Multi-Task Dense Retrieval via Model Uncertainty Fusion for Open-Domain Question Answering

Minghan Li[1,2]*, **Ming Li**[1,2], **Kun Xiong**[1,2], and **Jimmy Lin**[1,2]

[1]David R. Cheriton School of Computer Science, University of Waterloo
[2]RSVP.ai

{m692li, mli, k5xiong, jimmylin}@uwaterloo.ca

Abstract

Multi-task dense retrieval models can be used to retrieve documents from a common corpus (e.g., Wikipedia) for different open-domain question-answering (QA) tasks. However, Karpukhin et al. (2020) shows that jointly learning different QA tasks with one dense model is not always beneficial due to corpus inconsistency. For example, SQuAD only focuses on a small set of Wikipedia articles while datasets like NQ and Trivia cover more entries, and joint training on their union can cause performance degradation. To solve this problem, we propose to train individual dense passage retrievers (DPR) for different tasks and aggregate their predictions during test time, where we use uncertainty estimation as weights to indicate how probable a specific query belongs to each expert's expertise. Our method reaches state-of-the-art performance on 5 benchmark QA datasets, with up to 10% improvement in top-100 accuracy compared to a joint-training multi-task DPR on SQuAD. We also show that our method handles corpus inconsistency better than the joint-training DPR on a mixed subset of different QA datasets. Code and data are available at https://github.com/alexlimh/DPR_MUF.

1 Introduction

Open-domain question-answering requires finding answers to given questions from a large collection of documents (Voorhees and Tice, 2000). Therefore, a first-stage retrieval component that selects a set of potentially answer-containing documents is often involved for the second-stage reading comprehension model (Chen et al., 2017). Traditional term-matching methods such as tf–idf and BM25 (Robertson and Zaragoza, 2009; Lin et al., 2021) that leverage an inverted index to construct sparse textual representations have built strong baselines in the first-stage retrieval.

Items	Joint Training	Model Fusion
Task Flexibility		✓
Training Speed		✓
Inference Speed	✓	
Storage Space	✓	

Table 1: Comparisons between two multi-task solutions. Joint training: A single model trained on the union of multiple datasets. Model fusion: Independent experts trained on different datasets. "✓" means more advantageous compared to the other method.

Recently, neural-based dense retrievers (Seo et al., 2019; Lee et al., 2019; Guu et al., 2020; Karpukhin et al., 2020) have been shown to achieve better performance in open-domain question-answering, but they often fail to generalize outside of the training data distribution. A standard solution known as *joint training* that learns a single dense retriever on the union of different datasets (Maillard et al., 2021; Wang et al., 2021) provides a solution to a certain extent. However, Karpukhin et al. (2020) has shown that data from different tasks might have conflicts with each other, where joint training on their union can cause performance degradation. For example, SQuAD (Rajpurkar et al., 2016) only focuses on a small set of Wikipedia documents while datasets like NQ (Kwiatkowski et al., 2019) and Trivia (Joshi et al., 2017) cover more entries. Therefore, careful data re-balancing and hyperparameter search are required during training.

In this paper, we propose another solution to multi-task learning, which trains multiple DPR experts on different datasets separately and their predictions are aggregated during test time. This is also known as *model fusion* (Hoang et al., 2019) which differs from a *mixture of experts* (Shazeer et al., 2017) as it does not need to learn a gating function on the joint dataset. The model fusion method is easier to incorporate new data for continual learn-

* Correspondence to: Minghan Li <alexlimh23@gmail.com>

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 274–287
November 7–11, 2021.

ing without introducing conflicts, as each expert trains on independent tasks. In addition, these experts can be trained in parallel to speed up the learning process.

However, the challenge now becomes how to aggregate different expert's predictions without hurting their in-distribution performance. We propose model uncertainty estimation (Loquercio et al., 2020) as a dynamic weighting scheme, which helps the expert to identify whether a question belongs to its expertise.

Intuitively, a model that overfits to a training distribution should be more uncertain about the out-of-domain data than the in-domain data. For example, the question *"How many episodes in Season 2 Breaking Bad?"* might get a high uncertainty score from an expert trained on a medical QA dataset.

In practice, we leverage ensemble uncertainty where we train an ensemble of small neural networks for each pre-trained DPR expert (Lakshminarayanan et al., 2017). Specifically, we represent the model uncertainty as the mutual information (Poole et al., 2019) between the ensemble's predictions and parameters. For each question, we retrieve a set of the top-k documents using different DPR experts and then use its corresponding ensemble to compute the uncertainty score of the question. Finally, we aggregate all the expert's predictions into a normalized weighted sum and rerank the retrieved documents. Fig. 1 demonstrates a simplified pipeline of our algorithm and Tbl. 1 compares the differences between the joint training and model fusion solutions.

Extensive experiments show that our final fusion model not only outperforms individual specialists on 5 open-domain QA datasets but also outperforms the performance of the joint-training, multi-task DPR model, with up to 10% improvement in top-100 accuracy on SQuAD. Finally, our method manages to handle corpus conflicts on a mixed subset of different QA tasks, which even outperforms an oracle model using Bayesian optimization (Frazier, 2018).

2 Related Work

Retrieval and QA Traditional retrieval methods such as tf-idf or BM25 generate sparse, high-dimensional vectors (Robertson and Zaragoza, 2009; Lin et al., 2021) and have been proven effective in various QA tasks (Chen et al., 2017; Yang et al., 2019; Min et al., 2019). Recently, neural retrievers have made huge progress in open-domain question-answering (Seo et al., 2019; Lee et al., 2019; Guu et al., 2020). Especially, dense passage retriever (Karpukhin et al., 2020) is a popular approach that learns separate question and document representations from task-specific training data. Lewis et al. (2020b); Izacard and Grave (2021) further show that question generation using models such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020) can be incorporated into DPR's training. Multi-task DPR (Wang et al., 2019) trains jointly on an extensive selection of retrieval datasets, which leads to better performance on downstream knowledge-intensive tasks.

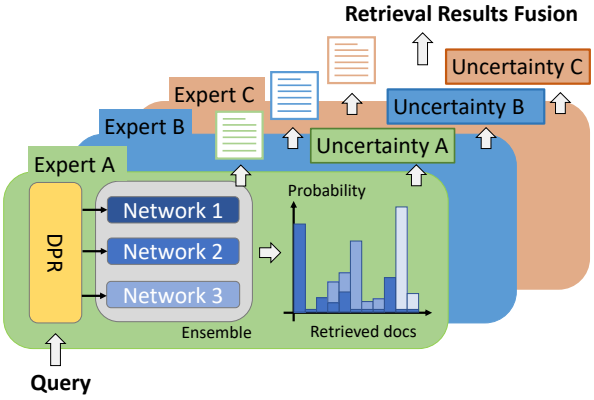


Figure 1: An illustration of model uncertainty fusion of 3 DPR experts, each with an ensemble of 3 fully-connected neural networks. Given a query, each DPR expert first retrieves top-k documents, followed by the uncertainty estimation using the corresponding ensemble. The weighted sum of predictions is then used to rerank the union of the retrieved documents.

Uncertainty Estimation Uncertainty estimation has wide applications in areas such as building safe AI systems (e.g., anomaly detection) (Amodei et al., 2016), especially for systems that include neural networks. Bayesian Neural Networks (BNNs) use probability distributions (MacKay, 1992; Neal, 2012) to represents the parameters of a neural net. Despite their compactness, in theory, BNNs have difficulty scaling to a large number of parameters and data points, which only works well in small-scale settings, e.g., MCMC methods (Neal, 2012). To adapt to modern networks' size, Gal and Ghahramani (2016) propose to use Monte Carlo dropout, which estimates model uncertainty by using Dropout (Srivastava et al., 2014) at test time. Another simple way to estimate uncertainty is ensembling, which aggregates the predictions of indi-

vidual ensemble members, and different weight initialization, data sampling, and regularization scheme is applied to encourage diversity in the ensemble (Lakshminarayanan et al., 2017; Snoek et al., 2019; Gustafsson et al., 2020; Pearce et al., 2020; Wen et al., 2020). Despite its simplicity, the ensembling approach scales well to large neural networks and massive datasets, while providing trustworthy uncertainty estimation.

3 Dense Passage Retrieval

Retrieval/Inference Given a collection of documents $\{d_1, d_2, \cdots, d_n\}$ and a question answering task, DPR (Karpukhin et al., 2020) encodes the questions and documents using a bi-encoder structure where encoders $f_Q(\cdot)$ and $f_D(\cdot)$ are independent functions that map a question/document into a low-dimensional, real-value vector. Specifically, the similarity s between the question q and document d is defined by the dot product between their encoded vectors $v_q = f_Q(q)$ and $v_d = f_D(d)$:

$$s = v_q^T v_d, \tag{1}$$

which is used as the ranking score. Both f_Q and f_D use BERT (Devlin et al., 2019) as the backbone model and the `[CLS]` vector as the output representation.

Training As pointed out by Karpukhin et al. (2020), training the encoders such that Eq. (1) becomes a good ranking function is essentially a metric learning problem (Kulis, 2012). Formally, let D be the random variable of documents, Q be the r.v. of questions, and $\mathcal{C}$ be the r.v. of the set of retrieved documents. Given a specific question q, let d^+ be the positive context that contains answers for q and $d_1^-, d_2^-, ...d_k^-$ be the negative contexts. The collection of contexts $\{d^+, d_1^-, d_2^-, ...d_k^-\}$ is first retrieved by BM25 which we denote as $\mathcal{C}_{\text{BM25}}$. The context prediction $p(D \mid Q = q, \mathcal{C} = \mathcal{C}_{\text{BM25}})$ is a softmax distrbution:

$$p(D \mid Q = q, \mathcal{C} = \mathcal{C}_{\text{BM25}}) = \frac{\exp(\lambda \cdot v_q^T v_D)}{\exp(\lambda \cdot v_q^T v_{d^+}) + \sum_{i=1}^{k} \exp(\lambda \cdot v_q^T v_{d_i^-})}, \tag{2}$$

where λ is the inverse temperature coefficient that controls the sharpness of the softmax distribution, which is often set to 1 during training. The negative log likelihood objective based on Eq. (2) is:

$$\begin{aligned}
&\mathcal{L}(q, \mathcal{C}_{\text{BM25}}) \\
&= -\log p(D = d^+ \mid Q = q, \mathcal{C} = \mathcal{C}_{\text{BM25}}) \\
&= -\log \frac{\exp(\lambda \cdot v_q^T v_{d^+})}{\exp(\lambda \cdot v_q^T v_{d^+}) + \sum_{i=1}^{k} \exp(\lambda \cdot v_q^T v_{d_i^-})}.
\end{aligned} \tag{3}$$

The single DPR expert and the joint-training DPR model follow the same training scheme. In the next section, we describe how the second option—model uncertainty fusion—is implemented.

4 Multi-Task Model Fusion

Given m question-answering tasks and m independent experts, the goal of multi-task model fusion is to find the optimal set of weights $\{w^{(i)}\}_{i=1}^m$ to combine all experts' predictions for each question. We use DPR as the expert model.

4.1 Ensemble Uncertainty Estimation

There are mainly two types of uncertainty: model uncertainty and data uncertainty (Malinin and Gales, 2018). Data uncertainty is often caused by mislabelling or missing features, while model uncertainty measures the confidence of the model's predictions given the training data, which is often used to identify whether a sample is within the training domain. Therefore, we use model uncertainty for weighting the experts' predictions, such that we know whether a question belongs to an expert's expertise.

As mentioned in Section 2, there are many ways to represent model uncertainty. In this work, we consider ensemble uncertainty due to its effectiveness and simplicity. The intuition is simple: the ensemble trained in a single domain will "agree" to similar predictions if in-domain samples occur, and will "disagree" otherwise. The disagreement would be more obvious if the functional space of the ensemble is complex enough, e.g., the space of neural networks. To quantify such uncertainty or "disagreement", we use *Mutual Information* (MI) (Poole et al., 2019) between the ensemble's prediction and its parameters as the proxy. For each DPR expert, we build an ensemble of m classifiers $\{p(D \mid \Theta = \theta_i, Q = q, \mathcal{C} = \mathcal{C}_{\text{DPR}})\}_{i=1}^m$ as in Eq. (2), where Θ denotes the r.v. of the ensemble parameters, θ_i denotes the parameters of the i^{th} ensemble member, and $\mathcal{C}_{\text{DPR}}$ denotes the collection of

contexts retrieved by the DPR expert. For simplicity, we will use $p(D \mid \theta_i, q, \mathcal{C}_{\text{DPR}})$ as a shorthand for the distribution. The mutual information $\mathcal{I}$ between D and Θ given a question q and a collection of contexts $\mathcal{C}_{\text{DPR}}$ is:

$$\begin{aligned}
&\mathcal{I}(D; \Theta \mid q, \mathcal{C}_{\text{DPR}}) \\
&= \mathcal{H}(D \mid q, \mathcal{C}_{\text{DPR}}) - \mathcal{H}(D \mid \Theta, q, \mathcal{C}_{\text{DPR}}) \\
&= \mathbb{E}_{p(D|\Theta,q,\mathcal{C}_{\text{DPR}})}[\ln p(D \mid \Theta, q, \mathcal{C}_{\text{DPR}})] \\
&\quad - \mathbb{E}_{p(D|q,\mathcal{C}_{\text{DPR}})}[\ln p(D \mid q, \mathcal{C}_{\text{DPR}})] \\
&\approx \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{p(D|\theta_i,q,\mathcal{C}_{\text{DPR}})}[\ln p(D \mid \theta_i, q, \mathcal{C}_{\text{DPR}})] \\
&\quad - \mathbb{E}_{p(D|q,\mathcal{C}_{\text{DPR}})}\left[\ln\left(\frac{1}{m}\sum_{i=1}^{m} p(D \mid \theta_i, q, \mathcal{C}_{\text{DPR}})\right)\right],
\end{aligned} \tag{4}$$

where $\mathcal{H}(\cdot)$ denotes the entropy operator, $\mathbb{E}[\cdot]$ denotes the expectation operator, and the approximations are done by Monte-Carlo simulation. The approximated mutual information is also upper-bounded by the log of the number of ensemble members m:

$$\mathcal{I}(D; \Theta \mid q, \mathcal{C}_{\text{DPR}}) \leq \ln m,$$

which gives us bounded uncertainty estimation for a certain domain. We normalize the mutual information and transform it into a confidence score w for weighting the DPR expert's prediction of q:

$$w = 1 - \frac{\mathcal{I}(D; \Theta \mid q, \mathcal{C}_{\text{DPR}})}{\ln m}. \tag{5}$$

4.2 Model Uncertainty Fusion

Given m DPR experts that are trained on m different tasks separately, we first encode the question and document representation into dense vectors $\{v_q^{(i)}\}_{i=1}^m$ and $\{v_d^{(i)}\}_{i=1}^m$ for all m experts, where the superscript represents the expert's id. We then build an ensemble of small neural networks, each uses the corresponding expert's $v_q^{(i)}$ as input and outputs another vector $u_q^{(i)}$ of the same dimension. We finally optimize the same objective function in Eq. (3) w.r.t each ensemble member $u_q^{(i)}$ and $v_d^{(i)}$ for each question-answering task.

During inference, given a new question q, we first retrieve m sets of top-k documents $\{C_{\text{DPR}}^{(i)}\}_{i=1}^m = \{d_1^{(i)}, d_2^{(i)}, \cdots, d_k^{(i)}\}_{i=1}^m$ using all m DPR expert's question vectors $\{v_q^{(i)}\}_{i=1}^m$ and document vectors $\{v_{d_1}^{(i)}, v_{d_2}^{(i)}, \cdots, v_{d_k}^{(i)}\}_{i=1}^m$. We then calculate the weights $\{w^{(i)}\}_{i=1}^m$ for each question according to Eq. (5) using the ensemble vectors $\{u_q^{(i)}\}_{i=1}^m$ and $\{C_{\text{DPR}}^{(i)}\}_{i=1}^m$ for question q. Finally, we re-rank the union of m sets of top-k documents using the uncertainty-weighted sum of each expert's score. The final score $\mathcal{S}(q, d_j)$ of a document d_j given question q is:

$$\mathcal{S}(q, d_j) = w^{(1)} s_j^{(1)} + w^{(2)} s_j^{(2)} \cdots + w^{(m)} s_j^{(m)}, \tag{6}$$

where $s_j^{(i)} = v_q^{(i)T} v_{d_j}^{(i)}$ according to Eq. (1). If we do not have a score from the i^{th} expert for a document d, we will use the minimum of $\{s_j^{(i)}\}_{j=1}^k$ as the ranking score for d. Fig. 1 visualizes the aforementioned retrieval fusion process of our method during inference.

4.3 Uncertainty Calibration

Despite its simplicity and effectiveness, one drawback of ensemble uncertainty is that it doesn't have a closed-form expression and the prediction of each ensemble might have a different range (Pearce et al., 2020). Therefore, the ensemble uncertainty needs to be calibrated before fusion, such that the confidence of an expert matches its prediction accuracy. We use the Expected Calibration Error (ECE) (Guo et al., 2017) as the metric where we search for the best inverse temperature in Eq. (3) on the dev sets for each expert to minimize the ECE. As ECE mainly uses the term "confidence" which is the w in Eq. (5), we switch to "confidence" instead of "uncertainty" in the following. We partition the samples in the dev set into T equally-spaced bins and take the weighted average of the confidence-accuracy difference:

$$\text{ECE} = \sum_{i=1}^{T} \frac{|B_i|}{N} \left|\text{conf}(B_i) - \text{acc}(B_i)\right|, \tag{7}$$

where B_i is the i^{th} bin and N is the number of samples. Functions $\text{conf}(B_i)$ and $\text{acc}(B_i)$ are the average confidence and top-1 accuracy within the i^{th} bin, respectively. Each confidence score is computed by an ensemble according to Eq. (5).

5 Experimental Setup

We follow the DPR paper (Karpukhin et al., 2020) to train and evaluate our dense retrievers. We replicate their results on all benchmark datasets, with a

Retriever	Top-20					Top-100				
	NQ	Trivia	WQ	TREC	SQuAD	NQ	Trivia	WQ	TREC	SQuAD
DPR-Single-domain	79.1	78.9	71.0	85.1	62.1	85.9	84.5	80.2	92.2	76.8
DPR-Single-worst	46.3	58.0	57.9	75.8	41.4	61.9	71.4	74.1	86.5	59.1
DPR-Multi (w/o SQuAD)	79.5	**78.9**	75.0	88.8	52.0	86.1	**84.8**	83.0	93.4	67.7
DPR-MUF (w/o SQuAD)	**79.8**	78.2	**76.2**	89.3	57.7	**86.5**	84.4	83.9	94.7	72.0
DPR-MUF (w/o domain)	68.9	74.1	73.5	89.6	57.7	79.4	82.1	82.5	94.4	72.0
DPR-MUF	79.5	78.6	75.9	**90.2**	**64.6**	86.4	84.7	**84.0**	**95.0**	**78.3**

Table 2: Top-20 and Top-100 retrieval accuracy (%) on benchmark QA test sets. Each score represents the percentage of top 20/100 retrieved passages that contain answers. All methods containing "DPR-MUF" stand for our Model Uncertainty Fusion (MUF) method. See Section 6.1 for details of different models.

maximum score difference between ours and their numbers of 1%. This work only focuses on retrieval accuracy as we only improve the retriever. We finally perform sensitivity analysis for the ensemble and uncertainty visualization of our fusion approach. More details are provided in Appendix A.

Datasets We train individual DPR models on 5 standard benchmark QA tasks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Trivia) (Joshi et al., 2017), WebQuestions (WQ) (Berant et al., 2013), CuratedTREC (TREC) (Baudiš and Šedivỳ, 2015), SQuAD-1.1 (SQuAD) (Rajpurkar et al., 2016). We evaluate the retriever models on the test sets of the aforementioned datasets, as well as their random mixes to test the out-of-distribution performance.

For retrieval, we chunk the Wikipedia collections (Guu et al., 2020) into passages of 100 words as in Wang et al. (2019), which yields about 21 million samples in total. We follow Karpukhin et al. (2020) using BM25 (Robertson and Zaragoza, 2009; Lin et al., 2021) to select the positive and negative passages.

Models and Training We first train independent DPR models on the training set of NQ, TriviaQA, WQ, CuratedTREC, and SQuAD-1.1 separately following Karpukhin et al. (2020). We then encode the training sets into dense vectors as the input to the ensemble. We train an ensemble of 20, 2-layer fully connected neural networks with 512 units for 100 epochs. We optimize the objective function in Eq. (3) with learning rate of 2e-05 using Adam (Kingma and Ba, 2015). We use different sub-batches and weight initialization to train each ensemble member to encourage diversity. The rest of the hyperparameter setting remains the same as described in Karpukhin et al. (2020).

Inference: Retrieval Fusion Given a question q during inference, a set of top-k documents is first retrieved by each DPR expert. For each expert, we use the corresponding ensemble to predict the expert's retrieved documents and obtain a collection of dot-product scores. We then apply softmax activation on the dot-products, yielding a collection of distributions over the retrieved documents. We calculate the normalized mutual information between the ensemble's predictions and the parameters as in Eq. (4) and Eq. (5), using it as the weight for the expert as described in Eq. (6) given the question.

In addition, we calibrate each ensemble's uncertainty prediction individually using the expected calibration error (ECE) (Guo et al., 2017) according to Eq. (7), as each ensemble from different domains might have a different range of uncertainty. We find the lowest ECE score is achieved with the inverse temperature λ of the softmax activation in Eq. (3) setting to be 1e-3. Finally, we normalize the calibrated uncertainty and re-rank the union of retrieved documents using the uncertainty-weighted sum of experts' scores. For documents that do not have all experts' scores, we use the minimum of the missing expert's prediction as the ranking score.

6 Results and Analysis

6.1 Benchmark Dataset Retrieval

Tbl. 2 shows retrieval performance using different types of DPR models on 5 benchmark datasets. We briefly describe each configuration below.

DPR-Single-domain: A single DPR model trained and tested on the same domain.

DPR-Single-worst: A single DPR model trained on one domain and transferred to the target test set in zero-shot and has the worst performance among all experts.

DPR-Multi (w/o SQuAD): A multi-task DPR model trained on the joint dataset of {NQ, Trivia, WQ, and Trec} without the SQuAD dataset, as implemented in Karpukhin et al. (2020).

DPR-MUF: Our model uncertainty fusion method using experts from {NQ, Trivia, WQ, Trec, and SQuAD}, which is our main approach.

DPR-MUF (w/o SQuAD): Our model uncertainty fusion method using experts from {NQ, Trivia, WQ, and Trec} without SQuAD to align with DPR-Multi (w/o SQuAD).

DPR-MUF (w/o domain): Our model uncertainty fusion method using all experts except DPR-Single-domain, to investigate out-of-domain generalization.

We can see from Tbl. 2 that our model uncertainty fusion method (DPR-UF) achieves the best performance on almost all benchmark QA datasets except Trivia, in terms of top-20/100 accuracy. The original multi-task DPR model does not include SQuAD for joint training as "SQuAD is limited to a small set of Wikipedia documents and thus introduces unwanted bias" (Karpukhin et al., 2020). In comparison, our DPR-UF that includes the SQuAD dataset significantly improves the performance on SQuAD as well as on other datasets. In addition, we find that our DPR-UF (w/o SQuAD) not only manages to beat the joint-training DPR trained on {NQ, Triviva, WQ, and Trec}, but also outperforms it on SQuAD by a large margin (10% in top-100 accuracy). We also test the fusion of experts without the one trained on the target domain, i.e., DPR-MUF (w/o domain), whose performance turns out to be maintained at a reasonable level.

One interesting result we find in the experiments is that DPR-MUF without the CuratedTrec/WQ expert outperforms the CuratedTrec/WQ experts on their domain test sets. We suspect that the CuratedTrec and WQ datasets are too small and might be covered by other datasets. Therefore, it is not surprising that the CuratedTrec and WQ experts trained on small data regimes fail to outperform the larger expert union.

6.2 Mixed-Dataset Retrieval

In a real-world application, the retriever often needs to deal with questions from different sources instead of just a single task. To test the ability to retrieve out-of-distribution questions, we evenly sample 5 subsets of 3,000 test questions from 4 benchmark datasets (NQ, Trivia, WQ, SQuAD). We average top-20/100 accuracy on the 5 subsets as the final accuracy. In addition, we design two oracle models which serve as references:

Retriever	Top-20	Top-100
DPR-Single-NQ	65.4	76.1
DPR-Single-Trivia	66.6	77.7
DPR-Single-WQ	54.1	68.3
DPR-Single-Trec	60.5	74.0
DPR-Single-SQuAD	57.3	73.2
DPR-Multi (w/o SQuAD)	71.5	80.7
DPR-MUF (w/o SQuAD)	72.7	81.7
DPR-MUF	**74.2**	**83.3**
DPR-Oracle-Indicator	72.8	82.0
DPR-Oracle-Bayesian	73.3	82.3

Table 3: Top-20/100 retrieval accuracy (%) on random mixes of 4 benchmark QA test sets. We average the metrics from 5 evenly-sampled subsets of 3,000 samples from NQ, Trivia, WQ, and SQuAD.

DPR-Oracle-Indicator: A mixture of experts that knows the domain each question comes from, and uses the corresponding expert for retrieval.

DPR-Oracle-Bayesian: A mixture of experts that uses Bayesian optimization (Frazier, 2018) to search for the weights. We initialize the weights with the indicator function and use scikit-optimize[1] to search for the optimal weight for 50 iterations for each question. This process is not guaranteed to find the best sets of weights as Bayesian optimization does not always find the global optimum. Although it is not the exact oracle, this is the best model we could find as exhaustive search is impractical due to its exponential time complexity.

Tbl. 3 shows that all single retrievers have severe performance degradation on the randomly mixed dataset, which is expected as they only specialize in their own domain. In contrast, the two multi-task models, DPR-Multi (w/o SQuAD) and DPR-MUF (w/o SQuAD), manage to maintain high scores on the random mixes, which reach the performance of the two oracle models. Moreover, DPR-MUF even outperforms both the indicator oracle and the Bayesian oracle, suggesting the benefits of using uncertainty to fuse the predictions of multiple experts from different domains.

[1] https://scikit-optimize.github.io/

Retriever	Top-20	Top-100
NQ		
BM25	62.9	78.3
+ DPR-Single-domain	82.5	88.2
+ DPR-Multi (w/o SQuAD)	82.6	88.6
+ DPR-MUF (w/o SQuAD)	**82.7**	**88.6**
+ DPR-MUF	82.0	88.2
Trivia		
BM25	76.4	83.2
+ DPR-Single-domain	82.8	86.8
+ DPR-Multi (w/o SQuAD)	82.6	86.5
+ DPR-MUF (w/o SQuAD)	**82.9**	**87.0**
+ DPR-MUF	82.4	86.5
WQ		
BM25	62.4	75.5
+ DPR-Single-domain	74.3	82.6
+ DPR-Multi (w/o SQuAD)	77.1	84.4
+ DPR-MUF (w/o SQuAD)	77.9	84.5
+ DPR-MUF	**78.1**	**84.9**
TREC		
BM25	80.7	89.9
+ DPR-Single-domain	90.1	94.7
+ DPR-Multi (w/o SQuAD)	90.1	95.0
+ DPR-MUF (w/o SQuAD)	90.8	95.5
+ DPR-MUF	**91.2**	**95.7**
SQuAD		
BM25	71.1	81.8
+ DPR-Single-domain	75.6	84.9
+ DPR-Multi (w/o SQuAD)	75.1	84.4
+ DPR-MUF (w/o SQuAD)	76.7	86.3
+ DPR-MUF	**78.7**	**86.7**

Table 4: Top-20/100 retrieval accuracy (%) of BM25 and DPR-BM25 hybrid model on test sets of NQ, Trivia, WQ, CuratedTrec, and SQuAD.

6.3 DPR-BM25 Hybrid Retrieval

Karpukhin et al. (2020) show that DPR can be combined with BM25 to further improve retrieval performance. Ma et al. (2021) further fine-tune the parameters for BM25 and obtain better accuracy using the Pyserini IR toolkit (Lin et al., 2021). We follow the experimental setting in Ma et al. (2021) where we re-rank the union of the top-1000 passages retrieved by DPR and BM25 separately, using the weighted sum of the two scores as the ranking value. We search for the optimal weights for BM25 and DPR on the dev set of each QA dataset.

Tbl. 4 shows the top-k accuracy of hybrid retrievers using the combination of different DPR models and BM25. Our model uncertainty method manages to outperform single DPR experts and multi-task, joint-training DPR on all benchmark QA datasets. Specifically, DPR-UF (w/o SQuAD) has the best performance on NQ and Trivia, while DPR-UF includes all experts, which has the best performance on WQ, CuratedTrec, and SQuAD. We conjecture it's because NQ and Trivia are much larger and therefore the SQuAD expert might have more conflict with BM25.

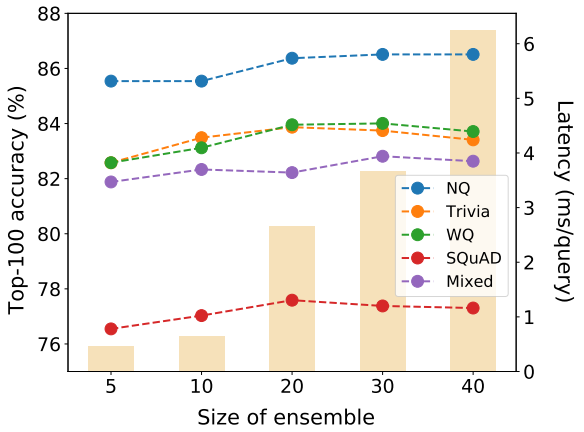


Figure 2: Line plot of top-100 accuracy (%) and bar chart of latency (ms/query) relative to a single DPR model of our method w.r.t. the size of the ensemble on NQ, Trivia, WQ, SQuAD, and their random mixes.

6.4 Ensemble Sensitivity and Latency

In this section, we analyze how sensitive the retrieval performance of the uncertainty fusion method is w.r.t. the ensemble size. Fig. 2 shows the top-100 accuracy and the relative latency of different sizes of ensembles. The accuracy increases as the size of the ensemble grows until it hits 20, which then plateaus or decreases. We conjecture it is because the functional space of the ensemble is not complex enough as we only use a 2 layer neural network with 512 units as the individual component. Therefore, there are only limited ways for the model to overfit the training sets, resulting in the saturation in diversity w.r.t. the ensemble size.

However, we find that overall, these results are good enough while having reasonable latency. The latency (ms/question) of the model is measured relative to a standard DPR model, which mainly includes the ensemble forward inference time. We evaluate the inference speed on a server with an Intel Xeon CPU E5-2699 v4 @ 2.20GHz. In summary, the retrieval accuracy is stable w.r.t. the ensemble size, and one can choose the ensemble size

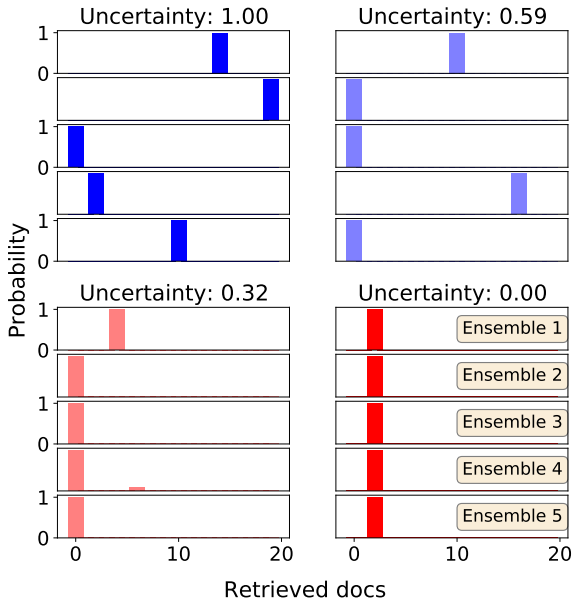


Figure 3: A visualization of uncertainty estimation using mutual information between the ensemble's predictions and parameters. Each subplot shows the prediction of 5 ensemble on the top-20 documents retrieved by the DPR expert. The uncertainty decreases as more ensemble members "agree" with each other.

to trade-off between accuracy and latency for different application scenarios.

6.5 Uncertainty Visualization

We visualize the model uncertainty in this section for better understanding. Fig. 3 shows 5 ensemble predictions of the top-20 documents on 4 samples from NQ with different uncertainty scores. Each strip in a subplot represents one ensemble member and all members in the same subplot share the same documents retrieved by the DPR expert on NQ. As we use small inverse temperature λ (1e-3) for the softmax distribution in Eq. (3), the probability mass of each distribution mainly concentrates on the top-1 document, which is the tallest bar in each strip. If the top-1 predictions from different ensemble members overlap at the same document, we say these members "agree" with each other and therefore the overall ensemble has low uncertainty. The overlap is quantified by Eq. (4) in practice. As we can see from Fig. 3, the ensemble has full uncertainty (1.0/1.0) when their top-1 predictions do not overlap at all, and has zero uncertainty when its members' predictions completely overlap. In other cases, the more overlap or "agreement" on the top-1 prediction, the less uncertain the ensemble is.

6.6 Space-Speed-Flexibility Trade-off

Despite the promising results we have shown in the previous section, the model uncertainty fusion method also has its drawback in open-domain question-answering. For now, all the experts are individually trained in their own domain but share a common corpus. That says if we have m experts, then the index size will grow by $O(m)$ compared to a single multi-task, joint-training model.

However, we argue that there's no free lunch as the joint-training model suffers from other problems such as data conflict mentioned before, as well as catastrophic forgetting: If new tasks are added, the joint-training model usually requires to re-train on the union of all tasks again to maintain performance on previous tasks, while our model only needs to train on the new task's data and the new expert can be directly added to the current set of models. Therefore, both methods have their pros and cons according to different application scenarios, and it is upon the users to consider the space-speed-flexibility trade-off. For memory and efficiency issues, possible solutions would be either learning a shared, query-agnostic index for all experts or leveraging model compression methods to compress the size of expert models.

7 Conclusions

In this paper, we propose a model fusion approach for multi-task dense retrieval. Instead of training a single DPR model on the union of datasets from different distributions, we leverage model uncertainty to merge different DPR expert's predictions during test time. For each expert, we train an ensemble of small neural networks on top of the pre-trained expert's dense representations and use the mutual information between the ensemble parameters and predictions as the weight, which can be interpreted as the "disagreement" among the ensemble.

We compare our model uncertainty fusion approach with single specialists and the multi-task, joint-training DPR model on 5 benchmark QA datasets, as well as their dataset random mixes to test out-of-distribution performance. Extensive experiments show that our method manages to outperform these approaches in terms of top-20/100 accuracy on most datasets, while it can also be combined with sparse retrieval methods such as BM25 for further performance gains. Our proposed method is simple to implement and effective while enjoying the benefits of continual learning, faster

training speed as the experts can be trained in parallel, as well as the flexibility to combine experts from different domains.

For future research directions, one could leverage model compression techniques to reduce the index size, or knowledge distillation to learn a single student model from the experts. Finally, learning a question-agnostic document index can further save storage space and enhance inference speed for this model fusion method.

Acknowledgements

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada; computational resources were provided by Compute Canada.

References

Dario Amodei, Christopher Olah, J. Steinhardt, P. Christiano, J. Schulman, and Dandelion Mané. 2016. Concrete problems in ai safety. *ArXiv*, abs/1606.06565.

Petr Baudiš and Jan Šedivỳ. 2015. Modeling of the question answering task in the YodaQA system. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 222–228. Springer.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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, Vancouver, Canada. Association for Computational Linguistics.

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

Peter I. Frazier. 2018. A tutorial on Bayesian optimization. *ArXiv*, abs/1807.02811.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B. Schon. 2020. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319.

Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

Minh Hoang, Nghia Hoang, Bryan Kian Hsiang Low, and Carleton Kingsford. 2019. Collective model fusion for multiple black-box experts. In *International Conference on Machine Learning*, pages 2742–2750. PMLR.

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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

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.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Brian Kulis. 2012. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6402–6413.

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, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. 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. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use Python toolkit to support replicable ir research with sparse and dense representations. *ArXiv*, abs/2102.10073.

Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. 2020. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160.

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. 2021. A replication study of dense passage retriever. *ArXiv*, abs/2104.05740.

David J. C. MacKay. 1992. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1098–1111, Online. Association for Computational Linguistics.

Andrey Malinin and Mark J. F. Gales. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 7047–7058.

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.

Radford M. Neal. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. 2020. Uncertainty in neural networks: Approximately Bayesian ensembling. In *International Conference on Artificial Intelligence and Statistics*, pages 234–244. PMLR.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.

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. *JMLR*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 13969–13980.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain 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 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2021. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357.

Yeming Wen, Dustin Tran, and Jimmy Ba. 2020. BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 DPR setting

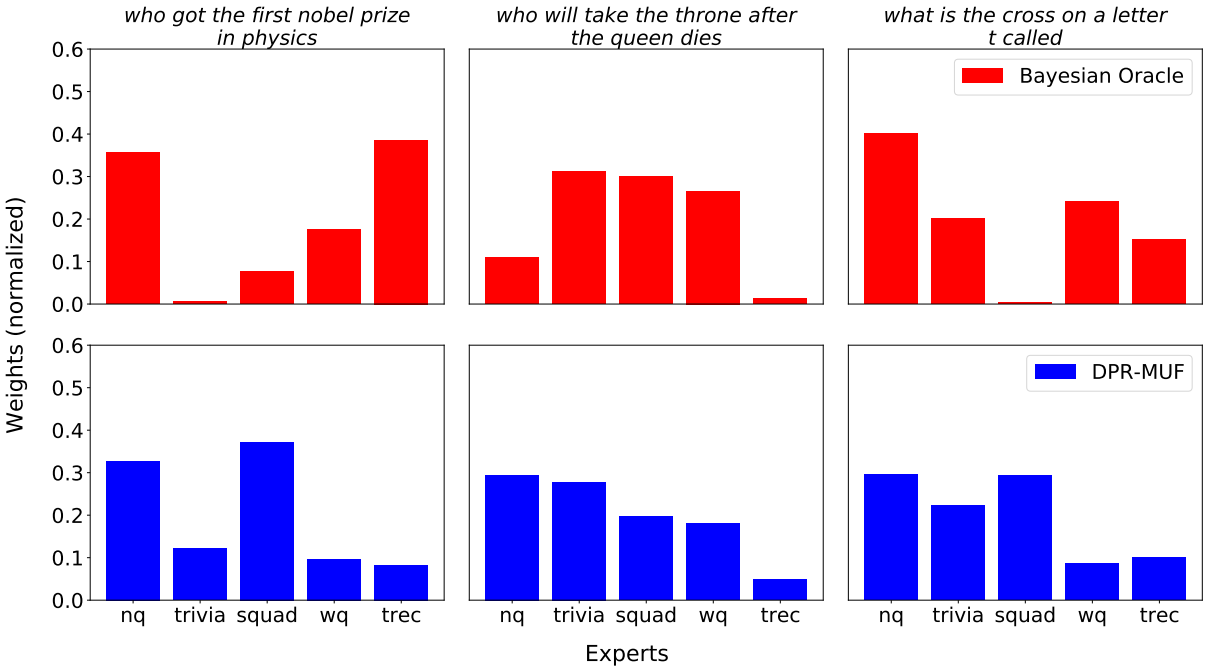
We provide details of training and inference for the DPR model here. Most of them can be found in the original DPR paper (Karpukhin et al., 2020). For the ensemble, we use the pre-trained DPR's `[CLS]` representations as inputs and train the ensemble on the same datasets.

Training We optimize the objective function in Eq. (3) using in-batch negative training. Each question of the batch is accompanied by a positive passage and a set of negative ones retrieved by BM25. The technique of in-batch negatives (Gillick et al., 2019; Karpukhin et al., 2020) boosts the number of training examples by viewing each positive context in the batch as the only positive and the rest of the batch as the negatives. Specifically, given a batch of B questions and each one is paired with just a positive passage (which is the base case): Let $Q \in \mathbb{R}^{B\times D}$ and $P \in \mathbb{R}^{B\times D}$ be the batches of question and passage embeddings of D dimensions. The in-batch negative technique calculates the similarity score matrix S using the outer-product $QP^T \in \mathbb{R}^{B\times B}$, every row of which contains a positive score and $B-1$ negative scores for a question. In this way, the computation is reused for efficient training. As for the strategy of selecting positive and negative samples for questions, we concatenate each question with answers to retrieve the top-100 documents using BM25. We then use the documents that contain answers as the positive passages and the rest as hard negatives.

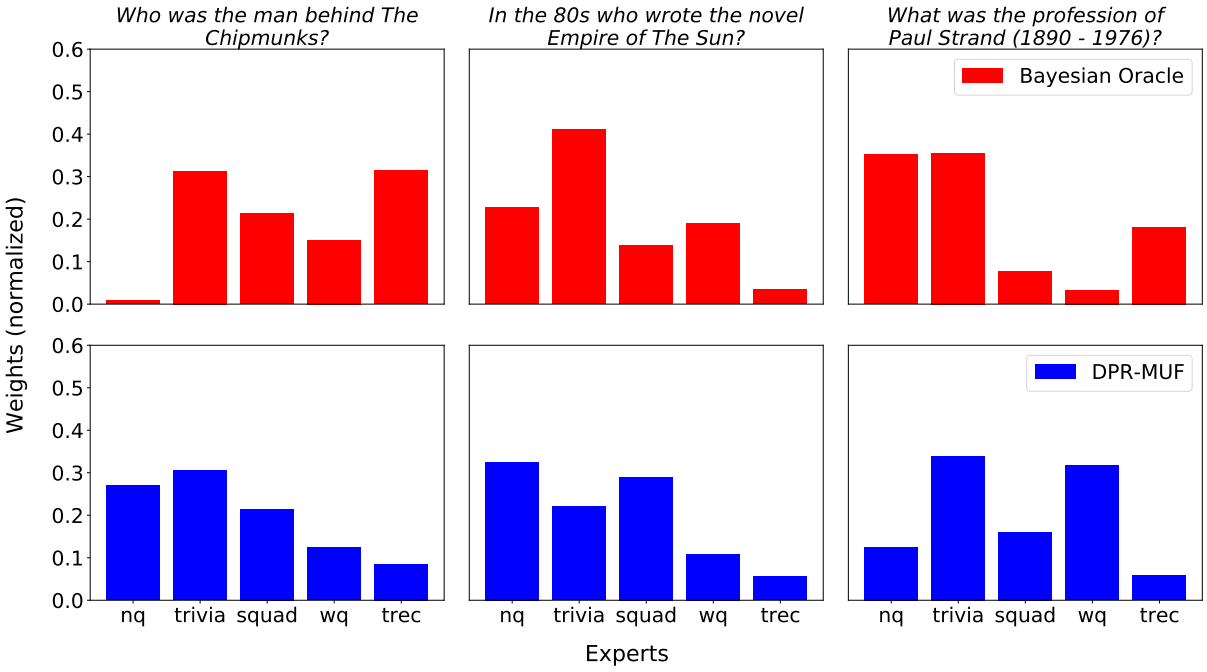
Inference During inference, we encode all the passages into dense vectors using the passage encoder and index them using FAISS (Johnson et al., 2021), which is an efficient, open-source library for vector searching and indexing that can scale to millions of vectors.

A.2 Uncertainty Weight Distribution

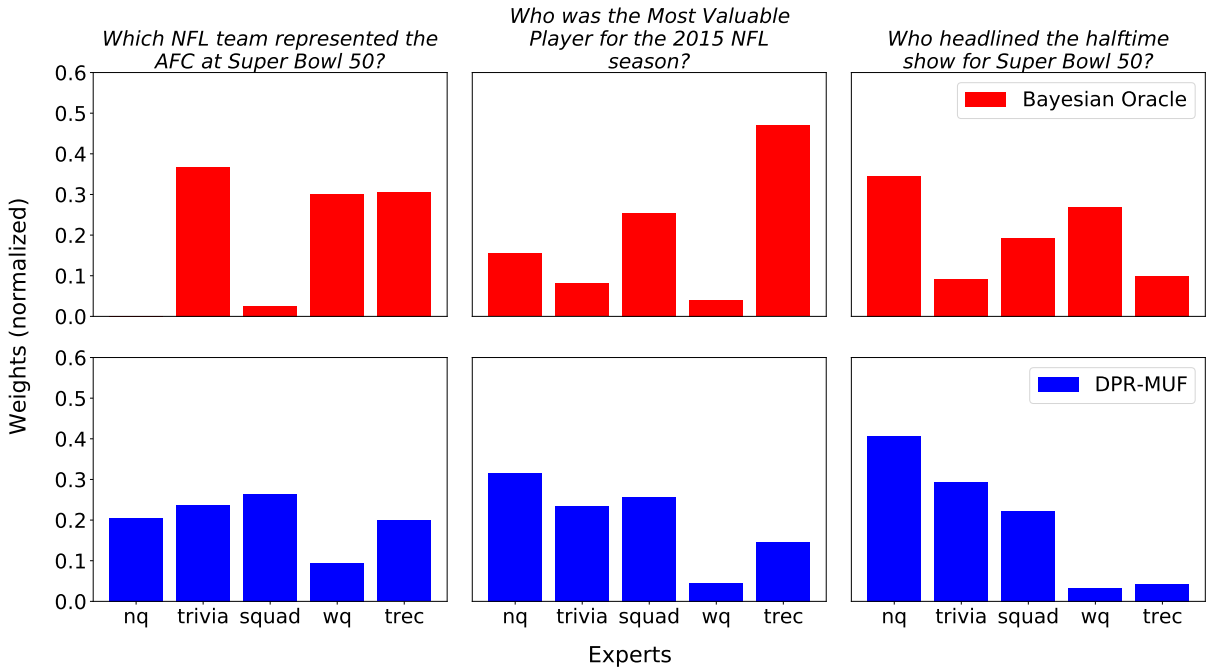
Section 6 shows that weighting the retrieval results from different experts leads to better generalization. In this section, we inspect the weight distribution over experts given a question, and see whether the fusion weights have a sharp distribution (i.e., mainly using a single expert for each question) or a more scattered one (i.e., a rather even mixture of experts). It turns out that both our uncertainty fusion method and the Bayesian oracle in Section 6.2 have more scattered weights for most questions. Fig. 5 shows the weight distribution over experts of some example questions from the NQ, Trivia, SQuAD, and WQ datasets. The distribution of the Bayesian oracle looks a little bit different from the uncertainty fusion method, which we conjecture is because we initialize the weight of the Bayesian optimization with the indicator function for faster searching. Therefore, it results in another solution whose probability often concentrates more on the domain's expert.



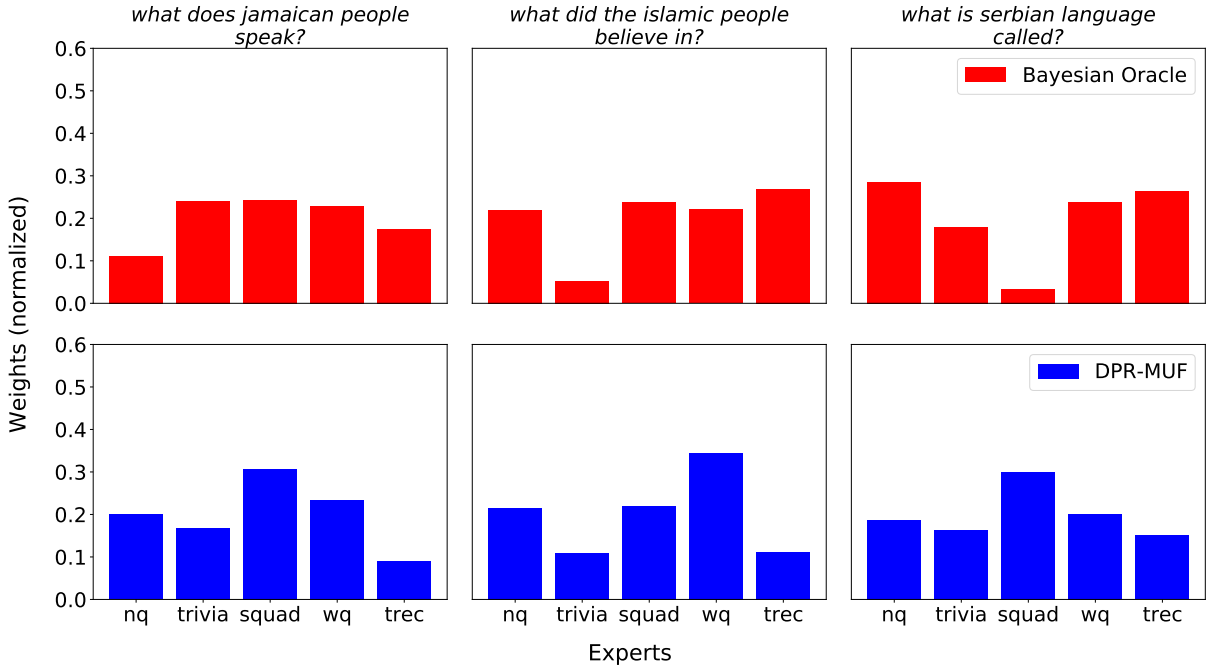
(a) Examples from NQ's test set.



(b) Examples from Trivia's test set.



(a) Examples from SQuAD's test set.



(b) Examples from WQ's test set

Figure 5: Weight distributions of the DPR-MUF model and the Bayesian oracle on some example queries from the NQ, Trivia, SQuAD, and WQ datasets. Both methods include independent experts trained on {NQ, Trivia, SQuAD, WQ, and Trec}. Despite differences in their weight distributions, these methods all have scattered distributions over each expert's prediction, which shows that fusing different expert's retrieval results indeed helps with generalization.