Cooperative Self-training of Machine Reading Comprehension

Anonymous ACL submission

Abstract

Pretrained language models have significantly improved the performance of downstream language understanding tasks, including extractive question answering, by providing high-quality contextualized word embeddings. However, training question answering models still requires large amounts of annotated data for specific domains. In this work, we propose a cooperative self-training framework, RGX, for automatically generating more non-trivial question-answer pairs to improve model performance. RGX is built upon a masked answer extraction task with an interactive learning environment containing an answer entity **R**ecognizer, a question **G**enerator, and an answer e**X**tractor. Given a passage with a masked entity, the generator generates a question around the entity, and the extractor is trained to extract the masked entity with the generated question and raw texts. The framework allows the training of question generation and answering models on any text corpora without annotation. We further leverage a reinforcement learning technique to reward generating high-quality questions and to improve the answer extraction model's performance. Experiment results show that RGX outperforms the state-of-the-art (SOTA) pretrained language models and transfer learning approaches on standard question-answering benchmarks, and yields the new SOTA performance under given model size and transfer learning settings.

1 Introduction

Recent studies have shown that language model pretraining provides high-quality text representations and significantly improves neural networks' performance on a variety of natural language processing (NLP) tasks (Peters et al., 2018). Based on the popular Transformer architecture (Vaswani et al., 2017), various language models have been proposed (Devlin et al., 2018; Liu et al., 2019; Clark et al., 2020). These models are pretrained to predict a masked word in a given context from large corpora, and generate a contextual representation that encodes semantic and syntactic information. After finetuning, these representations significantly improve performance on downstream NLP tasks. Although masked language modeling is a powerful self-supervised learning technique, annotation on large-scaled data is still necessary for finetuning on difficult downstream tasks, including extractive question answering (QA)[1] where a large number of labeled question-answer pairs are required as a training corpora.

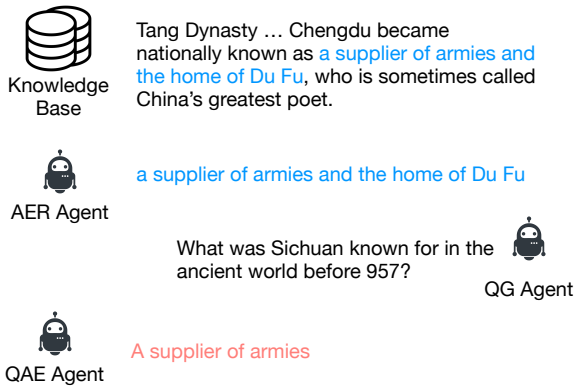

Figure 1: The pipeline of semi-supervised question answering (machine reading comprehension) by RGX. AER (answer entity **R**ecognition) agent recognizes answer entity from a given passage; QG (question **G**eneration) generates a question based on the passage and entity; QAE (question-answering e**X**tractor) extracts answer from the question and passage.

Previous studies showed that the QA models can be improved by training on synthetic question-answer pairs, namely self-training (Sachan and Xing, 2018; Puri et al., 2020; Shakeri et al., 2020; Bartolo et al., 2021). The core step of these work is pretraining a question-answer pair synthesis model on a seed corpus, and apply the generator on target domains to obtain synthetic training data. The QA model learns domain knowledge after finetuning on

[1] Also referred to as machine reading comprehension. The two terms are used interchangeably in this paper.

the synthetic data, and thus the domain adaptation is improved. However, the gap between the pretraining (i.e., seed) and the target corpus still exists, in terms of domain knowledge, question difficulty, and language style. The gap affects the quality of the synthetic training data.

We thus propose a framework that allows cooperative self-training for both QA pair synthesis and question answering to better adapt the synthesis models to the target domain and improve the learning of the QA models. In the framework, we construct a cooperative environment where a question generator and an answer extractor work together to solve a masked entity prediction problem. We first leverage an entity recognizer to mask out an entity in a provided passage. The question generator then outputs a question based on the masked passage. With the generated question and the original, unmasked passage, we train the answer extractor to select the correct answer spans, which are the masked entity. The extractor is also the final model used for extractive QA. To extract the spans accurately, the generator has to provide a good question, and the extractor should select the most likely tokens. We design the reward function such that it favors the questions leading to correct answers. We also gradually increase the difficulty of generated questions (Karpukhin et al., 2020) by rewarding the questions that are not answered correctly but with low extraction losses via a stochastic expectation-maximization technique. The technique allows us to train the extractor with challenging examples incrementally. We call our algorithm RGX since it incorporates an answer entity **R**ecognizer, a question **G**enerator, and an answer e**X**tractor. The RGX pipeline is illustrated in Figure 1.

With RGX, we can train a QA model for any unlabeled target domain given the corresponding text corpora and a labeled QA corpus in a seed domain (either the same or different from the target). We show that RGX outperforms SOTA approaches in QA benchmark datasets when domain specific human labels are not available during finetuning. In this work, we make the following contributions:

1. We propose a cooperative self-training framework, RGX, which contains an answer entity recognition, question generation, and answer span extraction to automatically generate non-trivial QA pairs on unlabeled corpora.
2. We design a expectation-maximization synthetic QA selection that identifies difficult but answerable questions without supervision to incrementally train the QA model with challenging examples, and a AER-based maximum mutual information inference method for question answering.
3. Experiments show that our method significantly outperforms SOTA pretrained QA models and self-training QA baselines.

2 Related Work

Reinforcement learning and self-training have emerged recently for learning language generation in addition to maximum likelihood training. To optimize text generation models directly with non-differentiable objective functions, Rennie et al. (2017) proposed self-critical sequence training (SCST) using a policy gradient (Kakade, 2001; Silver et al., 2014). On the other hand, self-training has been shown to be effective in many tasks, such as machine translation (He et al., 2019), image classification (Xie et al., 2020), and structured database-grounded question answering (Xu et al., 2020).

In the domain of question answering, a question generator can be used for joint answer prediction (Tang et al., 2017; Duan et al., 2017), and synthetic QA data are used for in-domain data augmentation (Sachan and Xing, 2018; Puri et al., 2020; Liu et al., 2020; Klein and Nabi, 2019) and out-of-domain adaptation. Lewis et al. (2019b) and Lee et al. (2020) introduced models for question answering under unsupervised/zero-shot settings. Shakeri et al. (2020) proposed generating synthetic question-answer pairs with an end-to-end model simultaneously. Bartolo et al. (2021) improved the question synthesis by training with difficult QA cases from the AdversarialQA corpus (Bartolo et al., 2020) and fine-grained answer synthesis by multi-model voting. We include more related studies in Appendix A.

In this work, we mainly compare our method with latest baselines, Shakeri et al. (2020) and Bartolo et al. (2021) that reported results on out-of-domain adaptation. Besides improved QA performance, our framework, RGX, differs from the previous work in the following aspects: (1). Our method features reinforced finetuning of the QA Synthesizer, (2). Our framework supports and improves maximize mutual information inference in test time, and (3). Our work did not use complicated data annotation, e.g. AdversarialQA.

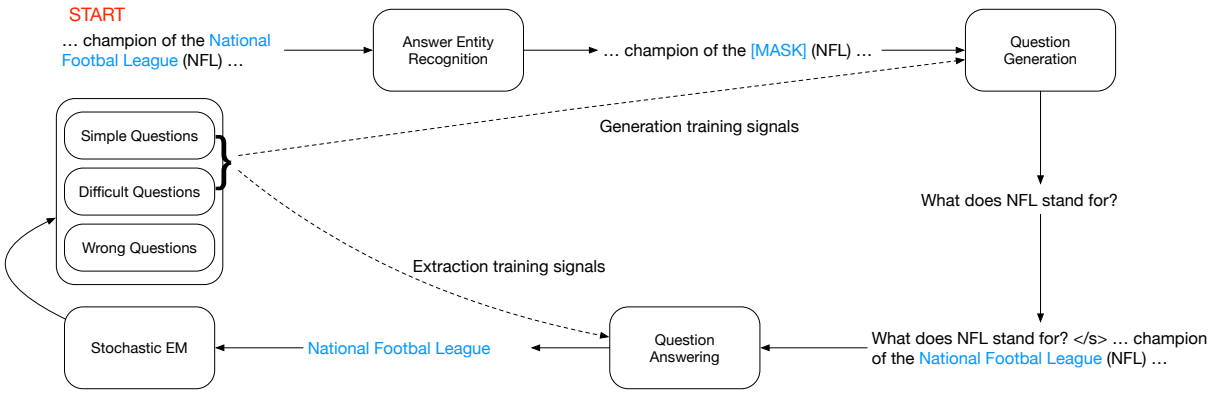

Figure 2: The cooperative learning pipeline for question answering. The pipeline starts from a passage and follows the steps: (1) recognizing a potential answer entity, (2) generating a question asking about the answer entity, and (3) answering the question by extracting the answer span in the passage.

3 RGX Framework

In this section, we first introduce (1). the QA synthesis pipeline, (2). cooperative self-training for both QA synthesis and question answering, and (3). an improved maximum mutual information inference strategy. The self-training pipeline of RGX is shown in Figure 2.

3.1 Data Synthesis

Given a passage p, our goal is generating a set of questions q and answers a for the self-training of the QA model. The RGX model first recognize potential answer entities (AE) in p with an answer entity recognition (AER) model, and then generate question based on the recognized AEs with a question generation (QG) model, and fine-grain the AEs with a pretrained question answer extraction (QAE) model.

3.1.1 Answer Entity Recognition (AER)

Latest QA synthesis models, QAGen2S (Shakeri et al., 2020) and SynQA (Bartolo et al., 2021), directly generate questions from passages by modeling $P_{qg}(q|p)$. In RGX, we first recognize all potential answer entities in a passage before generating questions for (1). increasing question diversity and coverage, and (2). modeling the mutual information between question generation and answering models in test time. The AER model in trained on the seed QA corpus.

We found that using an off-the-shelf named entity recognition (NER) model pretrained on the CONLL 2003 shared task (Bender et al., 2003) performs poorly as a AER model (shown in our experiments). To learn an effective recognizer, given a passage p and an annotated answer entity e, we select the sentence s containing e from p and train language models to recognize e in s. We tried two models for this task: a BIO sequence tagging model (AER-Tag) and a extractive AER model, which is similar to an extractive question answering model, for easier decoding. The model predicts the start and end positions of the answer entity e. With this method, we get potential answer entities by probabilities of all candidate spans.

3.1.2 Masked Question Generation

With AER, we replace the answer entity e in the passage p with a [MASK] token and obtain the masked passage p^*. We then build a question generator Q (denoted as QG interchangeably) that outputs answerable questions q in natural language with the concatenation of p^* and e as input, i.e., $q = Q([p^*, e])$. We adopt the BART sequence-to-sequence model (Lewis et al., 2019a) as the architecture of Q in our implementation, and we train Q on the question-answer pairs in the seed corpus by maximizing the likelihood of annotated questions.

3.1.3 Answer Extraction as Fine-grained AER

The answer extraction model A (denoted as QAE, question answering extractor) takes generated question q and the original passage p as inputs. Following the standard extractive QA method, we predict the answers by

$$I_{st}, I_{ed} = A([q, p]) \quad (1)$$

where I_{st} and I_{ed} stand for the start and end positions of e in p, respectively. We train the QAE model to predict I_{st} and I_{ed} separately with cross entropy losses.

Besides being trained with synthetic QA pairs and evaluated for the final QA performance, the

QAE model is also a part of the data synthesis pipeline. After generating questions with the QG model, we use a pretrained QAE model to answer the generated questions. The final synthetic dataset is constructed by selecting generated questions and their corresponding QAE outputs.

3.2 Cooperative Self-training

Although the pretrained models can generate synthetic QA pairs from corpora in unseen domains, there is always a domain shift from the seed QA corpus for pretraining to the target. To efficiently adapt the pretrained models to the new domains, we propose a cooperative self-training algorithm that allows finetuning on the target corpora without additional annotations. The finetuning is based on a three-agent (AER, QG, QAE) cooperative framework, RGX. The pipeline is illustrated in Figure 2 and comprises the following steps:

1. Produce a masked passage by replacing an answer entity selected by AER with the '[MASK]' token.
2. Generate a question asking about the masked entity.
3. Feed the generated question and the original passage into the QAE to predict an answer span.
4. Optimize the QAE model with selected QA pairs.
5. Optimize the QG model with selected QA pairs.

In the proposed pipeline, all the AER, QG, and QAE models need pretraining to provide a reasonable start point for the cooperative self-training. However, the domain gap between the pretraining and the target corpus causes performance degradation. To mitigate the gap, we propose to measure the quality of generated questions and incorporate the measurement in loss functions. The quality is defined in two folds, correctness and difficulty. Firstly, the question should be fluent and answerable, and secondly, it should not be too trivial. To automatically select high-quality generated QA pairs, we introduce a expectation-maximization (EM) method based on QAE losses that learns the question quality without supervision.

3.2.1 Synthetic QA Selection with EM

To select synthetic QA pairs for finetuning, we first divide the generated questions based on the QAE loss for each question into three groups: low-, medium-, and high- loss questions. We can interpret questions with low loss as simple ones that the QAE model can easily answer. Medium-loss questions are challenging for the QAE, while those with high loss usually contain noise (e.g., containing grammatical errors or asking about incorrect answers). If we train the answering model with all questions, the training signal would be very noisy due to the high-loss questions. If we only reward questions that are correctly answered, the generator will converge to a trivial local optima. Thus, we train the QG and QAE model with the low- and medium- loss questions, namely simple and challenging questions. For the entire pipeline to be fully-automatic, we classify a given QA pair into one of the three types described above. Note that simply setting the thresholds as hyper-parameters is difficult since the loss decreases as the QAE model varies with different passages and domains. In order to find the thresholds adaptively, we apply an expectation-maximization (EM) algorithm to cluster synthetic QA pairs for each passage.

We finetune both QG and QAE models with the selected simple and challenging QA pairs. After the training, re-running the RGX pipeline with the finetuned question generation model leads to improved data synthesis. Training the QAE model on the updated synthetic dataset can significant outperform the previous finetuned QAE model.

3.2.2 Maximum Mutual Information QA

Li and Jurafsky (2016) proposed a maximum mutual information (MMI) decoding method for machine translation, and Tang et al. (2017) proposed a MMI method for jointly learning question generation and answering models. There is no previous study to our knowledge that applies MMI inference in test time of question answering that improves the final performance, because (1). modeling $P(q|p,a)$ for all possible answers (spans) a is too inefficient, and (2). Unlike the QAE model that receives loss signals from all words in a given passage, the QG model does not receive loss signal from the passage directly, so $P_{qg}(q|p,a)$ it is less accurate for ranking answer spans.

However, the AER and self-training strategy enable efficient MMI inference for QA,

$$a = \underset{a}{\text{argmax}}[\alpha \log P_{qg}(q|p,a) + \beta \log P_{qa}(a|p,q)]$$

In test time, we run the RGX pipeline for each passage without additional training to get fine-grained AEs and corresponding questions. On the other hand, we take the top-k spans predicted by the QAE model, and only keep the top prediction and those which also appears in the fine-grained AE set. The filtering strategy dramatically reduces the number of potential answer spans, and removes unreasonable spans predicted by the QAE model.

4 Experiments

4.1 Modules

In this work, we train three modules for building the cooperative self-training environment RGX, i.e., the answer entity recognizer (AER), the question generator (QG), and the question-answering extractor (QAE). We used a BERT (Devlin et al., 2018) model for AER, a BART (Lewis et al., 2019a) model for QG, and an ELECTRA (Clark et al., 2020) model for AER and QAE. To compare with the results reported in Shakeri et al. (2020) and Bartolo et al. (2021), we also evaluate the performance of training BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) models on the synthetic QA data generated by RGX.

4.2 Data

In our experiment work, we leveraged Natural Questions (Kwiatkowski et al., 2019) and SQuAD v1.1 (Rajpurkar et al., 2016) as the seed corpora for pretraining all modules introduced above. To evaluate the performance of the proposed RGX on question answering tasks with different difficulty levels, we conduct experiments on both SQuAD v1.1 (Rajpurkar et al., 2016) and MRQA (Fisch et al., 2019) out-of-domains (BioASQ, TextbookQA, RACE, RelationExtraction, DuoRC, and DROP). In the following sections, we use the term SQuAD to represent the SQuAD v1.1 corpus. For self-training, we sample 3000 passages from the training set of each corpus for data synthesis. More details about the data are provided in Appendix C

4.3 Implementation Details

Pretraining We pretrain the AER, QG, and QAE models on NaturalQuestions and SQuAD (i.e., the seed) corpora. For NaturalQuestions, we only use the data points containing a short answer. For Cooperative training, we follow the steps described in Section 3.2 for the cooperative training phase.

Self-training We apply self-training for QG and QAE by finetuning the models on selected synthetic QA pairs using the same method as pretraining. The AER model is fixed after pretraining. The QAE model is finetuned using the official Huggingface (Wolf et al., 2019) training scripts for question answering. We will open-source the RGX framework if the submission is accepted. More details about the hyperparameters we use in different training phases are shown in Appendix B.

4.4 Experiment Results

We assess the performance of RGX with both semi-annotated and zero-annotated evaluation on unseen domains. In our semi-annotated setting, we use the annotated answer entities in the target corpora but utilize QG to generate questions for obtaining the training question-answer pairs. The labeled questions are not used. We employ no annotation from the target corpora for the out-of-domain task but automatically construct the question-answer training pairs with entities and questions inferred by AER and QG on the corpora.

4.4.1 Semi-annotated Evaluation

The model performance with the pretrained QA model, RGX, and SOTA trained with full-supervision is shown in Table 1.

Models	EM	F1
Source domain: NQ, Target domain: SQuAD		
ELECTRA-large (NaturalQuestions)	67.8	80.3
RGX	83.1	90.7
–w/o Coop. ST	81.2	89.1
ELECTRA-large (SQuAD)	89.7	94.9

Table 1: The performance of the question answering models in the semi-annotated setting. RGX stands for our cooperative training approach, and Coop. ST stands for cooperative self-training.

Table 1 shows that RGX yields improvement over the pretrained model, approaching the SOTA performance of the fully trained ELECTRA-large-discriminator model. The experiment result suggests that the cooperative learning strategy improves the question generation model.

4.4.2 Out-of-domain Evaluation

We also evaluate the models in unseen domains, where we do not use any annotated QA for fine-tuning. We train the QAE models based on the synthetic training data and evaluate the models on the target domains. We compare RGX with latest self-trainig QA methods, QAGen2S (Shakeri et al., 2020) and SynQA (Bartolo et al., 2021). Since QAGen2S did not report full MRQA results, we implemented our own version. We first present the RGX performance and the results reported by the authors QAGen2S and SynQA, and then conduct ablation study by training different language models on RGX synthetic QA data.

The full evaluation results on MRQA out-of-domains are shown in Table 2, and the experiment

Model / Domain	BioASQ / Bio		TextbookQA / Book		RACE / Exam		RelExt. / Wiki		DuoRC / Movie		DROP / Wiki		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Source Domain: NaturalQuestions$_{wiki}$, Method: Extraction														
ELECTRA-large	41.9	59.0	31.9	41.5	32.4	43.4	67.7	81.8	40.0	48.5	**39.3**	**51.1**	42.2	54.2
QAGen2S	43.2	64.1	39.9	51.7	33.7	45.5	71.6	84.4	43.8	53.2	24.2	37.1	42.7	56.0
RGX (Ours)	**50.3**	**70.1**	**49.9**	**60.9**	**40.3**	**52.4**	**76.1**	**87.2**	**47.8**	**58.4**	27.6	42.1	**48.7**	**61.9**
– w/o MMI	49.7	69.1	49.4	60.6	39.7	51.5	75.4	86.7	46.9	57.5	27.1	41.7	46.8	61.2
– w/o EM	48.2	67.9	47.4	59.8	38.3	50.5	74.1	86.2	46.6	56.9	26.1	40.9	46.8	60.4
– w/o Coop. ST	45.4	66.4	41.9	53.8	35.1	47.2	72.7	85.4	45.5	54.9	24.6	37.9	44.2	57.6
Source Domain: SQuAD$_{wiki}$ (SQuAD+AQA+Wiki for SynQA), Method: Extraction														
ELECTRA-large	58.7	73.1	43.0	53.6	38.3	52.5	79.0	88.4	53.1	64.2	48.3	60.8	53.4	65.4
QAGen2S	56.8	71.7	48.0	56.5	43.4	54.9	73.4	84.8	53.3	64.6	42.2	54.5	52.8	64.5
SynQA (extra data)	55.1	68.7	41.4	50.2	40.2	54.2	78.9	**88.6**	51.7	62.1	**64.9**	**73.0**	55.3	66.1
RGX (Ours)	**60.3**	**74.8**	**51.2**	**61.2**	**44.9**	**58.7**	**79.2**	**88.6**	**57.4**	**66.2**	47.6	60.9	**56.8**	**68.4**
– w/o MMI	59.2	73.6	50.1	60.4	46.3	57.6	78.9	88.5	56.2	65.7	46.9	60.6	56.3	67.7
– w/o EM	52.1	64.0	50.6	58.9	35.4	48.3	75.6	85.9	55.6	64.9	40.7	53.2	51.7	62.5
– w/o Coop. ST	57.5	72.1	48.6	57.0	43.8	55.2	74.3	85.3	53.9	65.3	43.0	55.1	53.5	65.0
Source Domain: SQuAD$_{wiki}$, Method: Prompt Tuning + Seq2seq Generation														
T5-large	54.6	71.1	37.9	61.9	15.0	53.1	74.5	86.5	48.2	65.2	40.4	51.9	45.1	64.9
T5-large + RGX	55.1	71.6	41.1	64.2	15.5	55.1	75.9	87.1	49.5	66.2	42.9	53.8	46.7	66.3

Table 2: The QA performance evaluation on the out-of-domains of the MRQA benchmark.

	QAGen2S	SynQA	RGX
Pretraining	XQ	SQ+AQA	XQ
Synthesis	Target	Wikipedia	Target
Finetuning	XQ+Syn	SQ+AQA+Syn	XQ+Syn
AER Model	None	None	ELECTRA
Coop. ST	No	No	Yes
QA Num.	1M	1.5M	0.3M

Table 3: Comparison of different self-training methods. XQ stands for "NaturalQuestions or SQuAD".

Models	EM	F1
Source domain: NQ, Target domain: SQuAD		
Pretrained NQ	67.8	80.3
RGX + NER	27.4	35.4
RGX + AER-Tag	71.4	82.4
RGX + AER-LM	72.7	85.9
RGX + AER-EM	79.2	89.4
Supervised ELECTRA-large	89.7	94.9

Table 4: Comparison of different AER strategies. NER stands for the BERT named entity recognition model trained on the CONLL 2003 shared task.

setting comparison is shown in table 3. The results show that the models trained with the RGX framework achieve significantly higher EM and F1 scores on most domains, comparing to both pretrained QA models and self-training baselines. The results showed that the RGX model achieves 7.7 and 3.0 average F1 improvement over ELECTRA, the SOTA pretrained language model for QA, by pretraining on NQ and SQuAD respectively. The improvement over previous SOTA self-training QA methods, QAGen2S and SynQA, is also significant on both pretraining corpora, although SynQA applies complicated adversarial QA annotation. The largest gain we got is adapting NQ model to TextbookQA domain, increasing 18.0 EM and 19.4 F1 scores. Note that our model still outperforms all baselines without MMI. The performance on the DROP benchmark drops since DROP requires multi-step reasoning, but the synthetic generation model tends to generate safe question-answer pairs. We also found that without selecting harder questions with SEM in RGX, the performance is significantly lower. These facts indicate that the QA model needs hard training examples for better performance, and explains the good performance of SynQA on DROP. For the same reason, the performance drop led by removing EM from RGX is significantly larger when the QG model is pretrained on SQuAD, since SQuAD questions are more coherent with the context than NQ, and selecting simple questions for RGX training will encourage the model to generate trivial questions, which is harmful for the QA training.

4.5 Analysis

4.5.1 Answer Entity Recognition

We first compare the performance of different AER models and strategies by setting NQ as the source domain and SQuAD 1.1 as the target domain in Table 4. The results showed that the choice of AER model and strategy significantly influences

	ELECTRA		Top-k+MMI		AER+MMI	
	EM	F1	EM	F1	EM	F1
BioASQ	58.7	73.1	57.8	72.9	59.9	74.0
TextbookQA	43.0	54.6	44.6	54.9	45.3	55.4
RACE	38.3	52.5	38.1	52.4	39.7	54.1
RelExt	79.0	88.4	78.6	88.3	79.2	88.6
DuoRC	53.1	64.2	52.6	64.3	53.8	65.1
DROP	48.3	60.8	46.7	60.8	49.7	61.5

Table 5: Comparison between maximum mutual information inference performance grounded on AER results and top-k ($k = 20$) predictions of the QA model.

Models	Mean Len.	Std Len.	Vocab
Ground-truth	11.29	3.72	988703
Semi-anno. RGX	10.54	1.91	923191
–w/o Coop. ST	10.49	2.48	919105
Zero-anno. RGX	10.53	1.94	873300
–w/o Coop. ST	10.57	2.63	789924
–w/o AER	10.60	1.87	743454
–w/o EM	10.18	1.62	692301

Table 6: The vocabulary sizes and lengths of Annotated and generated questions on SQuAD under both semi- and zero-annotated settings in unseen domains

the final QA performance. The low performance of the NER model trained on CONLL shared task suggests the importance of our AER module. We notice that the improvement from the cooperative learning over the pretrained models is higher in the zero-annotation setting than the semi-annotated task. The observation indicates that the model trained with RGX is more robust against the automatically recognized answer entities. More details about the AER methods are shown in Appendix D.

The AER method also enables and improves the maximum mutual information (MMI) inference in test time. Table 2 shows that MMI achieves the best performance, and we also show that the MMI accuracy is hurt without AER. Table 5 shows that MMI grounded on AER constantly outperform the ELECTRA model, but grounding on top-k seriously hurts the EM scores. Some invalid answer

Domain	RGX w/o Coop. ST		RGX	
	Hit	BLEU	Hit	BLEU
BioASQ	68.1	5.9	75.8	12.7
TextbookQA	43.7	7.5	58.2	13.2
RACE	8.3	5.2	12.3	6.8
RelExt.	47.4	2.8	54.2	3.3
DuoRC	53.5	6.7	60.0	7.5
DROP	73.5	12.3	75.3	9.3

Table 7: Evaluation of the answer hit rates and question BLEU scores of the synthetic data.

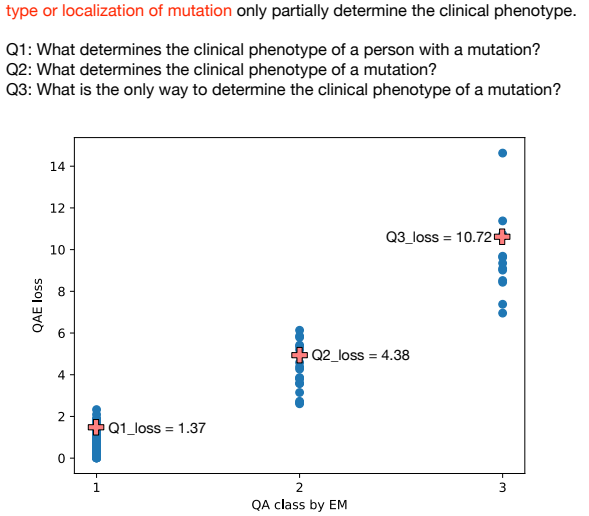

Figure 3: Generated questions about the same answer entity classified into different types by EM.

predictions leads to low question generation perplexities, which makes MMI inference noisy. Table 6 shows that the QG model generated more diversed questions based on the AER outputs.

4.5.2 Synthetic QA Selection with EM

Previous experiments showed that selecting non-trivial synthetic QA pairs is essential for RGX to achieve high performance. Table 2 shows that the performance of cooperative self-trained RGX is much lower than the pretrained baseline without EM. If selecting QA pairs with low perplexities instead of EM, the QA diversity is significantly lower as shown in Table 6, thus makes the QAE model overfit to simple training cases and hurts the QA accuracy. We show questions about the same answer entity being classified into simple, challenging, and difficult types by EM in figure 3. The data points in the plot represents the losses of synthetic QA pairs and the predicted QA type. Based on the highlighted answer entity, question 1 and 2 are predicted as correct questions, while question 3, which has a relatively high QAE loss, is regarded as a wrong question. Note that we only generate one question for each span recognized by the AER model, but different questions might be re-directed to the same AE after QAE fine-graining.

4.5.3 Cooperative Self-training

We found that the cooperative self-training method improves domain adaptation ability of self-trained QA models by increasing both accuracy and diversity of QA synthesis.

Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.

Annotated	Pretrained	RGX
Saint Bernadette Soubirous	**a Marian place of prayer and reflection**	**a Marian place of prayer and reflection**
To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?	what is the grotto at st bernadette's?	what is the grotto in st bernadette school?
a copper statue of Christ	**the grotto at Lourdes, France**	**Venite Ad Me Omnes**
What is in front of the Notre Dame Main Building?	where is the grotto located at st bernadette school?	what is the message on the statue in front of st bernadette school?
the Main Building	**Immediately behind the basilica is the Grotto**	**1858**
The Basilica of the Sacred heart at Notre Dame is beside to which structure?	what is the grotto in st peter's school?	when was the grotto at lourdes built?
a Marian place of prayer and reflection	**copper statue of Christ with arms upraised**	**a simple, modern stone statue of Mary**
What is the Grotto at Notre Dame?	what is it a statue of christ?	what is the statue at st bernadette school?
a golden statue of the Virgin Mary	**a replica**	**the grotto at Lourdes, France**
What sits on top of the Main Building at Notre Dame?	is the grotto at st bernadette school in paris a replica of which European landmark?	what is the replica of st bernadette's school in paris?

Table 8: An example of a passage in the training set of the SQuAD corpus. We list the annotated question-answer pairs, and the question-answer pairs generated by the models pretrained on NQ and finetuned by RGX. The bold texts are annotated or recognized answer entities. Adapting from NQ is difficult since the questions in NQ do not strictly coherent with a given context. More generation examples are shown in Appendix E.

Accuracy. We also evaluate the quality of the generated QA pairs without a downstream task by assessing the answer entity hit rate and the BLEU scores of generated questions using the evaluation sets of each domain. The results are shown in Table 7, indicating that RGX find mores human-annotated answer entities, and the generated questions have higher BLEU scores on all domains. The evaluation results show that the synthetic QA pars generated by RGX covers more human annotated answer entities, and the generated questions are more similar to human annotations than the pretrained question generation model.

Diversity We compare the lengths and vocabulary sizes of the questions and summarize the statistics in Table 6, which shows that the ground-truth questions are longer and more diverse in vocabulary than the generated ones. However, the cooperative self-training, together with AER and EM, improves the vocabulary diversity. We observe a correlation between the vocabulary size and the QA performance reported in Table 1 and 4, presumably because the QAE model requires diverse knowledge for training. Thus, we believe generating more diverse QA pairs with good quality will be a critical next step to improve RGX.

Case Study. An example of a SQuAD passage is shown in Table 8. We list the annotated and generated question-answer pairs by different models. The table shows that the models can recognize reasonable answer entities other than the annotated ones, and RGX generates more natural QAs.

5 Conclusion

We propose a cooperative self-training framework, RGX, consisting of an answer entity Recognizer, a question Generator, and an answer eXtractor, for question generation and answering. We also introduce in the framework an expectation-maximization method that measures the quality of generated questions for reinforced finetuning of the question generation models. Experiments show that RGX significantly outperforms pretrained and self-trained model baselines while adapted to unseen domains, suggesting that RGX is a promising framework for making extractive question answering methods more scalable and less dependent on human annotation.

References

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. *arXiv preprint arXiv:2104.08678*.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of CoNLL-2003*, pages 148–151. Edmonton, Canada.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Retrofitting structure-aware transformer language model for end tasks. *arXiv preprint arXiv:2009.07408*.

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.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Bhargav, Dinesh Garg, and Avirup Sil. 2019. Span selection pre-training for question answering. *arXiv preprint arXiv:1909.04120*.

Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in neural information processing systems*, pages 2149–2159.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2019. Convert: Efficient and accurate conversational representations from transformers. *arXiv preprint arXiv:1911.03688*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2021. Question answering infused pre-training of general-purpose contextualized representations. *arXiv preprint arXiv:2106.08190*.

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

Sham M Kakade. 2001. A natural policy gradient. *Advances in neural information processing systems*, 14:1531–1538.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*.

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. *arXiv preprint arXiv:2005.13837*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019b. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *arXiv preprint arXiv:2102.07033*.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Jiancheng Lv, Nan Duan, and Ming Zhou. 2020. Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space. *arXiv preprint arXiv:2010.01475*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hongyin Luo, Lan Jiang, Yonatan Belinkov, and James Glass. 2019. Improving neural language models by segmenting, attending, and predicting the future. *arXiv preprint arXiv:1906.01702*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

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.

Siamak Shakeri, Cicero Nogueira dos Santos, Henry 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. *arXiv preprint arXiv:2010.06028*.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. *arXiv preprint arXiv:1806.04168*.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. 2020. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. *arXiv preprint arXiv:2012.00857*.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

Silei Xu, Sina J Semnani, Giovanni Campagna, and Monica S Lam. 2020. Autoqa: From databases to qa semantic parsers with only synthetic training data. *arXiv preprint arXiv:2010.04806*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

A More Related Work

Representation learning has been an important topic in NLP area since neural language models were proposed (Bengio et al., 2003). Based on word co-occurrence, Mikolov et al. (2013) and Pennington et al. (2014) proposed language embedding algorithms to model word-level semantics. Recent studies have focused on pretraining contextualized word representations with large-scaled corpora (Peters et al., 2018). State-of-the-art representation models are pretrained with the masked language modeling task (Devlin et al., 2018; Liu et al., 2019; Clark et al., 2020) using the Transformer architecture (Vaswani et al., 2017).

Different variants of masked language models have been investigated to improve performance in downstream tasks. Joshi et al. (2020) leveraged a masked span generation task instead of word prediction. Fei et al. (2020) and Shen et al. (2020) proposed models that learns better syntax knowledge with syntactic distances (Shen et al., 2018) and heights (Luo et al., 2019). Henderson et al. (2019) and Humeau et al. (2019) showed that pretraining language models on dialog corpora perform better on dialog-related downstream tasks, as compared to pretraining on Wikipedia. A span selection pretraining objective is proposed in Glass et al. (2019) to reduce the gap between the pretraining and downstream finetuning stages and to improve the performance on the QA task. Some applications of generated questions are shown in (Lewis et al., 2021; Jia et al., 2021).

In contrast to self-training methods that usually adopt a teacher-student learning strategy, cooperative learning pipelines contain several agents working together to learn as much knowledge as possible. A typical cooperative learning framework is generative adversarial networks (GAN) (Goodfellow, 2016; Goodfellow et al., 2014), where a generator is optimized to confuse a discriminator, and a discriminator is trained to distinguish real examples from generated ones. Sequence GAN is further designed for learning diverse text generation (Yu et al., 2017). Unlike the adversarial learning method where two networks work for opposite goals, other studies proposed learning environments in which different agents learn the same objective functions for language emergence (Lazaridou et al., 2016; Mordatch and Abbeel, 2018; Havrylov and Titov, 2017), including simple natural language, compositional language, and

symbolic language.

B Hyper-parameters

There are three phases of model training in this work: pretraining on the Natural Question corpus, cooperative adaptation with reinforcement learning on the target corpora, and final finetuning on the target corpora. We adopt most of the hyper-parameters reported in the original BERT (Devlin et al., 2018), BART (Lewis et al., 2019a), and ELECTRA (Clark et al., 2020) papers. We select the final finetuning learning rates from $\{3e-5, 4e-5, 5e-5\}$ and report the highest performance. All the other hyper-parameters are the same as reported in the corresponding papers. For all the phases, we fix $eps = 1e-6$ and $s_w = 2000$, where s_w is the number of warm-up steps, and we apply no weight decays. In the following sections, we describe the details of each training phase. We use BART-large (406M parameters) and ELECTRA-large (335M parameters) models for our experiments. We run our experiments on 2 Tesla V100 GPUs. Training the QAE models on augmented data takes around 4 hours.

For maximum mutual information inference process shown in the equation below,

$$a = \underset{a}{\text{argmax}}[\alpha \log P_{qg}(q|p,a) + \beta \log P_{qa}(a|p,q)]$$

we fix $\beta = 1$. We used an adaptive α value by comparing the synthetic question generated by the QG model and the input question. For each answer entity a, we calculate

$$\alpha = \texttt{max}(1 - \texttt{abs}(\frac{q_{input}}{q_{gen}} - 1), 0.1)$$

This value normalizes the question probability $p(q|p,a)$ estimated by the QG model, since generated questions from some answer entities is easier than other spans in the same passage, which makes the QG model assign all natural questions a relative low perplexity.

C Data

The SQuAD v1.1 is the easiest QA corpus used in this paper. The dataset contains $107,785$ question-answer pairs on 536 articles, which are split into passages. Each question is labeled with an answer that can be extracted from the given passage.

The Natural Questions dataset is a large-scale corpus designed for open-domain question answering. The dataset is more challenging than SQuAD.

Dataset	Num. Synthetic QA
BioASQ	123121
TextbookQA	133773
RACE	115847
RelExt.	52142
DuoRC	250698
DROP	100394

Table 9: Number of synthetic QA of each MRQA domain.

All questions are collected from human search queries and are annotated with long and abstractive answers. Some of the questions are also labeled with a short answer for learning answer-span extraction or reading comprehension. Focusing on the machine reading comprehension task, we select $106,926$ questions labeled with both long and short answers from the dataset for experiments.

For each target domain in MRQA, we collect the corresponding training data and sample 3000 passages for QA synthesis. The number of synthetic QAs varies based on the length of input passages, and is shown in Table 9.

D Answer Entity Recognition Details

In this section, we describe details of the AER methods, which are not covered in detail in previous sections. All AER models are pretrained on the Natural Questions corpus. To solve the sparsity problem, in other words, the passages are long but not all potential question-answer pairs are annotated, we train all following AER models by using the sentence containing the annotated answer entities as inputs, instead of the whole passage. If a sentence in the passage does not contain an annotated answer entity, we do not use it for training.

In this work, we introduce two types of AER methods, tagging based AER (AER-tag) and extraction based AER (AER-Search and AER-Coop). We describe their training and how we use the trained model to recognize answer entities in our experiments.

D.1 AER-Tag

D.1.1 Training

We apply a BIO tagging model for answer entity recognition in the AER-Tab method. We train the model to classify all tokens in the input sentence into three classes,

- B(egin) - the first token of the annotated answer entity
- I(nsize) - other tokens of the annotated answer entity
- O(utside) - tokens that are not a part of the annotated answer entity

D.1.2 Evaluation

Given an input passage, we run the trained BIO tagging model on each of its sentences and greedily predict answer entities. There might be more than one answer entities predicted in each sentence, and we only use the answer entities start with a predicted B tag.

D.2 AER-LM

D.2.1 Training

For AER-LM method, we need to pretrain an extraction-based AER model. We also take a sentence of L tokens containing an annotated answer entity as an example. Using an extraction model, which is similar as our question answering model, we train the model to predict the start and end location of the annotated answer entity. The model outputs a start score and an end score for each token, and predicts the start/end locations by selecting the tokens that are assigned with highest scores. The model is trained with cross-entropy loss, by regarding the extraction task as two L-class classification tasks.

D.2.2 Evaluation

In evalution, we first run the model on each sentence of the input passages and calculate the start and end scores for each token. For each span $(x_i, x_{i+1}, \ldots, x_j)$ that is not longer than L_{span} tokens, we calculate the span score with

$$s_{ij} = s^i_{st} + s^j_{ed} \tag{2}$$

where s^i_{st} is the start score of the first token of span (i, j), and s^j_{ed} is the end score of the last token of the span. In practice, we set $L_{span} = 10$.

To re-rank all possible answer entities, we select top $N_0 = 40$ spans according to s_{ij} for each passage. For all selected answer entities, we generated questions with a pretrained question generator and collect the generation perplexity of the questions. We select $N_{search} = 5$ question-answer pairs with lowest perplexities for the final question-answering finetuning.

D.3 AER-Coop

In AER-Coop, we use the same extraction training method applied in AER-Search, and we also use the s_{ij} scores to select the top $N_0 = 40$ preliminary answer entities for further search. The difference is that we search for final answer entities cooperatively with the pretrained question generator and question answering extractor.

With the question generator and question answering extractor, we re-rank the recognized answer entities with the following score

$$s^c_{ij} = \gamma \cdot I_c - p \tag{3}$$

where γ is a large, positive coefficient, p is the perplexity of generated question based on span (i, j), and $I_c = 1$ if the generated question is correctly answered, and otherwise $I_c = 0$.

D.4 Answer Entity Overlapping

We found the extraction-based AER model leads to overlapping problems, since a large start or end score assigned to a token leads to many candidate answer entities start or end at the token. In practice, if an answer entity is selected by the AER-Search and AER-Coop method, we no longer consider any other answer entities that overlap with the selected ones.

E RGX Examples

In this section, we show some examples of our full model. The examples are contained in Table 10.

The National History Museum of Montevideo is located in the historical residence of General Fructuoso Rivera. It exhibits artifacts related to the history of Uruguay. In a process begun in 1998, the National Museum of Natural History (1837) and the National Museum of Anthropology (1981), merged **in 2001**, becoming the National Museum of Natural History and Anthropology. In July 2009, the two institutions again became independent. The Historical Museum has annexed eight historical houses in the city, five of which are located in the Ciudad Vieja. One of them, on the same block with the main building, is the historic residence of Antonio Montero, which houses the Museo Romantico.
When was the national history museum of montevideo founded?
In the 1920s, John Maynard Keynes prompted a division between microeconomics and macroeconomics. Under Keynesian economics macroeconomic trends can overwhelm economic choices made by individuals. Governments should promote aggregate demand for goods as a means to encourage economic expansion. Following World War II, Milton Friedman created the concept of monetarism. Monetarism focuses on using the **supply and demand of money** as a method for controlling economic activity. In the 1970s, monetarism has adapted into supply-side economics which advocates reducing taxes as a means to increase the amount of money available for economic expansion.
Monarism focuses on the relationship between the?
Starting in 2006, Apple's industrial design shifted to favor aluminum, which was used in the construction of the first MacBook Pro. Glass was added in 2008 with the introduction of the unibody MacBook Pro. These materials are billed as environmentally friendly. The iMac, MacBook Pro, MacBook Air, and Mac Mini lines currently all use aluminum enclosures, and are now made of a single unibody. Chief designer **Jonathan Ive** continues to guide products towards a minimalist and simple feel, including eliminating of replaceable batteries in notebooks. Multi-touch gestures from the iPhone's interface have been applied to the Mac line in the form of touch pads on notebooks and the Magic Mouse and Magic Trackpad for desktops.
Who is the designer of the macbook pro?
The city's total area is 468.9 square miles (1,214 km2). 164.1 sq mi (425 km2) of this is water and 304.8 sq mi (789 km2) is land. The highest point in the city is Todt Hill **on Staten Island**, which, at 409.8 feet (124.9 m) above sea level, is the highest point on the Eastern Seaboard south of Maine. The summit of the ridge is mostly covered in woodlands as part of the Staten Island Greenbelt.
Where is the highest point in new york city?
In 1922, the number of supporters had surpassed 20,000 and by lending money to the club, Barça was able to build the larger Camp de Les Corts, which had an initial capacity of 20,000 spectators. After the Spanish Civil War the club started attracting more members and a larger number of spectators at matches. This led to several expansion projects: the grandstand in 1944, the southern stand in 1946, and finally the northern stand in 1950. After the last expansion, Les Corts could hold **60,000 spectators**.
What is the capacity of barcelona's stadium?
On 1 November 2013, international postal services for Somalia officially resumed. **The Universal Postal Union** is now assisting the Somali Postal Service to develop its capacity, including providing technical assistance and basic mail processing equipment.
Who is responsible for supporting the somali postal service?
In addition to membership, as of 2010[update] there are 1,335 officially registered fan clubs, called penyes, around the world. The fan clubs promote Barcelona in their locality and receive beneficial offers when visiting Barcelona. Among the best supported teams globally, Barcelona has the highest social media following in the world among sports teams, with over 90 million Facebook fans as of February 2016. The club has had many prominent people among its supporters, including **Pope John Paul II**, who was an honorary member, and former prime minister of Spain José Luis Rodríguez Zapatero. FC Barcelona has the second highest average attendance of European football clubs only behind Borussia Dortmund.
Who was an honorary member of barcelona football club?
In April 1758, the British concluded the Anglo-Prussian Convention with Frederick in which they committed to pay him **an annual subsidy of £670,000**. Britain also dispatched 9,000 troops to reinforce Ferdinand's Hanoverian army, the first British troop commitment on the continent and a reversal in the policy of Pitt. Ferdinand had succeeded in driving the French from Hanover and Westphalia and re-captured the port of Emden in March 1758 before crossing the Rhine with his own forces, which caused alarm in France. Despite Ferdinand's victory over the French at the Battle of Krefeld and the brief occupation of Düsseldorf, he was compelled by the successful manoeuvering of larger French forces to withdraw across the Rhine.
What did france pay to the prussian monarchy?
Executives at Trump Entertainment Resorts, whose sole remaining property will be the Trump Taj Mahal, said in 2013 that they were considering the option of selling the Taj and **winding down and exiting the gaming and hotel business**.
What is the future of the trump taj mahal?
Vehicles typically include headlamps and tail lights. Headlamps are white or selective yellow lights placed in the front of the vehicle, designed to illuminate the upcoming road and to make the vehicle more visible. Many manufactures are turning to LED headlights as an energy-efficient alternative to traditional headlamps. Tail and brake lights are red and emit light to the rear so as to reveal the vehicle's direction of travel to following drivers. White rear-facing reversing lamps indicate that the vehicle's transmission has been placed in the reverse gear, warning anyone behind the vehicle that it is moving backwards, or about to do so. Flashing turn signals on the front, side, and rear of the vehicle indicate an intended change of position or direction. In **the late 1950s**, some automakers began to use electroluminescent technology to backlight their cars' speedometers and other gauges or to draw attention to logos or other decorative elements.
When did they start putting back up lights in cars?

Table 10: Examples of recognized answer entities and generated questions with the full RGX model