning	75.32	71.85

Phrase Retrieval

Given a passage and an input question, cross-attention is allows us to extract spans of text as answers from the given passage. However in the case of open-domain question answering, where we are not given a particular passage, scaling can get difficult. The retreiver-reader approach retrieves top k relevant passages so that the large and heavy reader has fewer passages to read. This approach also results in *error propagation* of the retriever into the reader.

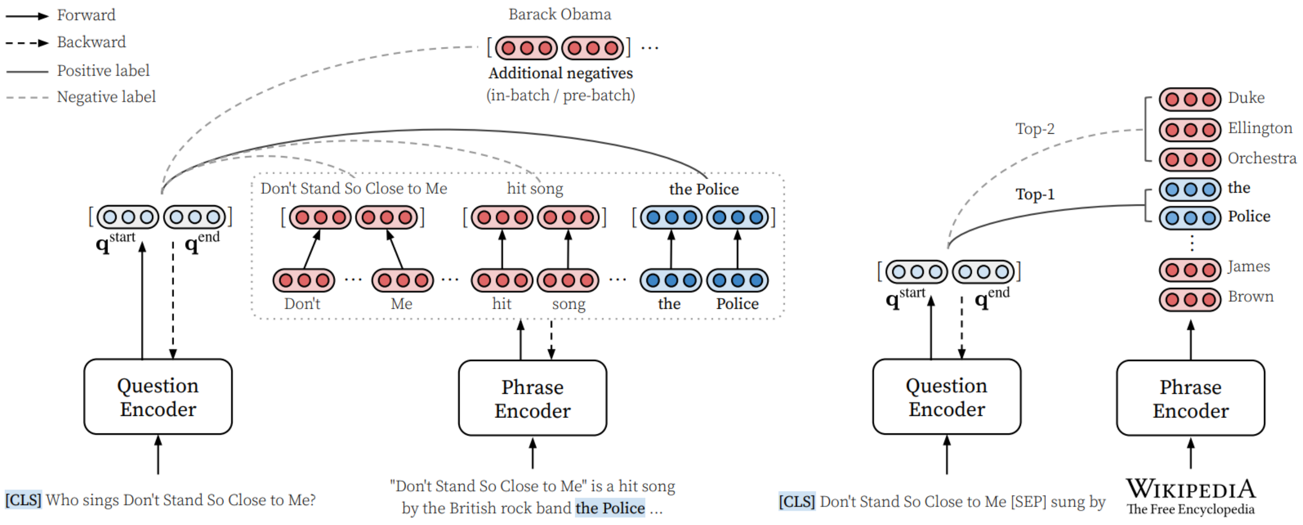
Phrase Indexed Question Answering

Phrase retrieval proposed by Seo et. al. in Phrase-Indexed Question Answering (PIQA) [19] helps take a step towards this issue. Once the passages are provided, phrases are computed and pre-indexed. To encode a phrase, the model is trained to maximize the score of a gold phrase that answers the question. This is done by using the answer annotations in QA datasets. This filters out tokens not likely to form a valid phrase which largely reduces the size of th e phrase index. Once the phrase index is built, we compute the question vector from the question encoder and answer it by performing maximum inner product search (MIPS). However, this approach was not evaluated on large-scale open domain QA.

Phrase Retrieval for Open-Domain QA

Seo et. al. then propose DenSPI [20] to pre-encode billions of phrase vectors from the entire Wikipedia database. They introduce indexable, query agnostic phrase representations for real-time open domain QA. Their phrase representations combine dense and sparse vectors. While dense representations encode the semantic meaning of tokens, sparse embeddings encode precise lexical information. Dense embeddings are obtained using BERT-Large finetuned to allow more coherent phrases to have similar start and end embeddings. For the sparse embeddings, the authors use a 2-gram based tf-idf representation. The authors were also able to reduce the size of the phrase index significantly with various engineering efforts such as the use of pointers, filtering and quantization.

Dense Phrase Retrieval



(a) Single-passage training w/ additional negatives

(b) Query-side fine-tuning & Inference

In an attempt to get rid of sparse representations in the phrase encoding model, DensePhrases [21] was proposed. DensePhrases uses SpanBERT as their language model to create dense embeddings. Motivated by contrastive learning, they apply in-batch negatives to improve dense passage representations. Another variant called the pre-batch negatives are introduced where the representations from previous batches are used as negative samples. The authors also use a T5 based question generation model to improve the quality of phrase representations. They also introduce query-side finetuning to improve the quality of question encoder. Results using this approach for our task have been tabulated below.

This approach gave us promising results during the initial part but was put aside in favour of the more flexible retriever-reader pipeline. Following are the evaluation metrics on SQuAD V2 using the DensePhrases approach.

F1	Exact Match
72.9	69.3

1. C. Xu, D. Guo, N. Duan, and J. McAuley, ‘LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval’. arXiv, 2022. ↩ ↩
2. K. Wang, N. Thakur, N. Reimers, and I. Gurevych, ‘GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval’. arXiv, 2021. ↩ ↩
3. Z. Yue, B. Kratzwald, and S. Feuerriegel, ‘Contrastive Domain Adaptation for Question Answering using Limited Text Corpora’. arXiv, 2021. ↩ ↩
4. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, ‘SQuAD: 100,000+ Questions for Machine Comprehension of Text’. arXiv, 2016. ↩
5. P. Rajpurkar, R. Jia, and P. Liang, ‘Know What You Don’t Know: Unanswerable Questions for SQuAD’. arXiv, 2018. ↩
6. V. Karpukhin *et al.*, ‘Dense Passage Retrieval for Open-Domain Question Answering’. arXiv, 2020. ↩
7. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. arXiv, 2018. ↩
8. M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, ‘SpanBERT: Improving Pre-training by Representing and Predicting Spans’. arXiv, 2019. ↩
9. Y. Liu *et al.*, ‘RoBERTa: A Robustly Optimized BERT Pretraining Approach’. arXiv, 2019. ↩
10. K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, ‘ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators’. arXiv, 2020. ↩
11. P. He, X. Liu, J. Gao, and W. Chen, ‘DeBERTa: Decoding-enhanced BERT with Disentangled Attention’. arXiv, 2020. ↩
12. P. He, J. Gao, and W. Chen, ‘DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing’. arXiv, 2021. ↩
13. D. B. Lee, S. Lee, W. T. Jeong, D. Kim, and S. J. Hwang, ‘Generating Diverse and Consistent QA pairs from Contexts with Information-Maximizing Hierarchical Conditional VAEs’. arXiv, 2020. ↩
14. S. Shakeri *et al.*, ‘End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems’. arXiv, 2020. ↩

15. X. Yue, Z. Yao, and H. Sun, ‘Synthetic Question Value Estimation for Domain Adaptation of Question Answering’. arXiv, 2022. ↩
16. Z. Yue, H. Zeng, Z. Kou, L. Shang, and D. Wang, ‘Domain Adaptation for Question Answering via Question Classification’, in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 1776–1790. ↩
17. Z. Yue, H. Zeng, B. Kratzwald, S. Feuerriegel, and D. Wang, ‘QA Domain Adaptation using Hidden Space Augmentation and Self-Supervised Contrastive Adaptation’. arXiv, 2022. ↩
18. M. Schmidt, A. Bartezzaghi, J. Bogojeska, A. C. I. Malossi, and T. Vu, ‘Improving Low-Resource Question Answering using Active Learning in Multiple Stages’. arXiv, 2022. ↩
19. M. Seo, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi, ‘Phrase-Indexed Question Answering: A New Challenge for Scalable Document Comprehension’. arXiv, 2018. ↩
20. M. Seo, J. Lee, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi, ‘Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index’. arXiv, 2019. ↩
21. J. Lee, M. Sung, J. Kang, and D. Chen, ‘Learning Dense Representations of Phrases at Scale’. arXiv, 2020. ↩