UnitedQA: A Hybrid Approach for Open Domain Question Answering

Hao Cheng[1]*, Yelong Shen[2]*, Xiaodong Liu[1], Pengcheng He[2],
Weizhu Chen[2], Jianfeng Gao[1]
[1] Microsoft Research [2] Microsoft Azure AI
{chehao,yeshe,xiaodl,penhe,wzchen,jfgao}@microsoft.com

Abstract

To date, most of recent work under the retrieval-reader framework for open-domain QA focuses on either extractive or generative reader exclusively. In this paper, we study a hybrid approach for leveraging the strengths of both models. We apply novel techniques to enhance both extractive and generative readers built upon recent pretrained neural language models, and find that proper training methods can provide large improvements over previous state-of-the-art models. We demonstrate that an hybrid approach by combining answers from both readers can effectively take advantages of extractive and generative answer inference strategies and outperform single models as well as homogeneous ensembles. Our approach outperforms previous state-of-the-art models by 3.3 and 2.7 points in exact match on NaturalQuestions and TriviaQA respectively.

1 Introduction

Open-domain question answering (QA) has been a long standing problem in natural language understanding, information retrieval, and related fields (Chen and Yih, 2020). An typical open-domain QA system follows the retrieval-reader framework (Chen et al., 2017; Guu et al., 2020; Karpukhin et al., 2020), where the relevant passages are first retrieved from a large text corpus, and a reader module then navigates multiple passages for answer inference. In this work, we study two paradigms of reader modules, *i.e. extractive* (Karpukhin et al., 2020; Guu et al., 2020) and *generative* (Lewis et al., 2020; Izacard and Grave, 2021) readers. The extractive reader extracts contiguous spans from the retrieved passages whereas the generative reader sequentially decodes the answer string which might not be contained in the retrieved passages.

Recent work on open-domain QA (Karpukhin et al., 2020; Guu et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021) explores either an extractive reader or a generative reader exclusively. We hypothesize that extractive and generative readers adopt different answer inference strategies, thus a hybrid extractive/generative reader can be a better option for open-domain QA tasks. As shown in Figure 1, compared with prediction agreement among only generative or extractive readers (top-left and bottom-right), the cross prediction agreement between extractive and generative readers (bottom-left) is relatively low (<50%). It indicates that answers produced by those two types of models are different and they can be complementary to each other. Therefore, we propose a hybrid reader approach, UnitedQA, which is a simple ensemble approach to combine the predictions from extractive and generative readers. It achieves state-of-the-art results on NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017).

In UnitedQA, the extractive reader (UnitedQA-E) and generative reader (UnitedQA-G) are built upon the pretrained language models, ELECTRA (Clark et al., 2020) and T5 (Raffel et al., 2020), respectively. For the UnitedQA-E, we adopt a weakly-supervised training objective to address the noisy supervision issue caused by the heuristics-based labeling and incorporate the posterior differential regularization (PDR) (Cheng et al., 2021) to improve the model robustness. The UnitedQA-G follows the T5 Fusion-in-Decoder (FID) (Izacard and Grave, 2021) and we make two improvements: first, we add a group of attention bias parameters into the decoder cross-attention block to feature the ranking information of retrieved contexts; second, we add the adversarial training (Ju et al., 2019; Jiang et al., 2020; Pereira et al., 2021) to improve the model generalization ability.

The experimental results highlight the effec-

[*] Equal Contribution

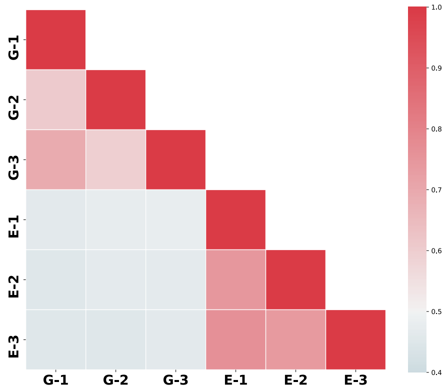

Figure 1: Pairwise prediction agreement ratio. G-1, G-2, G-3 and E-1, E-2, E-3 are three different generative and extractive readers respectively. All readers achieve similar performance ($\approx 52\%$ exact match) on NaturalQuestions. Higher agreement (>50%) in red and lower agreement (<50%) in gray. The agreement is calculated based on exact string match.

tiveness of the simple hybrid approach of UnitedQA. With both improved extractive and generative readers, UnitedQA sets new state-of-the-art results on two popular open-domain QA datasets, *i.e.* 54.7 and 70.3 in exact match on NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), respectively. It is worth noting that our UnitedQA model not only outperforms each single model but also brings more pronounced improvements over homogeneous ensembles of either extractive or generative readers. Last, based on our analyses, UnitedQA-E and UnitedQA-G have advantages in different cases, suggesting they may use different reasoning strategies.

2 Method

In this section, we present the overall pipeline of the UnitedQA system, which consists of three components: **Retrieval**, **Reading**, and **Re-ranking**. First, the retrieval module fetches a list of relevant passages from a Wikipedia dump for a given question. Then, the module of hybrid readers produces answer candidates from the set of retrieved passages. Last, the re-ranking module combines the answer candidates with linear interpolation and produce the final answer.

Retrieval Following Karpukhin et al. (2020), we consider two methods, BM25 and dense passage retrieval (DPR), for retrieving the support passages for a given question. For BM25, passages are encoded as bag of words (BOW), and inverse document frequencies are used as the ranking function. For DPR, passages and questions are represented as dense vectors based on two BERT (Devlin et al., 2019) models. The relevance score is then computed based on the dot production between the query and passage vectors. In this paper, we adopt the same implementation as Karpukhin et al. (2020) for retrieving passages. Specifically, the English Wikipedia dump from Dec. 20, 2018 is used as the source documents for retrieval, with the removal of semi-structured data, such as tables or lists. Each document is split into disjoint 100-word *passages* as the basic retrieval unit. The top-100 passages are then passed for reading.

Reading We combine the generative reader and the extractive reader to produce answer candidates over the retrieved passages. Here, we only give a high-level description of our approach. More details regarding our improved extractive and generative models are presented in §2.1 and §2.2 respectively.

The generative reader is based on a sequence-to-sequence model pre-trained in a forward-generation fashion on a large corpus, *i.e.* T5 (Raffel et al., 2020). Similar to Izacard and Grave (2021), the model takes the question and its relevant passages as input, and then generates the answer string token by token. Specifically, the concatenation of all retrieved passages and the corresponding question is used as the encoder input. Then, the decoder performs reasoning over the concatenation of all evidence through an attention mechanism.

Following state-of-the-art extractive QA models (Devlin et al., 2019; Karpukhin et al., 2020), our extractive reader is based on a Transformer neural network pre-trained with a cloze style self-supervised objective, *i.e.* ELECTRA (Clark et al., 2020). Here, a pair of a given question and a support passage is jointly encoded into neural text representations. These representations are then used to define scores or probabilities of possible answer begin and end positions, which are in turn used to define probabilities over possible answer spans. Finally, the answer string probabilities are based on the aggregation over all possible answer spans from the entire set of support passages.

2.1 UnitedQA-E

In §2.1.2, we give the problem definition of open-domain QA for extractive reader. Then, we detail the improvements of UnitedQA-E in §2.1.2.

2.1.1 Extractive Reader

Given a question q and a set of K retrieved passages $\mathrm{p}_1, \ldots, \mathrm{p}_K$, a text encoder produces contextualized representations: $\mathbf{h}_1^k, \ldots \mathbf{h}_T^k \in \mathbb{R}^n$ for the question-passage pair $(\mathrm{q}, \mathrm{p}_k)$ in the form of "`[CLS]`*question* `[SEP]`*passage* `[SEP]`", where `[CLS]` and `[SEP]` are special tokens for encoding inputs, T is the maximum sequence length of the input text, and $\mathbf{h}_i^k$ indicates the contextualized embedding of the i-th token in $(\mathrm{q}, \mathrm{p}_k)$.

The extractive reader computes the span-begin score of the i-th token as $s_b(i^k) = \mathbf{w}_b^T \mathbf{h}_i^k$ using a weight vector $\mathbf{w}_b \in \mathbb{R}^d$. The span-end score $s_e(j^k)$ is defined in the same way. Thus, the probabilities of a start position i^k and an end position j^k are $P_b(i^k) = \frac{\exp(s_b(i^k))}{Z_b}, P_e(j^k) = \frac{\exp(s_e(j^k))}{Z_e}$, where Z_b, Z_e are normalizing factors defined by the corresponding probability space. The probability of an answer span from i^k to j^k is defined as $P_s(i^k, j^k) = P_b(i^k) P_e(j^k)$.

Here, we consider two probability spaces, **passage level** and **multi-passage level**, with the only difference in the computing of Z_b, Z_e. Specifically, the passage-level probability of each answer begin and end is computed by normalizing all possible positions in the respective passage, *i.e.* $Z_b = Z_b^k = \sum_{\mathcal{I}^k \cup \text{NULL}} \exp(s_b(i)), Z_e = Z_e^k = \sum_{\mathcal{I}^k \cup \text{NULL}} \exp(s_e(j))$, where $\mathcal{I}^k$ is the set of all possible positions from the k-th passage and NULL indicates special positions if p_k does not support answering the question. Similarly, the multi-passage level probability is computed by normalizing over each answer positions across all K relevant passages, *i.e.* $Z_b = Z_b^* = \sum_k \sum_{\mathcal{I}^k} \exp(s_b(i)), Z_e = Z_e^* = \sum_k \sum_{\mathcal{I}^k} \exp(s_e(j))$, respectively.

Since there are usually multiple plausible mentions for open-domain QA, during training, it is typical to maximize either the marginal log-likelihood (MML) of all correct spans (Karpukhin et al., 2020) or the log-likelihood of the most likely correct span (HardEM) (Min et al., 2019). During inference, the prediction is made based on the candidate answer string score, obtaining as $P_a(y) = \sum_{(i,j) \in \mathcal{Y}} P_s(i, j)$, where $\mathcal{Y}$ is the set of spans corresponding to the answer string y.

2.1.2 Improvement Method

In addition to better text representations from Clark et al. (2020), we consider two methods for improving the training of the extractive reader.

Multi-objective for Weakly-supervised QA The multi-objective formulation is introduced in Cheng et al. (2020) for improving weakly supervised document-level QA. Different from Cheng et al. (2020) where only MML is considered for the multi-objective formulation, we found combining HardEM with MML is more effective for open-domain QA based on our experiments (§4.1). Specifically, we combine a multi-passage HardEM loss with K passage-level MML losses over a batch of K passages

$$\begin{aligned} \mathcal{L}_{\text{EXT}} &= \log \max_{(i,j)} P_s^M(i, j) + \\ &\quad \frac{1}{K} \sum_k \log \sum_{(i^k, j^k)} P_s^P(i^k, j^k), \quad (1) \end{aligned}$$

where P_s^M, P_s^P is the multi-passage level and passage level span probabilities respectively.

Posterior Differential Regularization Due to the noisy supervision for open-domain QA (Chen et al., 2017), we investigate the posterior differential regularization (PDR) (Cheng et al., 2021) to improve the robustness of the extractive reader. Different from Cheng et al. (2021) where only clean supervision setting is considered, in this work, we apply PDR to the weakly supervised open-domain QA scenario. Given it is computationally expensive to enumerate all possible spans, we apply two separate regularization terms for the begin and end probabilities at the multi-passage level, respectively,

$$\mathcal{L}_{\text{PDR}} = D(P_b(i) | P_b'(i)) + D(P_e(j) | P_e'(j)), \quad (2)$$

where $D(\cdot|\cdot)$ is the squared Hellinger distance, and P_b', P_e' are the probabilities of start and end positions with additive input noise to the token embeddings. Specifically, we sample noise vectors $\epsilon_1, \ldots, \epsilon_T$ from $\mathcal{N}(0, c^2 I)$, and add them to the token embeddings as the noisy input, *i.e.* $\mathbf{v}_1 + \epsilon_1, \ldots, \mathbf{v}_T + \epsilon_T$, where c is fixed to 1e$-$3 throughout our experiments.

Based on this, the overall training objective for the extractive reader is

$$\mathcal{L}^1 = \mathcal{L}_{\text{EXT}} + \gamma \mathcal{L}_{\text{PDR}}, \quad (3)$$

where γ is a regularization scalar hyperparameter.

2.2 UnitedQA-G

Here, we first formally define the setup of generative reader for open-domain QA in § 2.2.1 and then present our improvements in § 2.2.2.

2.2.1 Generative Reader

Given a question q and a set of K retrieved passages $\mathrm{p}_1, \ldots, \mathrm{p}_K$, the encoder model encodes each $(\mathrm{q}, \mathrm{p}_k)$ pair independently, and produces contextualized representation for each token: $\mathbf{h}_i^k \in \mathbb{R}^d$ for the i-th token of the k-th pair. The decoder then performs attention over the concatenation of the representations of all the retrieved passages, and generates the answer string.

Let $\mathbf{x}$ denote the input of the question and all retrieved passages $\mathbf{x} = \big((\mathrm{q}, \mathrm{p}_1), ..., (\mathrm{q}, \mathrm{p}_K)\big)$, and $\mathbf{y}$ the answer string with its tokens as $(y_1, ..., y_N)$. The generative reader is trained to maximize a sequence-to-sequence objective for a given $(\mathbf{x}, \mathbf{y})$,

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta) = \sum_{i}^{N} \log P_\theta(y_i | \mathbf{x}, y_{1:i-1}), \qquad (4)$$

where θ is the model parameter. During inference, a greedy decoding is used to produce the answer.

2.2.2 Improvement Method

Decoder Attention Bias The decoder in the T5 transformer model adopts a cross-attention mechanism to compute attention scores between the decoding answer tokens and all the retrieved passage tokens. Specifically, let $\mathbf{y}_i \in \mathbb{R}^d$ be the query vector of the i-th decoding token[1], and $\mathbf{m}_j^k \in \mathbb{R}^d$ be the key vector of the j-th token in $((q), \mathrm{p}_k)$. The multi-head cross-attention scores in T5 (Raffel et al., 2020) $\mathbf{s}_{i,j}^k$ is calculated as

$$\mathbf{s}_{i,j}^k = \text{MultiHeadAtt}(\mathbf{y}_i, \mathbf{m}_j^k) \in \mathbb{R}^{|\text{Head}|} \qquad (5)$$

where $|\text{Head}|$ is the number of attention heads. However, it doesn't capture the relevance information of retrieved passages into the reader in (5). To add the relevance feature into the attention block, we revise (5) by incorporating the attention bias

$$\mathbf{s}_{i,j}^k = \text{MultiHeadAtt}(\mathbf{y}_i, \mathbf{m}_j^k) + \mathbf{b}_k, \qquad (6)$$

where $\mathbf{b}_k \in \mathbb{R}^{|\text{Head}|}$ is a trainable attention bias vector for all the tokens in the k-th retrieved passage. In the experiments, the maximum retrieved passages is by default set to 100. Thus, the decoder attention bias introduces additional $100 * |\text{Head}|$ parameters for each layer.

Dataset	Train	Dev	Test
NQ	79168	8757	3610
TriviaQA	78785	8837	11313
EffcientQA	-	1800	-

Table 1: Number of questions in each QA dataset.

Adversarial Training Adversarial training creates *adversarial* examples by adding small perturbations to the embedding layer. Assuming the word(-piece) embedding layer is parameterized by a matrix $\mathbf{V} \in \mathcal{R}^{|V| \times d}$, $|V|$ is the vocabulary size, and d is the embed-dimension. The adversarial embedding matrix $\hat{\mathbf{V}}$ can be obtained by

$$g_{\mathbf{V}} = -\nabla_{\mathbf{V}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta), \qquad (7)$$

$$\hat{\mathbf{V}} = \mathbf{V} + \text{SG}(\epsilon g_{\mathbf{V}} / ||g_{\mathbf{V}}||_2), \qquad (8)$$

where $\text{SG}(\cdot)$ is the stop-gradient operation. We use the adversarial embedding matrix $\hat{\mathbf{V}}$ to replace the original $\mathbf{V}$ in model parameters θ, and obtain $\hat{\theta}$. Thus the adversarial loss can be calculated as

$$\mathcal{L}_{\text{AT}}(\mathbf{x}, \mathbf{y}; \theta) = \mathcal{L}(\mathbf{x}, \mathbf{y}; \hat{\theta}). \qquad (9)$$

Therefore, the overall training objective of the generative reader is

$$\mathcal{L}^2 = \alpha \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta) + \beta \mathcal{L}_{\text{AT}}(\mathbf{x}, \mathbf{y}; \theta), \qquad (10)$$

where $\alpha = 0.5, \beta = 0.5$ in all of the exepriments.

2.3 UnitedQA System

The UnitedQA system combines outputs from both extractive and generative models for a given question during inference. Since the output spaces of extractive and generative models are different, we use a simple linear interpolation based on best predictions from each model[2]. Denote the predicted strings from M extractive and N generative models as $y_1^E, ..., y_M^E$ and $y_1^G, ..., y_N^G$, respectively. The hybrid prediction y^* is obtained by

$$\operatorname*{argmax}_{y \in \mathcal{Y}} \tau \sum_{m=1}^{M} \mathbf{1}(y, y_m^E) + \delta \sum_{n=1}^{N} \mathbf{1}(y, y_n^G), \quad (11)$$

where $\mathcal{Y}$ is the set of all predicted strings, $\mathbf{1}(y, y')$ is an indicator function and $\tau = 0.6, \delta = 0.4$.

3 Experiments

3.1 Experiment Setup

We use two representative QA datasets and adopt the same training/dev/testing splits as in previous

[1] we omit the layer notation for simplification

[2] We have also tried a few more complex approaches for combining the extractive and generative models. For example, we first train an extractive model, and then append the top-k answer strings from the extractive model at the end of the input for training a generative model. None of them is as good as the simple ensemble approach.

Model	Reader Type	Reader Size (M)	NQ	TriviaQA
REALM(Guu et al., 2020)	Extractive	110	40.4	N/A
RAG(Lewis et al., 2020)	Generative	400	44.5	56.1
DPR(Karpukhin et al., 2020)	Extractive	110	41.5	57.9
T5-FID$_{\text{base}}$(Izacard and Grave, 2021)	Generative	220	48.2	65.0
T5-FID$_{\text{large}}$(Izacard and Grave, 2021)	Generative	770	51.4	67.6
UnitedQA-E$_{\text{base}}$ (Ours)	Extractive	110	47.7	66.3
UnitedQA-E$_{\text{large}}$ (Ours)	Extractive	330	51.8	**68.9**
UnitedQA-G$_{\text{large}}$(Ours)	Generative	770	**52.3**	68.6
UnitedQA-E$_{\text{large}}$++ (Ours)	Ensemble	3x330	52.4	69.6
UnitedQA-G$_{\text{large}}$++ (Ours)	Ensemble	3x770	53.3	69.2
UnitedQA (Ours)	Hybrid	2x770+330	**54.7**	**70.5**

Table 2: Comparison to state-of-the-art models on the test sets of NaturualQuestions (NQ) and TriviaQA. Exact match score is used for evaluation. The overall best model is in ▨, the best single model is in **bold**, and the best model with the smallest reader size is in underline.

work (Lee et al., 2019; Karpukhin et al., 2020). Both datasets (see Table 1 for statistics) have been heavily studied in recent work (Lee et al., 2019; Min et al., 2019; Karpukhin et al., 2020; Guu et al., 2020). We follow the standard evaluation protocol and use exact match (EM) as the evaluation metric.

NaturalQuestions (Kwiatkowski et al., 2019) is composed of questions by real users to Google Search, each with answers identified by human annotators in Wikipedia. The open-domain version of NaturalQuestions (Lee et al., 2019) only consider questions with short answers, *i.e.* answers with less than 5 tokens. In the NaturualQuestions, the questions are considered to be more information seeking given that the question askers didn't know the answer beforehand. In addition, we use another evaluation set, *i.e.* the dev set introduced recently by the EfficientQA competition (Min et al., 2021), which is constructed in the same way as the original NaturalQuestions dataset.

TriviaQA (Joshi et al., 2017) contains trivia question-answer pairs that were scraped from the web. Different from NaturalQuestions, the questions here are written with known answers in mind. Specifically, the unfiltered set has been used for developing open-domain QA models.

Implementation details For a fair comparison, we use the same retrieval module as Karpukhin et al. (2020) for NaturalQuestions and TriviaQA to mitigate the impact of retrieval difference. Specifically, we use DPR (single) for NaturalQuestions and BM25+DPR (multi) for TriviaQA because of their best end-to-end performance (Karpukhin et al. 2020). For all the experiments, we use 8 and 16 V100-32GB for base and large model training respectively. We train our models with Adam optimizer of a linear scheduler with a warmup raito of 0.1. The extractive models are trained for up to 8 epochs with a learning rate of 2e$-$5 and a batch passage size per question of 16. The generative models are trained for up to 10 epochs with a learning rate of 1e$-$4, a batch size of 64, and 100 retrieved passages per question for model training. We select γ in $\{4, 8\}$. After the best configuration is selected based on the dev set, we run our best models 3 times independently with different random seeds and report the median performance on the test set. We also report ensemble results which are based on the linear interpolation over answer predictions from the 3 models.

3.2 Main results

Single Model Results: We first compare our models to two recent models, REALM (Guu et al., 2020) and RAG (Lewis et al., 2020), which are first pre-trained with different retrieval augmented objectives and then fine-tuned for open-domain QA. In addition, we include as baselines DPR (Karpukhin et al., 2020) and T5-FID (Izacard and Grave, 2021), both of which are based on the same retriever as ours. As shown in Table 2, both our extractive and generative models achieve new state-of-the-art results for both studied datasets. Compared with the recent state-of-the-art extractive model (DPR), our base model leads to pronounced 15% relative improvements for both NaturalQuestions (+6.2 absolute improvement) and TriviaQA (+8.4 absolute improvement). More importantly, UnitedQA-E$_{\text{base}}$ achieves comparable or even better performance with regard to generative models of larger size, *i.e.* RAG and T5-FID$_{\text{base}}$. It highlights the importance of proper training strategies for open-domain QA models.

Hybrid Model Results: In order to evaluate the advantage of the hybrid of the extractive and generative models (UnitedQA), we include two homogeneous ensemble baselines, one consisting of only extractive readers (UnitedQA-E++) and the other ensemble of exclusively generative models (UnitedQA-G++). For homogeneous ensemble cases, the three-way majority prediction is used. For the hybrid of extractive and generative readers, we select a three-model combination from the set of three generative and three extractive models based on the dev set. We observed that combining predictions from two generative models and one extractive model results in the best hybrid model for both datasets. As expected, all ensemble models show an improvement over their single model counterparts. However, the two homogeneous ensemble baselines, UnitedQA-E++ and UnitedQA-G++, only provide marginal gains over the corresponding best single models. The significant improvement brought by our proposed hybrid approach indicates the benefit of combining extractive and generative readers for open-domain QA.

Discussion: Although the proposed hybrid approach has been shown to be highly effective for open-domain QA, we point out that the improved performance comes with increased computational cost. The best combination requires approximately three times the computational cost of a single generative model. Therefore, it would be interesting to explore more efficient hybrid methods, such as effective parameter sharing strategies or unified formulations. Another interesting future direction is to explore customized compression approaches for reducing the model size of retriever and reader separately or jointly through pruning (Han et al., 2016), quantization (Hubara et al., 2018), and knowledge distillation (Hinton et al., 2015). Specifically, given that the hybrid model is more effective, it is likely that a student model can learn more effectively from a hybrid teacher model via knowledge distillation for open-domain QA.

Model	NQ	TriviaQA
(Cheng et al., 2020) +`PDR`	43.3	60.1
BERT$_{\text{base}}$	44.2	62.2
`-Multi-obj`	43.5	61.3
`-PDR`	41.8	60.2
`-Multi-obj & PDR`	40.6	58.5
UnitedQA-E$_{\text{base}}$	46.0	65.4
`-Multi-obj`	45.2	64.3
`-PDR`	43.1	63.8
`-Multi-obj & PDR`	42.5	61.2

Table 3: Ablation experiments of the extractive model on the dev sets of NaturalQuestions (NQ) and TriviaQA. Exact match score is reported. The top and bottom models are built on BERT$_{\text{base}}$ and ELECTRA$_{\text{base}}$, respectively.

4 Analysis

In this section, we first carry out ablation study on the extractive and generative model improvements. Moreover, we aim to take a deeper look and understand the difference between the two models.

4.1 Ablation Study

In Table 3, we present ablation experiments on the effectiveness of different textual representations and methods for improving the extractive model UnitedQA-E$_{\text{base}}$. Here, we focus on base models, *i.e.* BERT$_{\text{base}}$ and ELECTRA$_{\text{base}}$. Note that the row `UnitedQA-E`$_{\texttt{base}}$ is the corresponding base model reported in Table 2. Compared with the MML-based multi-objective (Cheng et al., 2020), we find that a new multi-objective with HardEM at the multi-passage level and MML at the passage level is more effective for open-domain QA. In addition to the multi-objective training, there is a noticeable improvement brought by the regularization method (PDR) which indicates the importance of proper regularization for learning with noisy supervision. Last but not least, the large improvement of ELECTRA over BERT indicates the importance of deriving better text representations for weakly supervised NLP problems. For the UnitedQA-G, we present the ablation study on analyzing the effectiveness of decoder attention bias component and adversarial training mechanism in Table 4. Both techniques contribute to decent improvements over T5-FID with more pronounced gains brought by adversarial training.

Model	NQ	TriviaQA
T5-FID$_{\text{large}}$	51.4	67.6
UnitiedQA-G$_{\text{large}}$	52.3	68.6
`-Adv Training`	52.0	68.2
`-Attention Bias`	51.8	68.1

Table 4: Ablation experiments of the generative model on the test sets of NaturalQuestions (NQ) and TriviaQA. Exact match score is reported.

		Top-20	Top-100	Δ
NQ	Retrieval	78.4	85.4	+9%
	United-E	49.8	51.8	+4%
	United-G	49.3	52.3	+6%
TriviaQA	Retrieval	79.9	84.4	+6%
	United-E	67.1	68.9	+3%
	United-G	65.4	68.6	+5%

Table 5: Retieval top-k accuracy and end-to-end QA extact match scores on the test sets of NaturalQuestions (NQ) and TriviaQA. United-E and United-G stand for our extractive and generative models respectively.

4.2 Impact of Retrieval Accuracy

Here, we vary the number of retrieved passages during inference and report the evaluation results in terms of end-to-end QA exact match score of UnitedQA-E and UnitedQA-G along with the corresponding top-k retrieval accuracy. The results are summarized in Table 5. As expected, when the number of retrieved passages increases, both top-k retrieval accuracy and the end-to-end QA performance improve. However, there is a noticeable gap between the improvement of retrieving more passages (i.e., recall) and that of the corresponding end-to-end QA performance, especially for the extractive reader. This is likely caused by additional noise introduced with improved retrieval recall. Specifically, only half of the retriever improvement can be effectively utilized by the extractive model while the generative model can benefit more from retrieving more passages. This suggests that by concatenating all passages in vector space, the generative model are more effective in de-noising in comparison to the extractive model.

4.3 Breakdown Evaluation

Following Lewis et al. (2021), we carry out a breakdown evaluation of model performance over the NaturalQuestions and TriviaQA test sets. Given their superior performance, we again only consider our improved extractive and generative models, *i.e.* UnitedQA-E$_{\text{large}}$ and UnitedQA-G respectively. The evaluation is summarized in Table 6. In comparison to their corresponding overall performance, both the extractive and generative models achieve much better performance on the "Overlap" categories (*i.e.* "Question Overlap" and "Answer Overlap") for both NaturalQuestions and TrivaQA, which indicates that both models perform well for question and answer memorization. Different from question and answer memorization, there is a pronounced performance drop for both models on the"Answer Overlap Only" category where certain amount of relevance inference capability is required to succeed. Lastly, we see that both extractive and generative models suffer some significant performance degradation for the "No Overlap" column which highlights model's generalization evaluation. Nevertheless, the extractive model demonstrate a better QA generalization by achieving a better overall performance on the "No Overlap" category for both datasets.

4.4 Error Analysis

Here, we conduct analyses into prediction errors made by the extractive and generative models based on automatic evaluation. For this study, we use the EfficientQA dev set (Min et al., 2021) which is constructed in the same way as the original NaturalQuestions dataset. Specifically, we group prediction errors into three categorizes: 1) common prediction errors made by both the extractive and generative models, 2) prediction errors made by the extractive model, 3) prediction errors produced by the generative model. In the following, we first carry out a manual inspection into the common errors. Then, we compare the prediction errors made by extractive and generative models, respectively.

First of all, there is an error rate of 29% of those consensus predictions made by both extractive and generative models according to the automatic evaluation. Based on 30 randomly selected examples, we find that around 30% of those predictions are actually valid answers as shown in the top part of Table 7. In addition to predictions that are answers at different granularity or semantically equivalent ones, some of those prediction errors are likely caused by the ambiguity in questions. As the given example in Table 7, based on the specificity, the model prediction is also a valid answer. This high-

Dataset	Model	Total	Question Overlap	No Question Overlap	Answer Overlap	Answer Overlap Only	No Overlap
NQ	UnitedQA-G	**52.3**	**72.2**	40.5	**62.7**	**45.4**	34.0
	UnitedQA-E	51.8	69.4	**41.5**	60.1	45.1	**37.6**
TriviaQA	UnitedQA-G	68.6	88.4	62.5	78.1	69.6	**44.5**
	UnitedQA-E	**68.9**	**89.3**	**62.7**	**78.6**	**70.6**	44.3

Table 6: Breakdown evaluation on NaturalQuestions (NQ) and TriviaQA based on test splits defined in (Lewis et al., 2021). Exact match scores are reported. UnitedQA-E and UnitedQA-G denote our extractive and generative models respectively.

Valid Answers	
Different granularity	Q: When was harry potter and the deathly hallows part 2 movie released Prediction: 2011 / Gold: 15 July 2011
Semantically equivalent	Q: minimum age limit for chief justic of india Prediction: 65 / Gold: 65 years
Ambiguity question	Q: who won her first tennis grand slam in 2018 Prediction: Carolin Wozniacki / Gold: Simona Halep
Wrong Answers	
Part as whole error	Q: the official U.S. poverty line is based on the cost of what Prediction: food / Gold: ICP purchasing power
Entity confusion	Q: actor who played tommy in terms of endearment Prediction: Jeff Daniels / Gold: Troy Bishop
Event confusion	Q: when did the saskatchewan roughriders last won the grey cup Prediction: 2007 / Gold: 2013

Table 7: Examples of prediction errors as judged by the automatic evaluation.

lights the limitation of the current evaluation metric, which does not accurately estimate the existing open-domain QA system capabilities. As shown in the bottom part of Table 7, most of representative errors are due to the confusion of related concepts, entities or events that are mentioned frequently together with the corresponding gold answers.

Next, all questions from the dev set are categorized based the *WH* question word, *i.e. what, which, when, who, how, where*. We then report the relative performance change of each *WH* category for both extractive and generative models over their corresponding overall prediction accuracy in Figure 2. First, it is easy to see that both extractive and generative models achieve the best performance for entity related *who* questions, which is likely to be the result of high ratio of samples of this type seen during training. In contrast, the answers to *what* questions can play a much richer syntactic role in context, making it more difficult for both extractive and generative models to perform well. Interestingly, the generative model exhibits the strength for temporal reasoning, whereas the extractive model does not. This difference suggests that it is worth exploring better temporal modeling strategies to improve the extractive model in the future.

5 Related Work

Open-domain QA Open-domain QA requires a system to answer questions based on evidence retrieved from a large corpus such as Wikipedia (Voorhees, 2000; Chen et al., 2017). Recent progress has been made towards improving evidence retrieval through both sparse vector models like TF-IDF or BM25 (Chen et al., 2017; Min et al., 2019), and dense vector models based on BERT (Lee et al., 2019; Karpukhin et al., 2020; Guu et al., 2020; Qu et al., 2021). Generally, the dense representations complement the sparse vector methods for passage retrieval as they can potentially give

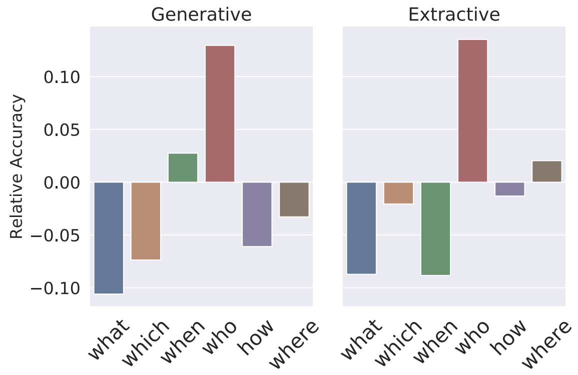

Figure 2: Relative accuracy of different *WH* questions. The relative accuracy is the relative change of a *WH* category accuracy to the overall model accuracy.

high similarity to semantically related text pairs, even without exact lexical overlap. Unlike most work focusing on a pipeline model, Lee et al. (2019) propose a pre-training objective for jointly training both the retrieval encoder and reader. It is further extended by Guu et al. (2020) with a dynamic update of the passage index during the training. Instead, in this work, we focus on a hybrid reader approach for open-domain QA. By simply combing answer predictions from extractive and generative models, our UnitedQA achieves significant improvements over state-of-the-art models.

Reading Comprehension with Noisy Labels There has been a line of work on improving distantly-supervised reading comprehension models by developing learning methods and model architectures that can better use noisy labels. Most of them focus on the document-level QA, where all paragraphs share the same document context. Clark and Gardner (2018) propose a paragraph-pair ranking objective for learning with multiple paragraphs so that the model can distinguish relevant paragraphs from irrelevant ones. In (Lin et al., 2018), a coarse-to-fine model is proposed to handle label noise by aggregating information from relevant paragraphs and then extracting answers from selected ones. Min et al. (2019) propose a hard EM learning scheme where only passage-level loss is considered for document-level QA. More recently, different probabilistic assumptions with corresponding training and inference methods are examined in (Cheng et al., 2020) again for document-level QA with distant supervision. In our work, we further extend the multi-objective formulation proposed in (Cheng et al., 2020) with the hard EM learning (Min et al., 2019) for enhancing extractive open-domain QA, where the input passages are given by a retrieval model and are typically from different documents.

6 Conclusion

In this study, we propose a hybrid model for open-domain QA, called UnitedQA, which combines the strengths of extractive and generative readers. We demonstrate the effectiveness of UnitedQA on two popular open-domain QA benchmarks, NaturalQuestions and TriviaQA. Our results show that the proposed UnitedQA model significantly outperforms single extractive and generative models as well as their corresponding homogeneous ensembles, and sets new state-of-the-art on both benchmarks. We also perform a comprehensive empirical study to investigate the relative contributions of different components of our model and the techniques we use to improve the readers.

For future work, it would be interesting to explore model compression approaches for reducing the model size of retriever and reader separately or jointly through pruning, quantization, and knowledge distillation.

Acknowledgments

We would like to thank the anonymous reviewers for valuable suggestions, Yuning Mao for valuable discussions and comments, and Microsoft Research Technology Engineering team for computing support.

References

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

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online. Association for Computational Linguistics.

Hao Cheng, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Probabilistic assumptions matter: Improved models for distantly-supervised document-level question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5657–5667, Online. Association for Computational Linguistics.

Hao Cheng, Xiaodong Liu, Lis Pereira, Yaoliang Yu, and Jianfeng Gao. 2021. Posterior differential regularization with f-divergence for improving model robustness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1078–1089, Online. Association for Computational Linguistics.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2018. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. NeurIPS 2020 EfficientQA competition: Systems, analyses and lessons learned.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Lis Pereira, Xiaodong Liu, Hao Cheng, Hoifung Poon, Jianfeng Gao, and Ichiro Kobayashi. 2021. Targeted adversarial training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5385–5393, Online. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ellen Voorhees. 2000. The TREC-8 question answering track report.